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 другий (магістерський)

Дослідження методів виявлення та розпізнавання жестів рук у системах жестової мови
(тема)

Виконав:
студент 2 курсу, групи ІНФм-23-1
Легкий М.Г.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Машталір С.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Легкому Максиму Григоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів виявлення та розпізнавання жестів рук у системах жестової мови

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 28 грудня 2024 р.

3. Вихідні дані до роботи математичні моделі перетворення зображень, перелік використаних програмних засобів, теоретичні відомості про методи сегментації, виявлення та класифікації жестів за допомогою нейронних мереж.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд сучасних методів комп'ютерного зору для сегментації та виявлення рук.

2. Аналіз математичних моделей, зокрема перетворення Хафа, для знаходження геометричних примітивів.

3. Дослідження ефективності нейронних мереж для розпізнавання жестів.

4. Розробка та оптимізація прототипу системи для автоматичного розпізнавання жестів у реальному часі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) схема архітектури нейронної мережі (CNN, RNN), ілюстрації виявлення рук у різних кольорових просторах (RGB, HSV), приклади сегментації рук та аналізу жестів, порівняльні графіки ефективності методів (точність, швидкодія), результати роботи прототипу на тестових зображеннях та відео.

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	25.11.24-26.11.24	
3	Аналіз літератури з досліджуваної проблеми	26.11.24-28.11.24	
4	Аналіз методів визначення та розпізнання жестів рук	28.11.24-03.11.24	
5	Розробка системи для розпізнання жестів	03.12.24-10.12.24	
6	Програмна реалізація	10.12.24-13.12.24	
7	Оформлення пояснювальної записки	13.12.24-18.12.24	
8	Перевірка на плагіат	18.12.2024	
9	Рецензування	23.12.2024	
10	Підготовка презентації та доповіді	25.12.2024	
11	Занесення роботи в електронний архів	07.01.2025	
12	Попередній захист кваліфікаційної роботи	07.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Машталір С.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 66 с., 3 табл., 12 рис., 43 джерело.

ГЛИБИННЕ НАВЧАННЯ, ЖЕСТОВА МОВА, КОМП'ЮТЕРНИЙ ЗІР, МУЛЬТИМОДАЛЬНІ МЕТОДИ, РОЗПІЗНАВАННЯ ЖЕСТІВ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ.

Об’єктом дослідження є методи розпізнавання жестів рук у системах жестової мови. Метою дослідження є розробка моделей та алгоритмів, що комбінують згорткові та рекурентні нейронні мережі для ефективного виявлення та розпізнавання жестів у реальному часі з можливістю адаптації до різних культур та мовних систем.

Використано методи глибинного навчання, комп’ютерного зору та мультимодальної інтеграції даних, проведено аналіз існуючих методів та розроблено алгоритм, що поєднує CNN та RNN для обробки статичних і динамічних жестів. Реалізовано методи передобробки даних для підвищення стійкості моделі до змін освітлення та фону.

У результаті створено комп’ютерну модель, яка демонструє високу точність та швидкодію в розпізнаванні жестів рук у реальних умовах, а також адаптивність до різних систем жестової мови, сприяючи покращенню комунікації та інтеграції людей з вадами слуху в суспільство.

DEEP LEARNING, SIGN LANGUAGE, COMPUTER VISION, MULTIMODAL METHODS, GESTURE RECOGNITION, RECURRENT NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORKS.

The object of the research is the methods of hand gesture recognition in sign language systems. The aim of the research is to develop models and algorithms that combine convolutional and recurrent neural networks for the effective detection and recognition of gestures in real-time, with the ability to adapt to different cultures and sign language systems.

Deep learning, computer vision, and multimodal data integration methods were employed, an analysis of existing methods was conducted, and an algorithm combining CNN and RNN was developed to process both static and dynamic gestures. Data preprocessing techniques were implemented to enhance the model's robustness against changes in lighting and background.

As a result, a computer model was created that demonstrates high accuracy and speed in recognizing hand gestures under real-world conditions, as well as adaptability to various sign language systems. This contributes to improving communication and the integration of individuals with hearing impairments into society.

ЗМІСТ

Вступ 8
1 Аналіз систем розпізнавання жестової мови 9
1.1 Актуальність застосування систем розпізнавання жестової мови 9
1.2 Основи жестової мови 10
1.3 Методи виявлення рук у системах комп'ютерного зору 12
1.3.1 Класичні методи виділення руки з фону 12
1.3.2 Методики визначення ключових точок та поз людини 14
1.3.3 Сучасні фреймворки на основі глибинного навчання 16
1.4 Методи розпізнавання жестів уже виявлених рук 18
1.4.1 Глибинні згорткові нейронні мережі (CNN) 19
1.4.2 Рекурентні нейронні мережі (RNN, LSTM, GRU) 20
1.4.3 Поєднання класичних алгоритмів та глибинних моделей 21
1.5 Постановка задачі дослідження 21
2 Математичні моделі фільтрації зображень 24
2.1 Теоретичні основи фільтрації та попередньої обробки зображень 24
2.2 Математична модель та архітектура згорткових нейронних мереж 27
2.3 Аналіз ефективності різних моделей та алгоритмів 32
2.3.1 CNN проти класичних методів: порівняння точності та швидкодії 32
2.3.2 Рекурентні нейронні мережі (LSTM) для послідовних жестів 33
2.3.3 Порівняння сталих методів з сучасними глибинними нейронними мережами 34
2.4 Переваги та недоліки розглянутих підходів 36
2.4.1 Чутливість до освітлення, фону та шуму 36
2.4.2 Вимоги до апаратних ресурсів 37
2.4.3 Можливість узагальнення та масштабування 38
2.4.4 Точність та надійність розпізнавання 39
2.4.5 Простота інтеграції та використання 40

2.4.6 Ефективність навчання та адаптації 40
3 Дослідження комп'ютерної моделі фільтрації зображень 42
3.1 Обґрунтування вибору програмного середовища та інструментів ... 42
3.1.1 Використання мови Python 42
3.1.2 Бібліотека Mediapipe 43
3.1.3 TensorFlow та Keras 44
3.1.4 OpenCV 44
3.2 Опис експериментальної моделі 46
3.2.1 Архітектура обраної глибинної мережі 46
3.2.2 Набір даних, підготовка та аугментація 47
3.3 Навчання та оптимізація моделі 49
3.3.1 Підбір гіперпараметрів, функцій втрат та оптимізаторів 50
3.3.2 Використання попередньо підготовлених моделей та перенавчання 52
3.3.3 Результати різних варіантів та оптимізацій 52
3.3.4 Результати тренування 53
3.4 Результати експериментів та їх аналіз 53
3.4.1 Робота розпізнавання самої руки 54
3.4.2 Визначення жестів 56
3.4.3 Порівняння точності та продуктивності 57
3.4.4 Аналіз помилок та пропозиції щодо покращення 58
Висновки 60
Перелік джерел посилання 62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional Neural Networks (Згорткові нейронні мережі)

RNN – Recurrent Neural Networks (Рекурентні нейронні мережі)

LSTM – Long Short-Term Memory (Довга короткочасна пам’ять)

SGD – Stochastic Gradient Descent (Стохастичний градієнтний спуск)

GRU – Gated Recurrent Unit (Рекурентний блок із затворами)

ASL – American Sign Language (Американська жестова мова)

SVM – Support Vector Machines (Методи опорних векторів)

УЖМ – українська жестова мова

БЖМ – британська жестова мова

HSV – Hue, Saturation, Value (Відтінок, насиченість, яскравість)

YCrCb – Luminance, Chrominance (Яркість, хромінанс)

FPS – Frames Per Second (Кадрів на секунду)

GPU – Graphics Processing Unit (Графічний процесор)

TPU – Tensor Processing Unit (Тензорний процесор)

RMSProp – Root Mean Square Propagation (Пропагація середньоквадратичного значення)

ReLU – Rectified Linear Unit (Виправлена лінійна одиниця)

API – Application Programming Interface (Інтерфейс програмування застосунків)

BGR – Blue, Green, Red (Синій, зелений, червоний — порядок кольорів у деяких форматах)

RGB – Red, Green, Blue (Червоний, зелений, синій — порядок кольорів у форматах зображень)

RNN – Recurrent Neural Networks (Рекурентні нейронні мережі)

ВСТУП

У сучасну епоху інтенсивного розвитку інформаційних технологій особливого значення набувають системи, здатні полегшувати взаємодію між людьми та комп'ютерами, а також розширювати можливості комунікації між людьми з різними потребами. Однією з таких актуальних проблем є автоматичне розпізнавання жестової мови [1], яке здатне суттєво покращити доступність інформації та засобів спілкування для людей з порушеннями слуху.

Жестова мова базується на візуальних жестах рук, міміці й положенні тіла. Вона є повноцінною мовною системою зі своєю граматикою. Однак для людей, які не володіють цією мовою, існує бар'єр у спілкуванні, що ускладнює інтеграцію осіб з порушеннями слуху в суспільство.

Розвиток глибинного навчання, зокрема згорткових [2] нейронних мереж, дозволив значно вдосконалити технології комп'ютерного зору. Їх використання у розпізнаванні жестів дає змогу автоматизувати переклад жестової мови у текст чи усний формат. Це знаходить застосування у навчанні, мобільних застосунках, відеоконференціях, інформаційних терміналах та засобах зв'язку.

Актуальність дослідження полягає у необхідності створення більш доступних та точних систем для автоматичного виявлення рук та розпізнавання жестів у реальному часі. Такі системи [3] сприятимуть зниженню комунікаційних бар'єрів, інтеграції людей з порушеннями слуху в соціум, а також розширенню можливостей застосування комп'ютерного зору в інтерактивних середовищах. Крім того, постійний розвиток апаратного забезпечення (покращення камер, мобільних процесорів та графічних чипів) і програмних методів (вдосконалені архітектури нейронних мереж, нові алгоритми оптимізації) дає підстави очікувати більш швидкої та точної обробки візуальної інформації.

1 АНАЛІЗ СИСТЕМ РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ

1.1 Актуальність застосування систем розпізнавання жестової мови

Сучасне суспільство дедалі більше орієнтується на інформаційні технології та візуально-комунікаційні канали, однак люди з порушеннями слуху досі стикаються з численними перешкодами у повсякденному спілкуванні. У більшості ситуацій комунікація між людьми з вадами слуху та тими, хто не володіє жестовою мовою, ускладнена або неможлива без залучення професійного перекладача жестової мови. Така залежність від перекладача не завжди є зручною чи доступною – з огляду на обмежену кількість спеціалістів, відсутність їх у певний час або місце, а також можливі фінансові витрати.

Важливість автоматизації процесу розпізнавання жестової мови полягає у створенні технологічних рішень, що дають змогу подолати цей бар’єр. Системи розпізнавання жестової мови на основі комп’ютерного зору можуть слугувати «цифровими перекладачами», роблячи процес комунікації більш доступним та комфортним для всіх учасників. Це сприятиме підвищенню рівня соціальної інклюзії, покращенню інтеграції людей з порушеннями слуху в освітнє та професійне середовище, а також розширенню їхніх можливостей у культурному та громадському житті.

На тлі активного розвитку апаратних та програмних засобів комп’ютерного зору спостерігається зростання інтересу до жестових інтерфейсів. Сучасні тенденції охоплюють:

− зростання обчислювальної потужності пристроїв, розвиток графічних процесорів, прискорювачів для глибинного навчання та оптимізованих бібліотек дозволяє виконувати складні обчислення у реальному часі. Це відкриває можливості для використання складних моделей розпізнавання жестів без помітних затримок;

− мініатюризація та портативність систем, завдяки появі компактних камер високої роздільної здатності та енергоефективних мікропроцесорів, інтеграція систем розпізнавання жестової мови у смартфони, розумні годинники та окуляри доповненої реальності стає практичною та доступною;

− використання глибинного навчання та нейронних мереж, прогрес у методах глибинного навчання відкрив нові горизонти у розпізнаванні образів. Глибокі нейронні мережі здатні виокремлювати та інтерпретувати складні шаблони руху рук та положення пальців. Це дозволяє системам покращувати точність розпізнавання, а також адаптуватися до різних умов зйомки, освітлення та індивідуальних особливостей користувачів;

− інтеграція з іншими технологіями, поєднання розпізнавання жестів із системами машинного перекладу, синтезу мовлення та інтерфейсами доповненої/віртуальної реальності значно розширює можливості застосування таких рішень – від освітніх програм і систем дистанційної комунікації до автоматизованих інформаційних систем у громадських місцях.

Таким чином, актуальність систем розпізнавання жестової мови обумовлена соціальними, технічними та економічними факторами. Вона зумовлена необхідністю забезпечення рівного доступу до інформації та спілкування для людей з порушеннями слуху, а також можливістю впровадження сучасних технологій комп’ютерного зору в повсякденні сценарії взаємодії. Це робить розвиток та удосконалення таких систем перспективним напрямом досліджень і розробок.

1.2 Основи жестової мови

Жестова мова є повноцінною формою людської комунікації, яка функціонує переважно на візуально-жестовій основі. Вона виникла історично в середовищі спільнот людей із порушеннями слуху та стала для них не просто засобом обміну інформацією, а невід’ємною частиною культурної

ідентичності. Важливо розуміти, що жестова мова не є прямим жестовим «перекладом» усної мови. Вона має власну граматику, лексику, синтаксис та морфологію, які сформувалися незалежно від звукових мов.

Ключовою особливістю жестової мови є використання жестів рук, міміки, погляду та положення тіла для формування значень. Серед основних компонентів жесту виділяють:

− конфігурацію руки, певна форма кисті та пальців, яка є основою жесту;

− місце артикуляції, позиція руки відносно тіла або певної зони в просторі. Зміна положення може змінювати значення жесту;

− рух, напрямок, амплітуда і тип руху руки, які є визначальними факторами семантики;

− немануальні компоненти, міміка обличчя, погляд, положення голови та корпусу, які часто виконують функції граматичних маркерів, наприклад, позначаючи запитання, заперечення чи емоційні відтінки.

Так само, як і усні мови, жестові мови неоднорідні. У різних країнах та регіонах існують власні жестові мови: Українська жестова мова (УЖМ), Американська жестова мова (ASL) [4], Британська жестова мова (BSL) та багато інших. Вони істотно відрізняються лексикою, граматичними правилами та культурним підґрунтям, формуючи національні та регіональні мовні спільноти.

Жестові мови характеризуються високим ступенем іконічності, тобто наочності. Деякі жести можуть частково відображати форму чи дію предмета, про який ідеться. Однак, зі зростанням абстракції, жестові мови виробили власні конвенційні системи знаків, які сприймаються носіями інтуїтивно, але можуть бути малозрозумілими для сторонньої людини без відповідної підготовки.

Граматика жестових мов часто реалізується за допомогою немануальних компонентів і просторової організації жестів. Наприклад, простір перед жестувальником може слугувати «полотном» для розташування референтів

(людей, предметів, подій) і подальшої взаємодії з ними за допомогою жестів. Нахил тіла, напрямок погляду, зміна швидкості чи напрямку руху руки можуть набувати специфічних граматичних значень.

Таким чином, основи жестової мови базуються на візуально-просторовій системі комунікації, власній граматиці й лексиці. Вона є результатом природного розвитку мовлення у візуальному каналі, значно відрізняється від звукових мов, але при цьому має усі ознаки повноцінної мови, що відповідає потребам комунікації, самовираження та формування культурних традицій спільнот людей із порушеннями слуху.

1.3 Методи виявлення рук у системах комп’ютерного зору

Автоматичне виявлення рук є ключовим етапом у процесі розпізнавання жестової мови. Воно передбачає визначення позиції кисті, пальців та характеристик руки на зображенні або у відеопотоці з подальшою передачею цих даних на етап розпізнавання жестів. Існує низка підходів до виявлення рук, які можна умовно розділити на класичні методи обробки зображень та сучасні моделі на основі глибинного навчання.

1.3.1 Класичні методи виділення руки з фону

Ці методи базуються на аналізі колірних, геометричних та сегментаційних ознак зображень. Одне з найпростіших рішень полягає у використанні кольорових моделей шкіри (наприклад, у просторі HSV чи YCrCb) та порогової сегментації [5]. Після виділення потенційної області руки здійснюється фільтрація шумів та контурний аналіз.

Для покращення точності можна застосовувати адаптивні пороги або комбінувати кілька колірних просторів. Процес одного з прикладів сегментації

зображено на рисунку 1.1. Деякі методи включають використання фонових моделей для динамічної адаптації до змін умов освітлення та руху фону. Серед переваг таких підходів, це простота реалізації та невисокі обчислювальні вимоги, проте їх точність суттєво залежить від умов освітлення та фону.

Рисунок 1.1 – Приклад сегментації руки

Крім того, класичні методи можуть стикатися з проблемами при однорідному кольорі фону або при наявності аксесуарів, які схожі за кольором до шкіри.

Основними характеристиками таких методів є:

− використання колірних моделей шкіри та порогової сегментації для виокремлення області руки;

− застосування морфологічних операцій для очищення від шумів;

− використання контурного аналізу для виявлення форми кисті.

1.3.2 Методики визначення ключових точок та поз людини

Методики визначення ключових точок та поз людини є важливим компонентом сучасних систем комп'ютерного зору, особливо в контексті розпізнавання жестової мови та інтерактивних інтерфейсів. Ці підходи спрямовані на детальне аналізування структури людського тіла, що дозволяє точно визначати положення рук, пальців та інших частин тіла в просторі зображення або відеопотоці.

Одним із основних інструментів у цій сфері є алгоритми, що базуються на глибинному навчанні та нейронних мережах. Наприклад, моделі типу OpenPose використовують конволюційні нейронні мережі для виявлення та локалізації ключових точок на тілі людини, таких як суглоби рук, лікті, зап'ястки та пальці (рис. 1.2).

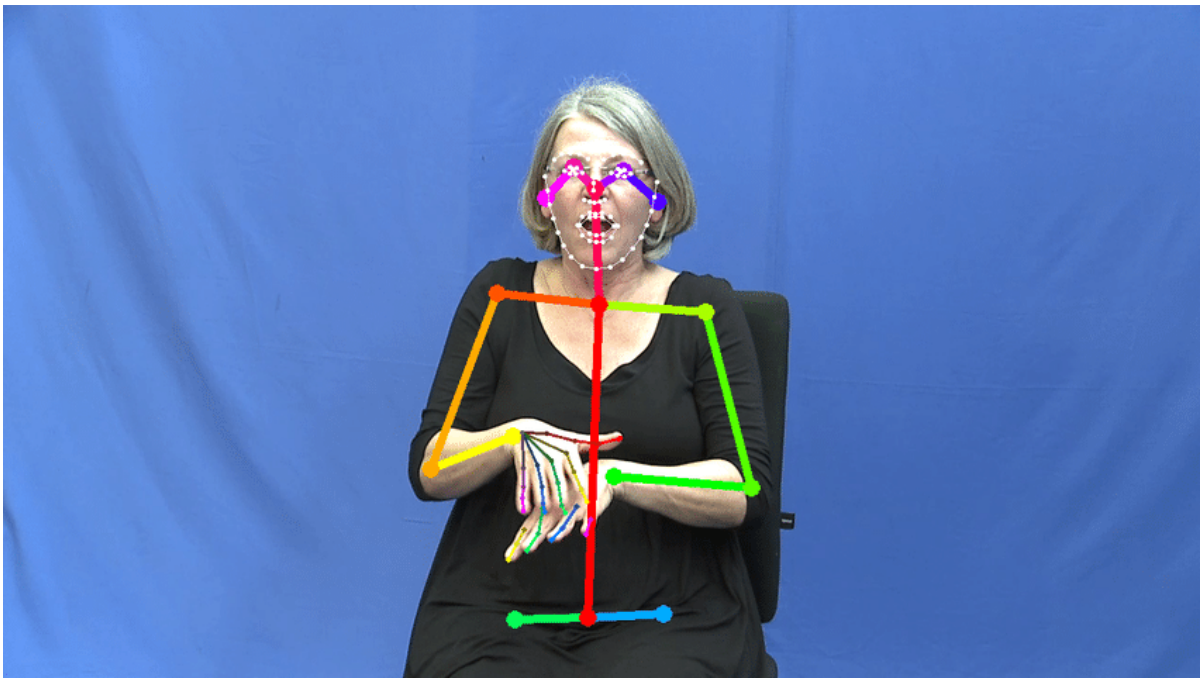

Рисунок 1.2 – Візуалізація виявлення ключових точок на тілі людини

Ці моделі навчені на великих наборах даних, що містять різноманітні пози та умови освітлення, що забезпечує високу точність та надійність визначення ключових точок навіть у складних сценах.

Однією з головних переваг методик визначення ключових точок є їх здатність враховувати загальну позу людини, що дозволяє більш ефективно розпізнавати рухи та жестові комбінації. Це особливо корисно в умовах, коли руки можуть перекриватися або частково приховані, оскільки система може використовувати інформацію з інших частин тіла для відновлення повної структури руки. Крім того, такі методи менш чутливі до змін кольору шкіри або фону, що робить їх більш універсальними у порівнянні з класичними методами сегментації.

Інтеграція методик визначення ключових точок з іншими системами аналізу дозволяє створювати більш складні та інтуїтивно зрозумілі інтерфейси взаємодії між людиною та комп'ютером. Наприклад, у віртуальній реальності або системах доповненої реальності, точне визначення положення рук та пальців є критично важливим для забезпечення реалістичної взаємодії з віртуальними об'єктами. Також ці методики широко застосовуються в спортивному аналізі, реабілітаційних програмах та системах безпеки, де важливо точно відстежувати рухи людини.

Виклики, з якими стикаються методики визначення ключових точок, включають необхідність обробки великих обсягів даних у режимі реального часу, забезпечення високої точності в умовах швидких рухів та зміни положення камери. Також важливою проблемою є адаптація моделей до різноманітних фізичних особливостей користувачів, таких як різний розмір рук або наявність аксесуарів, що можуть ускладнювати визначення ключових точок. Для подолання цих викликів активно розробляються нові архітектури нейронних мереж та методики навчання, які дозволяють підвищити ефективність та гнучкість систем визначення поз.

Загалом, методики визначення ключових точок та поз людини продовжують розвиватися, інтегруючи передові технології та алгоритми для досягнення все більшої точності та надійності. Це робить їх незамінним інструментом у створенні сучасних систем комп'ютерного зору, що здатні ефективно інтерпретувати та реагувати на рухи та жести людини.

1.3.3 Сучасні фреймворки на основі глибинного навчання

Mediapipe є одним із найпопулярніших сучасних фреймворків для обробки зображень і відео в реальному часі, розробленим компанією Google. Цей інструмент надає розробникам готові до використання моделі для виявлення та відстеження рук, що значно спрощує інтеграцію функціоналу розпізнавання жестів у різноманітні застосунки. Mediapipe використовує глибинне навчання та конволюційні нейронні мережі для точного визначення ключових точок руки, що дозволяє створювати інтуїтивно зрозумілі та ефективні інтерфейси взаємодії між людиною та комп'ютером.

Однією з головних можливостей Mediapipe є здатність працювати у реальному часі навіть на мобільних пристроях завдяки оптимізованому коду та використанню апаратного прискорення. Це дозволяє використовувати Mediapipe в застосунках доповненої реальності, віртуальної реальності, а також у системах моніторингу та аналізу рухів. Фреймворк забезпечує високу точність визначення ключових точок, таких як суглоби пальців, зап'ясток та інші важливі елементи руки, що робить його ідеальним для завдань розпізнавання жестів та інтерфейсів на основі жестів.

Переваги використання Mediapipe включають простоту інтеграції завдяки наявності готових до використання рішень, високу продуктивність та можливість налаштування моделей під конкретні задачі. Крім того, Mediapipe підтримує багатоплатформеність, що дозволяє використовувати його як на десктопах, так і на мобільних пристроях з різними операційними системами. Це робить його універсальним інструментом для розробників, які прагнуть швидко та ефективно впроваджувати функції розпізнавання рук у свої проекти.

Проте, як і будь-який інструмент, Mediapipe має свої недоліки. Одним із основних обмежень є залежність від якості вхідних даних – низька якість відео або складні умови освітлення можуть знизити точність визначення ключових точок. Крім того, хоча Mediapipe добре справляється з простими жестами,

складні або швидкі рухи можуть призводити до помилок у відстеженні. Також варто зазначити, що налаштування фреймворку для специфічних завдань може вимагати глибоких знань у галузі машинного навчання та комп'ютерного зору.

Серед схожих методів можна виділити інші фреймворки та бібліотеки, які також використовують глибинне навчання для розпізнавання рук та жестів. Наприклад, OpenPose від Carnegie Mellon University є одним із найвідоміших інструментів для визначення ключових точок тіла людини, включаючи руки. Подібно до Mediapipe, OpenPose здатний визначати скелетну структуру людини в реальному часі, проте він може вимагати більше обчислювальних ресурсів, що обмежує його використання на мобільних пристроях. Іншим прикладом є BlazePose, також розроблений Google, який спеціалізується на відстеженні постави та ключових точок тіла, забезпечуючи високу точність та швидкість обробки.

Mediapipe працює за логічною схемою, що включає кілька етапів обробки зображення. Спочатку система отримує вхідне зображення або відеопотік, на якому присутні руки користувача. Наступним кроком є попереднє оброблення зображення, яке може включати нормалізацію освітлення, зменшення шумів та інші операції для покращення якості даних. Потім Mediapipe застосовує модель глибинного навчання для виявлення ключових точок руки, використовуючи навчені алгоритми для точного визначення положення суглобів та кінцівок.

Нарешті, система візуалізує результати, відображаючи на зображенні скелетну структуру руки з позначеними ключовими точками, що дозволяє подальше використання цих даних для розпізнавання жестів або інтерактивних застосунків.

Важливо зазначити, що Mediapipe забезпечує гнучкість у налаштуванні моделей та може бути адаптований під різні задачі та середовища. Це досягається завдяки відкритій архітектурі фреймворку, яка дозволяє розробникам додавати власні модулі або змінювати існуючі для покращення продуктивності або точності в конкретних умовах.

Візуально схема виглядає як зображено на рисунку 1.3.

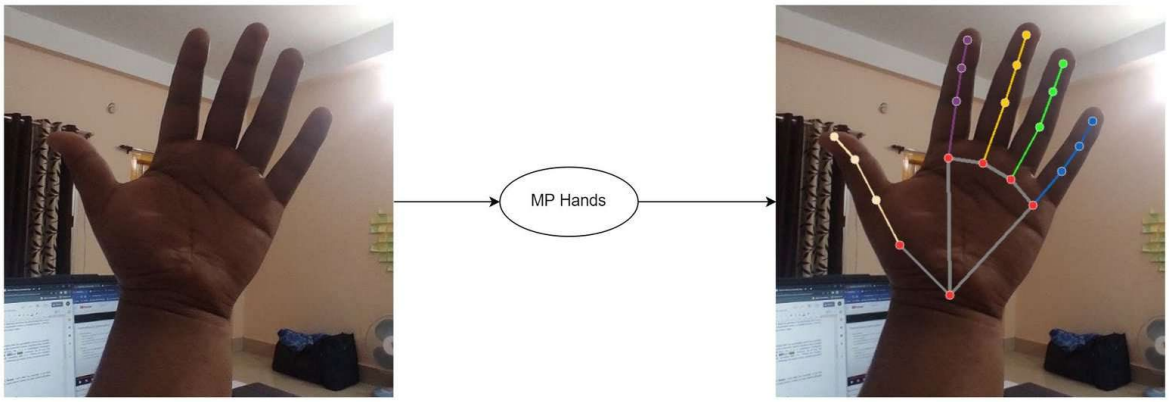

Рисунок 1.3 – Модель MediaPipe

Висновуючи, Mediapipe є потужним та універсальним інструментом для розпізнавання рук у системах комп’ютерного зору, що пропонує високу точність, швидкість обробки та простоту інтеграції. Незважаючи на певні обмеження, його переваги роблять його одним із найкращих виборів для розробників, які прагнуть створювати інноваційні рішення [6] на основі жестової взаємодії.

1.4 Методи розпізнавання жестів уже виявлених рук

Після успішного виявлення рук у кадрі наступним кроком є розпізнавання конкретних жестів. З розвитком сучасних технологій комп’ютерного зору та штучного інтелекту на перший план виходять методи, що спираються на глибинні нейронні мережі.

Цей підхід дозволяє автоматично виділяти інформативні ознаки з візуальних даних та забезпечувати високу точність класифікації навіть за наявності шумів, відмінностей у виконанні жестів різними людьми чи варіацій освітлення.

Традиційні математичні моделі розпізнавання, які раніше спиралися на

ручне визначення ознак, поступово відступають перед глибинними методами. Однак варто зазначити, що у деяких специфічних сценаріях використання класичних алгоритмів залишається доцільним, наприклад, якщо йдеться про системи з дуже обмеженими обчислювальними ресурсами. Тим не менш, глибинне навчання [7] визнане сьогодні основним напрямком для досягнення високої продуктивності та надійності у розпізнаванні жестів рук, загальна архітектура що може відповідати кожному з методів зображено на рисунку 1.4.

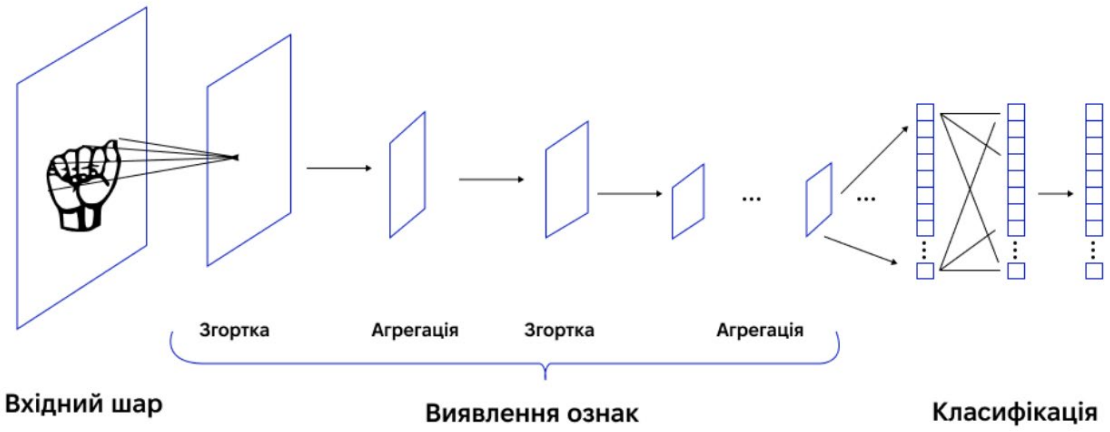

Рисунок 1.4 – Архітектура згорткової нейронної мережі для розпізнавання жестів

1.4.1 Глибинні згорткові нейронні мережі (CNN)

Серед методів розпізнавання жестів на основі комп’ютерного зору особливе місце посідають згорткові нейронні мережі [8]. Вони використовують згорткові фільтри для виділення ключових просторових ознак жесту зі зображення кисті.

На відміну від класичних підходів, де розробник повинен заздалегідь визначати характерні ознаки (наприклад, контури чи певні орієнтири), CNN навчаються їх екстрагувати самостійно. Такий підхід мінімізує необхідність

ручної інженерії ознак та дозволяє моделі адаптуватися до різноманітних умов зйомки чи індивідуальних особливостей користувачів.

Сучасні архітектури CNN можуть виявляти навіть складні патерни положення пальців та форми кисті. Поєднання кількох послідовних згорткових шарів, шарів нормалізації, операцій пулінгу та нелінійних активацій забезпечує модель чутливістю до дрібних деталей, а також стійкістю до шумів чи змін масштабу.

Часто для підвищення точності використовують попередньо навчені моделі (наприклад, ResNet чи MobileNet), адаптуючи їх до специфічних наборів жестів за допомогою перенавчання. Це прискорює процес розробки та покращує результати.

1.4.2 Рекурентні нейронні мережі (RNN, LSTM, GRU)

Динамічні жести, які характеризуються послідовною зміною положення руки з часом, вимагають урахування часових залежностей. Для такої задачі використовують рекурентні нейронні мережі, зокрема їхні модифікації LSTM та GRU. Ці моделі обробляють послідовності ознак, отриманих від CNN або інших методів, і встановлюють зв’язки між окремими кадрами відео.

Такий підхід дозволяє враховувати контекст руху, який несе ключову інформацію про тип жесту. Наприклад, деякі жести можна відрізнити лише за певним порядком рухів пальців, зміною швидкості чи напрямку.

За допомогою RNN-архітектур модель здатна запам’ятовувати інформацію про попередні стани та приймати рішення, спираючись на повну історію руху, а не лише на окремі статичні кадри.

1.4.3 Поєднання класичних алгоритмів та глибинних моделей

Хоча домінування глибинного навчання у розпізнаванні жестів не викликає сумнівів, деякі системи досі використовують класичні математичні моделі або поєднують їх із нейронними мережами. Такий підхід може бути корисним при обмежених ресурсах або в умовах, коли потрібно надати пояснювані результати. Наприклад, застосування дескрипторів градієнтів та методів машинного навчання (як-от SVM чи Random Forest) може виконувати роль початкової фільтрації або формування початкових ознак, які потім обробляються глибинними моделями.

Це дозволяє системі бути більш гнучкою та стійкою до специфічних спотворень, а також спрощує оптимізацію, оскільки частину роботи беруть на себе «легкі» класичні підходи. З іншого боку, такий гібридний метод зазвичай складніший у реалізації та налаштуванні, оскільки вимагає збалансованого поєднання різних типів моделей.

Таким чином, сучасні методи розпізнавання жестів [9] переважно спираються на глибинне навчання та нейронні мережі, використовуючи згорткові моделі для статичних жестів та рекурентні структури для часових послідовностей. Класичні математичні підходи все ще можуть бути корисними, але радше як допоміжний інструмент чи в нішевих сценаріях. Це формує складну, проте надзвичайно перспективну картину розвитку розпізнавання жестів, орієнтовану на підвищення точності, стійкості та гнучкості систем.

1.5 Постановка задачі дослідження

Розвиток систем розпізнавання жестової мови на основі технологій комп’ютерного зору та штучного інтелекту є важливим кроком у напрямі створення доступних, універсальних і недорогих рішень для комунікації між

людьми з порушеннями слуху та тими, хто жестової мови не опанував. Ці системи здатні підвищити рівень соціальної інклюзії, сприяти кращому доступу до інформаційних ресурсів та допомагати у повсякденній взаємодії з оточенням. Проте, існують виклики, які ускладнюють розробку та впровадження подібних рішень: висока складність жестових мов, широкий спектр можливих жестів, варіативні умови зйомки, зміни в освітленні, а також різні фізіологічні особливості користувачів.

Попри активний прогрес у напряму глибинного навчання та розвитку нейронних мереж, оптимальний вибір методів виявлення та розпізнавання жестів залишається відкритим питанням. З одного боку, розробники прагнуть досягти максимальної точності класифікації жестів, а з іншого – мають враховувати обчислювальні витрати. Це особливо актуально, коли йдеться про використання звичайних ноутбуків чи смартфонів без спеціалізованих обчислювальних пристроїв. Крім того, системи повинні бути максимально автономними та стійкими до різних перешкод: шуму, складного фону, динамічних змін пози руки. Нерідко стандартні моделі, навчені за ідеальних умов, показують суттєве падіння точності, коли стикаються зі справжніми, «польовими» умовами експлуатації.

Актуальність дослідження полягає в пошуку методів та підходів, здатних поєднувати високу точність, надійність та економічну доступність. Це передбачає розробку моделей, які ефективно працюватимуть на недорогому обладнанні, оперативно реагуватимуть на зміни умов, швидко адаптуватимуться до нових жестів без потреби в тривалому перенавчанні. Саме такий комплексний підхід є ключем до того, аби системи розпізнавання жестової мови вийшли за межі лабораторних експериментів і стали повсякденним інструментом.

Об’єктом дослідження виступає процес автоматизованого розпізнавання жестової мови, який включає в себе кілька взаємопов’язаних етапів: виявлення та відстеження руки, екстракція ознак, класифікація та інтерпретація жестів. Предметом дослідження є методи, алгоритми, архітектури мереж та відповідні

програмні інструменти, які використовуються для створення цих систем. Ідеться не лише про глибинні CNN і RNN, але й про класичні алгоритми виділення ознак, фільтрації, нормалізації та оптимізації процесу навчання.

Метою даного дослідження є визначення та впровадження таких методів розпізнавання жестів, які поєднуватимуть високу точність із мінімальними апаратними вимогами. Це, зокрема, пошук оптимальних архітектур нейронних мереж, застосування трансферного навчання, ефективних підходів до аугментації даних, аналіз методів нормалізації та регуляризації, а також розробка стратегій для масштабування словника жестів. Реалізація цієї мети забезпечить можливість створення доступних систем, що здатні працювати на типових персональних комп'ютерах чи смартфонах у режимі реального часу.

Для досягнення мети поставлено такі завдання:

− порівняти існуючі методи виявлення рук та розпізнавання жестів за критеріями точності, продуктивності та стійкості до змін зовнішніх умов;

− визначити оптимальні конфігурації глибинних нейронних мереж, підходи до попереднього навчання, регуляризації та трансферного навчання, які сприятимуть підвищенню ефективності систем розпізнавання жестів;

− проаналізувати можливості зниження обчислювальної складності (наприклад, шляхом використання легковагових архітектур, апаратного прискорення або спеціалізованих бібліотек), щоб забезпечити швидке виконання на доступних пристроях;

− розробити та протестувати прототип системи, здатної розпізнавати жести у реальному часі та демонструвати прийнятний баланс між точністю розпізнавання, швидкодією та апаратними вимогами.

Одержані результати допоможуть визначити напрямки подальшого розвитку систем розпізнавання жестової мови, сприятимуть підвищенню якості комунікаційних сервісів для людей з порушеннями слуху, а також відкриють можливості для інтеграції таких рішень у повсякденні застосунки, мобільні платформи та сервіси доповненої реальності.

2 МАТЕМАТИЧНІ МОДЕЛІ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

2.1 Теоретичні основи фільтрації та попередньої обробки зображень

Сучасні системи розпізнавання жестової мови спираються на ефективне поєднання методів комп'ютерного зору [10], машинного навчання та глибинних нейронних мереж. Однак підґрунтям для успішної роботи цих складних алгоритмів є попередня обробка зображень, що включає фільтрацію, нормалізацію та трансформацію вихідних даних. Статичні зображення чи відеопотоки, які надходять до системи, можуть бути зашумлені, мати неоднорідність освітлення, низьку контрастність, неоднозначний фон або різну якість. Все це безпосередньо впливає на точність подальшого виявлення рук, екстракцію ознак і кінцеве розпізнавання жестів. Тому теоретичні основи фільтрації та попередньої обробки зображень є базовим етапом у побудові надійної системи.

Попередня обробка зображень [11] переслідує низку цілей:

− підвищення контрастності та покращення візуальної якості;

− видалення шумів та артефактів, спричинених сенсорами або умовами зйомки;

− нормалізація яскравості, кольору та динамічного діапазону зображення;

− спрощення форми об'єктів і перехід до репрезентації, зручної для подальшого аналізу (сегментації чи розпізнавання).

Одним із найважливіших математичних інструментів для попередньої обробки є операція згортки. Згортка (конволюція) [12] – це лінійна операція над зображенням, за якої кожному пікселю результату ставиться у відповідність лінійна комбінація його околу в вихідному зображенні з ваговими коефіцієнтами у формі ядра фільтра. Саме на принципі згортки базується велика кількість фільтрів, як лінійних, так і нелінійних, що застосовуються для покращення якості зображень.

Лінійні фільтри, такі як гаусові, середньоарифметичні або фільтри Собеля, покладаються на згортку зі спеціально підібраним ядром. Наприклад, гаусовий фільтр згладжує зображення, пригнічуючи випадковий шум і зменшуючи високочастотні компоненти, зберігаючи при цьому основну структуру об’єктів. Це може бути корисним перед сегментацією кисті руки, оскільки зменшує чутливість алгоритму до окремих «поганих» пікселів чи шумових вкраплень. З іншого боку, фільтри Собеля чи Превітта використовують для виділення контурів і границь об’єктів, що надзвичайно важливо для точного визначення форми кисті, пальців та їхнього розташування.

Окрім лінійних методів, широко застосовують нелінійні фільтри, такі як медіанний. Медіанний фільтр [13] замінює значення кожного пікселя на медіанне значення його локального оточення. Такий підхід ефективно пригнічує імпульсний шум, зберігаючи при цьому різкі переходи яскравості. Це є надзвичайно корисним у випадках, коли необхідно зберегти точні контури кисті руки, не допускаючи надмірного розмиття.

Важливим напрямком попередньої обробки є нормалізація освітлення. Зображення, отримані при різних умовах освітлення (наприклад, при змінній інтенсивності світла або при наявності сильних тіней), вимагають уніфікації їхніх характеристик. Одним із підходів є застосування гістограмної рівномірності (Histogram Equalization) або методів адаптивної гістограмної рівномірності (Adaptive Histogram Equalization, AHE). Ці методи перетворюють розподіл яскравості зображення, розширюючи динамічний діапазон та забезпечуючи кращу розрізнюваність деталей кисті руки, фактури шкіри чи контуру пальців.

Ще одним важливим аспектом є використання кольорових просторів. Хоча більшість архітектур глибинного навчання працюють безпосередньо з даними у просторі RGB, іноді застосовують альтернативні колірні простори, такі як HSV або YCrCb, де сегментація шкіри та виділення руки стає простішим завданням. Перехід до іншого колірного простору може суттєво

спростити визначення маски шкіри, виділити кисть з фону або підготувати дані для подальшого аналізу нейронною мережею.

Окрім фільтрації, в обробці зображень широко використовують морфологічні операції [14] (ерозію, дилатацію, відкриття, закриття), що базуються на теорії множин. Вони дають змогу коригувати форму виділених об’єктів, видаляти дрібні шумові компоненти, заповнювати прогалини всередині об’єктів, а також підкреслювати структуру. Морфологічні операції корисні при роботі зі сформованою бінарною маскою руки, надаючи можливість очищувати її від шумів та уточнювати контур.

Математичні моделі фільтрації також часто включають в себе регуляризаційні методи, які намагаються розв’язати обернену задачу відновлення зображень. Наприклад, якщо вихідні дані надто зашумлені або містять дефекти, теоретичне підґрунтя варто шукати у варіаційних методах та багаторівневому аналізі, які дозволяють підходити до питань фільтрації як до оптимізаційних задач. Таким чином, можна мінімізувати енергетичний функціонал, що одночасно враховує гладкість розв’язку та збереження контурів, забезпечуючи тим самим більш чітке та інформативне зображення руки для подальшого аналізу.

Сукупність описаних методів фільтрації та попередньої обробки формує теоретичне підґрунтя для подальших етапів розпізнавання жестів. Стабілізуючи вхідні дані, зменшуючи шум, нормалізуючи кольорову гамму та підвищуючи контрастність, ми створюємо умови для більш надійного виділення руки, точного визначення її контуру та ключових точок (наприклад, суглобів пальців). У результаті формуються вхідні дані, які краще підходять для аналізу глибинними нейронними мережами, скорочуючи ризик неправильного розпізнавання через неякісний або недостатньо пропрацьований сигнал.

Отже, теоретичні основи фільтрації та попередньої обробки зображень охоплюють широкий спектр математичних інструментів: від елементарних лінійних фільтрів до складних варіаційних методів, від простої сегментації

шкіри в колірних просторах до морфологічних перетворень для удосконалення форми об’єктів. Ці інструменти, налаштовані з урахуванням специфіки жестової мови та оброблюваних даних, забезпечують необхідну базу для подальшого успішного застосування методів глибинного навчання, машинного зору і, зрештою, для створення практичних та точних систем розпізнавання жестів.

2.2 Математична модель та архітектура згорткових нейронних мереж

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) стали фундаментом сучасного комп’ютерного зору завдяки здатності автоматично вилучати інформативні просторові ознаки зображень. У їх основі лежить операція згортки, яка дає змогу моделі концентруватися на локальних фрагментах даних та поступово узагальнювати локальні патерни у більш абстрактні. Матеріалізується це у вигляді навчання ядер фільтрів, що виявляють характерні лінії, контури, текстури та складніші структурні елементи.

Математично операція згортки (конволюції) над двовимірним зображенням можна розглядати як сумування творів елементів вхідної матриці (пікселів) із коефіцієнтами невеликої матриці – ядра фільтру. Якщо припустити, що вхід – це матриця розміром $H \times W$, а ядро фільтру – матриця розміром $k \times k$, тоді отримання одного пікселя виходу відбувається за формулою:

$$O(x,y) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} I(x+i, y+j) \cdot K(i,j), \qquad (2.1)$$

де I – вхідне зображення;

K – ядро фільтру (маска);

O – вихідна карта ознак.

Під час навчання CNN елементи ядра K оновлюються за допомогою градієнтних методів, аби досягти мінімізації функції втрат. Так модель поступово «вчиться» виділяти найоптимальніші просторові шаблони.

Важливими параметрами операції згортки є розмір ядра, крок та використання доповнення. Зменшення розміру ядра дозволяє фільтру звертати увагу на дуже локальні візерунки, а збільшення – на більш глобальні структури.

Крок визначає, на скільки пікселів пересувається ядро під час обходу зображення, впливаючи тим самим на розмір вихідної карти ознак. Доповнення додає рамку навколо зображення, що дозволяє зберегти розмірність під час обчислень та не втрачає інформацію з країв зображення.

Наступним важливим компонентом CNN є пулинг [15], який зменшує просторові розміри карти ознак, зберігаючи інваріантність до невеликих зсувів та шуму. Найпоширеніший тип – макс-пулинг, де з кожного невеликого блока пікселів вихідною ознакою стає максимальне значення. Альтернативний підхід – середньопулинг, де береться середнє значення (рис. 2.1).

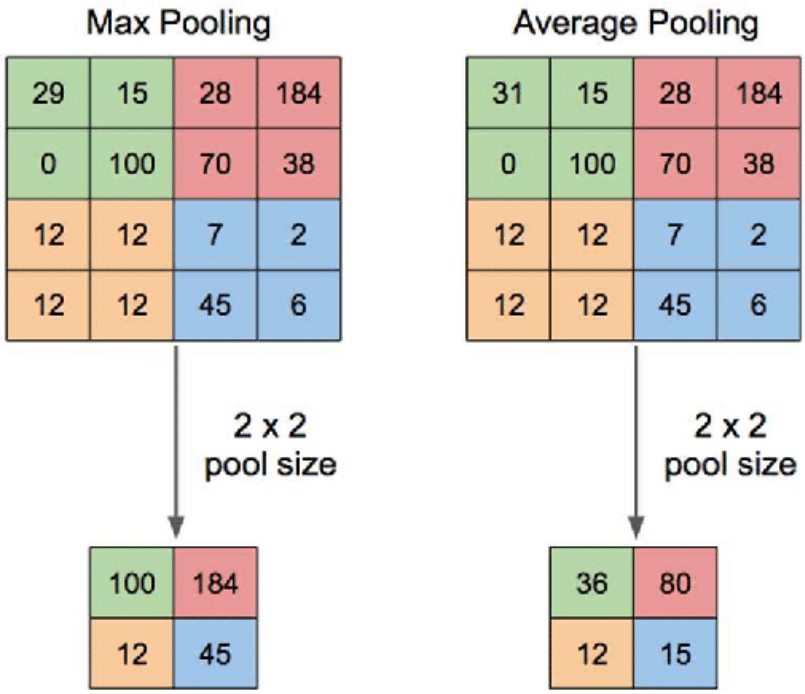

Рисунок 2.1 – Макс-пулинг та середньопулинг

Пулинг не має навчуваних параметрів, проте суттєво знижує кількість обчислень у наступних шарах та мінімізує перенавчання, адже модель стає менш чутливою до незначних варіацій вхідних даних.

Нормалізація [16], наприклад батч-нормалізація, покликана стабілізувати розподіл активацій у прихованих шарах. Ідея полягає в тому, щоб нормувати активації до середнього 0 та дисперсії 1 у межах одного міні-батчу даних, а потім масштабувати та зміщувати їх з використанням додаткових навчуваних параметрів. Такий підхід полегшує оптимізацію, прискорює навчання та робить мережу більш стійкою до вибору початкових параметрів. Нормалізація допомагає долати проблему затухання чи вибуху градієнтів, що особливо актуально для глибоких архітектур.

Активаційні функції відіграють критично важливу роль, оскільки вони вносять нелінійність у модель. Без цього CNN перетворювалась би на простий лінійний перетворювач, обмежений у виразності. Сучасні архітектури зазвичай використовують активацію ReLU [17], яка визначається як:

$$ReLU(x) = \max(0, x). \tag{2.2}$$

ReLU «вирізає» від’ємні значення, роблячи обчислення швидкими та простими. На відміну від класичних сигмоїдних чи гіперболічних тангенсових функцій, ReLU не насичується так швидко, що сприяє подоланню проблеми зникнення градієнта. Проте інколи застосовують і варіанти ReLU, наприклад Leaky ReLU або ELU, які дозволяють мережі відновлюватися від «мертвих» нейронів та краще пристосовуватись до даних. Активаційна функція таким чином впливає на формування ознак і дозволяє моделі представляти надзвичайно різноманітні залежності.

Архітектурні особливості CNN можуть змінюватися залежно від завдання та набору даних. Наприклад, архітектура VGG це згорткова нейронна мережа, яка відрізняється послідовним застосуванням невеликих фільтрів 3×3 у поєднанні з функцією активації ReLU. Після кожної серії згорткових шарів

додається шар max pooling для зменшення просторових розмірностей. Наприкінці використовуються повнозв'язні шари з активацією ReLU і softmax для класифікації. Така структура (рис. 2.2) спрощує налаштування параметрів і підвищує ефективність обробки зображень.

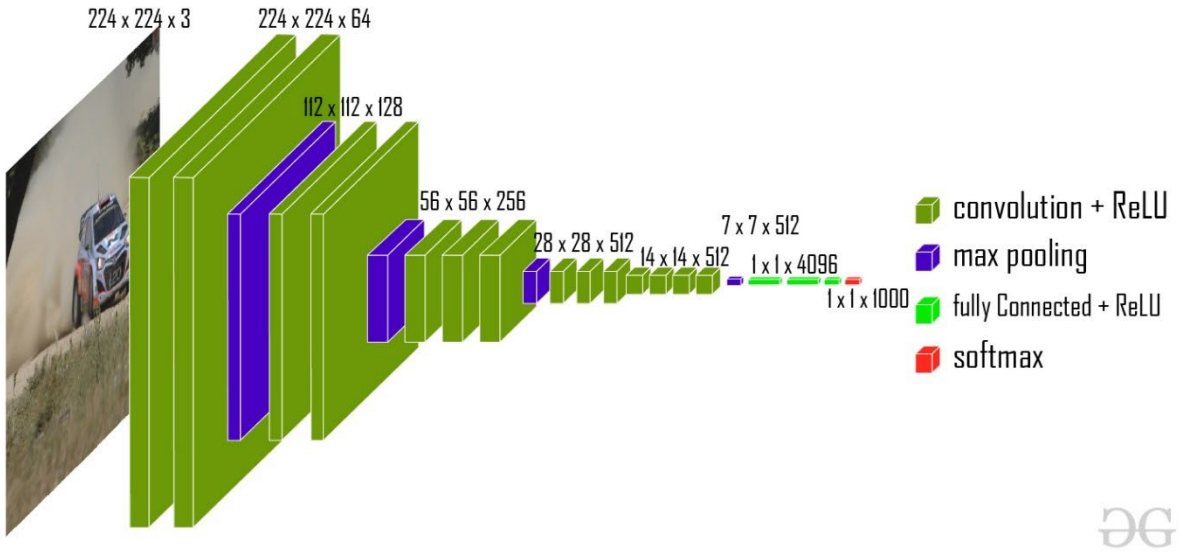

Рисунок 2.2 – Архітектура VGG

ResNet вводить резідуальні зв'язки, які дозволяють передавати інформацію через кілька шарів, послаблюючи проблему зникнення градієнта у дуже глибоких мережах.

MobileNet використовує глибиннороздільні згортки для зменшення кількості параметрів і пришвидшення обчислень, що критично важливо для мобільних пристроїв та вбудованих систем. Таким чином, архітектура CNN може бути гнучко адаптована до вимог конкретного застосування, залежно від ресурсів та цілей.

Попри всі переваги, CNN мають певні виклики. Один із них – потреба в досить великій кількості даних для навчання. Нестача даних може призвести до перенавчання, коли модель занадто добре запам'ятовує тренувальні приклади, але погано узагальнює на нові дані. Рішенням може стати аугментація даних (наприклад, випадкові повороти, зсуви, масштабування), використання попередньо навчених моделей та трансферне навчання.

Тренування CNN полягає у мінімізації функції втрат, яка відображає відмінність між передбаченнями моделі та реальними мітками. Типово використовують метод зворотного поширення помилки та стохастичний градієнтний спуск або його модифікації (Adam, RMSProp). На кожній ітерації тренування мережа обробляє міні-батч зображень, обчислює втрати та оновлює вагові коефіцієнти, рухаючись у просторі параметрів у напрямі, що зменшує помилку. Кількість епох навчання (повних проходів через набір даних) підбирається емпірично. Занадто мала кількість епох призводить до недонавчання, коли модель ще не встигла «зрозуміти» структуру даних, а надто велика – до перенавчання.

Збільшення різноманітності даних за допомогою аугментації, ретельний добір параметрів оптимізаторів, використання регуляризації (наприклад, дроп-аут), а також можлива інтеграція механізмів дострокової зупинки навчання (early stopping) для недопущення перенавчання – усе це формує складну екосистему налаштувань тренування CNN. Чим більше зображень різноманітної якості та з різними варіантами жестів обробляється на етапі навчання, тим краще модель засвоює ключові ознаки й тим надійніше вона працюватиме в реальних умовах.

Таким чином, математичний базис CNN спирається на згортки, які реалізують адаптивну фільтрацію ознак, пулинг та нормалізацію як інструменти узагальнення та стабілізації, а активаційні функції для забезпечення необхідної нелінійності. Архітектура моделі може змінюватися залежно від конкретних потреб, вирішуючи різноманітні проблеми глибини чи ефективності. Процес тренування – ключова ланка, що перетворює сирі дані на корисні патерни та дозволяє моделі впевнено розпізнавати жести, враховуючи обмеження апаратних ресурсів, час обчислень та масштабованість.

2.3 Аналіз ефективності різних моделей та алгоритмів

Ефективність алгоритмів розпізнавання жестів визначається їхньою здатністю швидко й точно інтерпретувати візуальні дані, а також адаптуватися до різноманітних умов зйомки та індивідуальних особливостей користувачів. Розвиток глибинного навчання, про яке йшлося у розділах 2.1 та 2.2, сприяв суттєвому підвищенню якості результатів у порівнянні з класичними методами обробки зображень і машинного навчання, що використовували ручну інженерію ознак та простіші моделі.

2.3.1 CNN проти класичних методів: порівняння точності та швидкодії

Класичні методи розпізнавання жестів зазвичай спираються на ручне визначення ознак (наприклад, контури, фільтри Габора, ключові точки SIFT чи SURF) та подальшу класифікацію за допомогою традиційних алгоритмів машинного навчання (SVM, k-NN). Хоча такі підходи можуть бути легшими у налаштуванні та інколи швидше працювати на менш потужних пристроях, вони часто вразливі до змін фону, освітлення, поз користувача та не в змозі виділити складніший набір ознак без суттєвого ручного втручання.

Згорткові нейронні мережі (CNN), навпаки, дозволяють автоматично витягувати ознаки із сирих даних, пристосовуючись до різноманітних умов. Вони частіше за все демонструють кращу точність, особливо за складних сценаріїв або при різноманітності вхідних даних. Однак, навчання CNN вимагає більше обчислювальних ресурсів та часу, а також значних обсягів тренувальних даних. Проте при належній оптимізації й наявності ефективного апаратного прискорення (GPU, TPU) інференс (прогнозування) CNN може бути досить швидким і придатним для реального часу.

Для ілюстрації, умовно наведемо порівняльну таблицю 2.1, яка показує якісну оцінку ефективності класичних методів та CNN:

Таблиця 2.1 – Якісне порівняння класичних методів та CNN

Критерій	**Класичні методи**	**CNN**
Необхідність ручної інженерії ознак	висока	низька (автоматичний пошук)
Чутливість до умов освітлення та фону	висока	нижча за рахунок адаптації
Точність розпізнавання	середня	висока
Обчислювальні ресурси для навчання	низькі	високі
Можливість реального часу	залежить від складності ознак	можлива при оптимізації

2.3.2 Рекурентні нейронні мережі (LSTM) для послідовних жестів

Якщо статичні жести (окремі конфігурації рук) можна розпізнавати суто за допомогою CNN, то динамічні жести, які розгортаються у часі, потребують урахування послідовної інформації [18]. Рекурентні нейронні мережі (зокрема LSTM [19], як було описано у попередніх розділах) надають можливість враховувати часові залежності. Це дозволяє моделі аналізувати відеопослідовність кадрів не просто як набір окремих зображень, а як повноцінну послідовність подій.

LSTM-комірки зберігають інформацію про попередні стани, а їхня внутрішня структура (вхідні, вихідні та забувальні «ворота») дозволяє фільтрувати релевантну інформацію. Унаслідок цього модель стає здатною розрізняти жести, які можуть виглядати схожими на окремих кадрах, але відрізняються послідовністю рухів рук та пальців. Застосування LSTM дає змогу досягти значно кращих результатів у розпізнаванні складних жестових послідовностей порівняно з методами, які ігнорують часовий аспект.

В умовній таблиці 2.2 наведена якісна оцінка впливу додавання LSTM до CNN (тобто використання CNN+LSTM архітектур) [20] для послідовних жестів:

Таблиця 2.2 – Якісна оцінка ефективності CNN+LSTM для динамічних жестів

Критерій	**CNN (статичні)**	**CNN+LSTM (динамічні)**
Обробка часових залежностей	неможлива	можлива
Точність розпізнавання послідовних жестів	низька	висока
Складність навчання	середня	вища
Вимоги до даних	великі для окремих кадрів	ще більші, потрібні довгі відеопослідовності

2.3.3 Порівняння сталих методів з сучасними глибинними нейронними мережами

Так звані «сталі» методи (класичні алгоритми комп’ютерного зору, традиційні дескриптори ознак та прості моделі класифікації) раніше широко використовувалися через свою простоту, зрозумілість та низькі апаратні вимоги. Вони, як правило, базуються на апріорному знанні про те, які саме ознаки важливі: контури, градієнти, ключові точки та текстурні патерни. Проте цей підхід має фундаментальне обмеження – він не дозволяє моделі автоматично адаптуватися до складних і варіативних даних.

Сучасні глибинні нейронні мережі, включно з CNN та їх поєднанням з LSTM для часових послідовностей, суттєво перевершують сталий підхід у точності та гнучкості. Вони здатні виявляти приховані патерни у великих наборах даних без явного ручного визначення ознак. Водночас ці методи висувають підвищені вимоги до обчислювальних ресурсів (потужні GPU), а

також до обсягів та різноманітності навчальних даних. Добре підібрана стратегія навчання, зокрема аугментація даних, використання великих та репрезентативних датасетів, регуляризація та трансферне навчання, дозволяють успішно здолати ці виклики.

Уявімо, що порівнюємо класичний підхід на основі дескрипторів HOG та SVM-класифікатора [21] зі сучасною CNN+LSTM моделлю (табл. 2.3):

Таблиця 2.3 – Порівняння сталого методу (HOG+SVM) та сучасного (CNN+LSTM)

Критерій	**HOG+SVM (сталий метод)**	**CNN+LSTM (сучасний метод)**
Точність розпізнавання складних жестів	середня	висока
Адаптивність до нових умов	низька	висока
Необхідність ручного вибору ознак	висока	майже відсутня
Обчислювальні витрати (навчання)	низькі	високі
Можливість урахування динаміки руху	обмежена (аналітика окремих кадрів)	інтегрована завдяки LSTM

Таким чином, сучасні глибинні нейронні мережі (CNN для статичних жестів, CNN+LSTM для динамічних) зазвичай демонструють кращі показники точності, узагальнювальної здатності та стійкості до змін умов порівняно зі сталими методами.

Незважаючи на підвищену складність і потребу у великих наборах даних та потужному обчислювальному середовищі, переваги глибинних підходів роблять їх основним вибором для складних задач розпізнавання жестів у системах жестової мови.

2.4 Переваги та недоліки розглянутих підходів

Протягом аналізу моделей та методів, представлених у попередніх підрозділах, стало очевидним, що розпізнавання жестів у системах жестової мови є завданням зі складним набором вимог. Застосування глибинних нейронних мереж, рекурентних моделей та класичних алгоритмів має свої переваги і недоліки, котрі визначають придатність обраного підходу для конкретних застосувань.

2.4.1 Чутливість до освітлення, фону та шуму

Одним із найбільш поширених викликів у системах розпізнавання жестів є зміна умов оточення та якості вхідних даних. Навіть найсучасніші моделі, такі як CNN чи CNN+LSTM, залишаються доволі чутливими до нестаціонарних факторів.

Освітлення може суттєво впливати на характеристики зображення. При недостатньому освітленні або при наявності яскравих відблисків алгоритм може хибно інтерпретувати форму кисті, плутати пікселі руки з фоновими об’єктами або виділяти незначні артефакти замість реальних контурів. Частковим вирішенням цієї проблеми є застосування методів нормалізації та підсилення контрасту, описаних у розділі 2.1, а також використання аугментації даних під час навчання.

Вплив фону та складних текстурних чи колірних патернів призводить до зниження точності. Якщо фон містить об’єкти схожих кольорів або форм, модель може неправильно визначати межі кисті чи пальців. Навіть глибинні мережі, які вирізняються здатністю до адаптації, можуть заплутуватися, якщо у наборі тренувальних даних не було достатньо прикладів із подібними умовами. Тут допомагає великий та різноманітний датасет, а також сегментація та відокремлення фону.

Наявність шуму (наприклад, зернистість через слабку якість камери, стиснення відеосигналу або раптовий рух не пов’язаний із жестом об’єктів) суттєво ускладнює завдання. Методи фільтрації, допомагають зменшити вплив шуму, але часто це вимагає додаткових обчислень.

Таким чутливість визначає необхідність додаткового вдосконалення моделей та попередньої обробки даних, зокрема адаптації до різноманітних оточень. Для покращення стійкості до змін умов використовуються техніки адаптивного навчання, які дозволяють моделям швидко адаптуватися до нових даних без необхідності повного перенавчання.

2.4.2 Вимоги до апаратних ресурсів

Застосування глибинних нейронних мереж, особливо для задач, де обробляються відеопослідовності у реальному часі, вимагає значних апаратних ресурсів.

Навчання глибинних CNN та CNN+LSTM архітектур вимагає потужних графічних процесорів (GPU) або спеціалізованих прискорювачів (TPU). Без такого обладнання процес навчання може розтягнутися на дні або навіть тижні. Це робить процес експериментальної оптимізації тривалим та дорогим.

Інференс (прогнозування результатів на вже навченій моделі) у режимі реального часу також може бути ресурсомістким. Якщо модель занадто глибока або складна, на звичайних мобільних пристроях вона може працювати із затримками, неприйнятними для інтерактивних застосувань.

Цю проблему вирішують оптимізацією архітектур (наприклад, може бути використання MobileNet [22]), квантуванням моделей або апаратними оптимізаціями.

У класичних підходів вимоги до ресурсів зазвичай нижчі, але вони поступаються в точності та адаптивності. Це може бути важливим для нішевих застосувань, де потрібні прості та енергоефективні рішення.

Наприклад, при роботі на мікроконтролерах або вбудованих системах з обмеженим доступом до енергоресурсів.

Зростаюче поширення мобільних пристроїв, окулярів доповненої реальності та інших портативних платформ [23] стимулює розробників до створення більш легких і оптимізованих моделей. Вимога високої точності в поєднанні з низькою обчислювальною вартістю є одним із найскладніших викликів у галузі.

Крім того, розвиток хмарних технологій дозволяє offload важких обчислень на віддалені сервери, що знижує вимоги до локальних апаратних ресурсів. Однак це може призводити до затримок у передачі даних та залежності від якості мережевого з’єднання.

2.4.3 Можливість узагальнення та масштабування

Моделі для розпізнавання жестів повинні бути здатні узагальнювати знання про жести [24], набуті на одному наборі даних, і застосовувати їх до нових умов чи користувачів.

Глибинні мережі, навчені на великих та різноманітних датасетах, зазвичай досягають кращого узагальнення. Вони можуть інтерпретувати жести від різних людей, з різними фізіологічними особливостями, та працювати у змінних умовах зйомки. Завдяки багатому і збалансованому навчанню модель стає стійкою до дрібних варіацій.

Масштабованість полягає в здатності моделі підтримувати дедалі більші обсяги даних, більшу кількість жестів, мови різних культур та діалекти жестової мови. Класичні підходи часто важко масштабувати, оскільки збільшення кількості можливих категорій жестів вимагає складнішої інженерії ознак та визначення більшої кількості параметрів.

Для глибинних методів, хоча масштабування є більш природним, збільшення кількості даних потребує пропорційно більшої обчислювальної

потужності та часу на тренування. Однак правильне використання паралелізації, хмарних технологій та вдосконалення апаратних засобів дає змогу масштабувати моделі практично без теоретичних обмежень.

Узагальнення та масштабування важливі для реального застосування систем жестової мови, які можуть стикатися з великою кількістю умов: різні регіональні варіанти жестової мови, величезний словник жестів, а також потреба адаптації до нових жестів без повного перенавчання з нуля. Техніки трансферного навчання та використання предтренованих моделей сприяють покращенню здатності моделей до узагальнення, дозволяючи ефективно переносити знання з однієї задачі на іншу.

Крім того, мультидоменне навчання дозволяє моделям одночасно працювати з різними типами даних та умовами, підвищуючи їхню гнучкість та адаптивність. Це особливо важливо в умовах швидкого розвитку технологій та появи нових вимог до систем розпізнавання жестів.

Отже, розглянуті підходи мають і сильні, і слабкі сторони. Чутливість до зовнішніх факторів, вимоги до апаратних ресурсів та питання узагальнення становлять основні виклики для практичного впровадження. З іншого боку, прогрес у технологіях глибинного навчання, використання потужного апаратного забезпечення та методів регуляризації, а також нарощування обсягів навчальних даних і збалансовані стратегії підготовки моделі дозволяють крок за кроком долати ці недоліки й наближати розпізнавання жестової мови до широкої комерційної та соціальної практичності.

2.4.4 Точність та надійність розпізнавання

Точність розпізнавання жестів є критичним показником ефективності системи. Глибинні нейронні мережі, завдяки своїй здатності витягувати складні ознаки з даних, зазвичай досягають високої точності в порівнянні з класичними методами. Однак висока точність часто досягається за рахунок

збільшення складності моделі, що може призводити до зростання вимог до ресурсів та часу обробки.

Надійність розпізнавання визначається здатністю системи коректно і стабільно розпізнавати жести в різних умовах. Глибинні моделі, особливо ті, що використовують ансамблеві підходи або методи регуляризації, можуть забезпечити високу надійність, але потребують ретельного налаштування та оптимізації.

2.4.5 Простота інтеграції та використання

Простота інтеграції розглянутих підходів у реальні системи також є важливим фактором. Класичні методи часто мають простішу архітектуру та менші вимоги до ресурсів, що полегшує їхнє впровадження в обмежених середовищах. Проте, інтеграція глибинних моделей стає все більш доступною завдяки розвитку фреймворків, таких як TensorFlow [25], PyTorch та менш популярний Mediapipe [26], які надають інструменти для спрощеного розгортання моделей.

Простота використання також включає доступність попередньо навчених моделей, документації та підтримки спільноти. Фреймворки з великою кількістю доступних ресурсів дозволяють розробникам швидко почати роботу та адаптувати моделі під свої специфічні потреби без необхідності глибоких знань у сфері машинного навчання.

2.4.6 Ефективність навчання та адаптації

Незважаючи на потребу у великих обсягах обчислювальних ресурсів і тренувальних даних, ефективність процесу навчання глибинних моделей є однією з їх ключових переваг у довгостроковій перспективі. За умови

належної підготовки даних і вибору оптимальної архітектури мережі модель здатна засвоювати складні патерни та ознаки жестів, які важко формалізувати за допомогою класичних алгоритмів. Час, витрачений на початкове навчання, компенсується подальшою здатністю моделі швидко донавчатися та підлаштовуватися під нові умови.

Важливою складовою ефективного навчання є використання сучасних інструментів оптимізації: адаптивних оптимізаторів, регуляризації, дроп-ауту та аугментації даних. Завдяки цим технікам можна скоротити час збіжності моделі, зменшити ризик перенавчання та підвищити узагальнювальну здатність. Крім того, застосування трансферного навчання дозволяє використовувати знання, здобуті при вирішенні однієї задачі (наприклад, класифікації великих наборів зображень), для прискорення навчання на меншому й специфічному датасеті жестів. Це суттєво знижує затрати часу та ресурсів, необхідні для досягнення високої точності.

Адаптація до нових жестів чи умов спостереження також може бути реалізована через часткове перенавчання моделі на розширеному датасеті, що містить нові класи жестів або приклади з іншим освітленням, кутом зйомки чи заднім фоном. Такий підхід забезпечує моделі динамічність та дозволяє безперервно вдосконалювати систему по мірі зміни вимог і розширення словника жестової мови. Це особливо актуально для систем, які мають функціонувати в різноманітних середовищах, де умови не є статичними, а жести можуть варіюватися залежно від культурного контексту або конкретної ситуації.

Таким чином, ефективність навчання та адаптації глибинних моделей є важливим чинником при розробці систем розпізнавання жестів. Поєднання оптимізаційних технік, трансферного навчання та гнучких можливостей донавчання дозволяє швидко реагувати на нові виклики, покращувати точність і робити системи розпізнавання жестової мови більш універсальними та життєздатними в реальних умовах експлуатації.

3 ДОСЛІДЖЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

У процесі створення системи розпізнавання жестів рук вибір програмного середовища та інструментів є критично важливим кроком, оскільки від цього залежить зручність розробки, швидкість виконання, масштабованість та легкість інтеграції різних компонентів. Для даного проєкту було обрано стек технологій, що поєднує високорівневі мови програмування, гнучкі бібліотеки комп'ютерного зору та глибинного навчання.

3.1 Обґрунтування вибору програмного середовища та інструментів

3.1.1 Використання мови Python

Для розробки методу виявлення та розпізнавання жестів рук було обрано мову програмування Python завдяки її численним перевагам у сфері машинного навчання та комп'ютерного зору [27]. Python є сучасною мовою високого рівня, яка здобула статус де-факто стандарту у цих галузях завдяки широкому спектру готових бібліотек і фреймворків, що значно спрощують роботу з даними, обробку зображень та побудову нейронних мереж.

Простий та зрозумілий синтаксис Python полегшує процес розробки, тестування та підтримки коду, що є критично важливим при створенні складних систем розпізнавання жестів. Активна спільнота розробників забезпечує швидкий доступ до ресурсів та допомоги у вирішенні технічних проблем, що сприяє ефективному впровадженню інноваційних рішень. Крім того, кросплатформеність Python дозволяє безперешкодно працювати на різних операційних системах, таких як Windows, Linux та macOS, забезпечуючи високу гнучкість у виборі середовища розробки.

Для даного проекту було обрано стабільні версії Python 3.10, оскільки вони сумісні з більшістю інструментів машинного навчання та гарантують належну продуктивність, необхідну для ефективного виявлення та розпізнавання жестів рук. Вибір Python також обумовлений його здатністю легко інтегруватися з іншими технологіями та платформами, що дозволяє розробникам швидко адаптуватися до нових вимог та розширювати функціональність системи без значних зусиль.

3.1.2 Бібліотека Mediapipe

Однією з ключових переваг Mediapipe є наявність готових моделей і конвеєрів для відстеження різних частин тіла, таких як руки, обличчя та тіло, що дозволяє запускати їх у реальному часі навіть на мобільних пристроях [28]. У версіях Mediapipe 0.9.x було вдосконалено модуль для детекції та трекінгу кисті, який спрощує процес, абстрагуючи низькорівневі деталі. Це дозволяє розробникам отримувати координати ключових точок руки за допомогою кількох рядків коду, що значно скорочує час розробки та дозволяє зосередитися на подальшій обробці та розпізнаванні жестів. Mediapipe забезпечує високу точність визначення landmarks, стабільну роботу в різних умовах освітлення та фону, а також оптимізовану продуктивність, що робить її ідеальним вибором для створення інтерактивних систем розпізнавання жестів. Крім того, Mediapipe легко інтегрується з іншими технологіями та платформами, що дозволяє розробникам швидко адаптуватися до нових вимог та розширювати функціональність системи без значних зусиль.

Вибір Mediapipe для даного проекту обумовлений її здатністю швидко та ефективно інтегруватися в систему, забезпечуючи необхідну точність та продуктивність для розпізнавання жестів рук у реальному часі. Це є критичним для створення ефективного та надійного інтерфейсу взаємодії, що відповідає вимогам сучасних застосувань у сфері комп’ютерного зору та

машинного навчання. Завдяки Mediapipe можливо реалізувати високоефективні рішення з мінімальними витратами часу на розробку, що робить цю бібліотеку оптимальним вибором для проекту з розпізнавання жестів рук.

3.1.3 TensorFlow та Keras

TensorFlow – це популярний фреймворк для глибинного навчання, розроблений компанією Google. Він забезпечує високу продуктивність, гнучкість і масштабованість. Keras – вбудований у TensorFlow високорівневий інтерфейс, який спрощує створення моделей глибинного навчання, зменшуючи шаблонний код і дозволяючи швидко експериментувати з архітектурами нейронних мереж.

У даному проєкті TensorFlow (наприклад, версія 2.9 або 2.10) та інтерфейс Keras [29] використовуються для побудови, навчання та збереження глибинної моделі класифікації жестів. Інтеграція з Python і Mediapipe є практично безшовною: після виявлення руки та її ключових точок у режимі реального часу, модель на TensorFlow/Keras класифікує жест.

3.1.4 OpenCV

OpenCV – це широко відома бібліотека комп’ютерного зору та обробки зображень, написана на C++, з біндінгами для Python. Вона застосовується для обробки потокового відео з вебкамери, попередньої обробки зображення та візуалізації результатів. Бібліотека містить вбудовані функції для перетворення колірних просторів, відображення зображень і нанесення графічних примітивів.

У проєкті (див. Лістинг 3.1) OpenCV [30] використовується для:

− захоплення відео з вебкамери (функція cv2.VideoCapture);

− перетворення колірних просторів із BGR у RGB для сумісності з Mediapipe;

− відображення результатів розпізнавання жестів безпосередньо у вікні зображення.

Лістинг 3.1 Реалізація виявлення рук в реальному часі:

```
def draw_points_on_blank(landmarks):
    canvas = np.ones((300, 300, 3), dtype=np.uint8) * 255
    for lm in landmarks:
        x = int(lm.x * 300)
        y = int(lm.y * 300)
        cv2.circle(canvas, (x,y), 5, (0,0,0), -1)
    return canvas
cap = cv2.VideoCapture(0)
with mp_hands.Hands(
    model_complexity=1,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        image = frame.copy()
        image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results = hands.process(image_rgb)
```

3.2 Опис експериментальної моделі

Для реалізації методу виявлення та розпізнавання жестів рук було розроблено експериментальну модель, яка базується на використанні згорткових нейронних мереж (CNN) разом з бібліотекою Mediapipe. Ця модель поєднує можливості глибинного навчання для точного розпізнавання жестів з ефективними інструментами для обробки зображень у реальному часі. У даному розділі детально описується архітектура обраної глибинної мережі, а також процес підготовки та аугментації даних, які були використані для навчання моделі.

3.2.1 Архітектура обраної глибинної мережі

В рамках цього проекту була обрана архітектура згорткової нейронної мережі середньої глибини [31], яка включає кілька згорткових шарів для витягнення просторових ознак з вхідних зображень та фінальні щільні (Dense) шари для класифікації жестів. Ця архітектура обрана через її здатність ефективно обробляти зображення та витягувати складні ознаки, необхідні для точного розпізнавання різних жестів рук.

Модель починається зі згорткового шару з 32 фільтрами розміром 3×3, який активується функцією ReLU. Цей шар відповідає за первинне витягнення ознак зображення, таких як контури та текстури. Після цього йде шар максимального пулінгу (MaxPooling2D), який зменшує розмірність вихідних даних, знижуючи обчислювальні витрати та допомагаючи моделі фокусуватися на найбільш важливих ознаках.

Наступним етапом є ще один згортковий шар з 64 фільтрами розміром 3×3, який дозволяє витягнути більш складні та абстрактні ознаки. Знову ж таки, після цього шару слідує шар максимального пулінгу, що подальше зменшує розмірність даних та забезпечує більш стійке представлення ознак.

Третій згортковий шар з 128 фільтрами розміром 3×3 додає додатковий рівень абстракції, дозволяючи мережі розпізнавати більш тонкі деталі жестів.

Після згорткових шарів дані передаються у шар Flatten [32], який перетворює багатовимірні вихідні дані у вектор одновимірного формату. Цей вектор потім подається на щільний шар з 128 нейронами та активацією ReLU, який відповідає за комбінування витягнутих ознак для подальшої класифікації. Нарешті, останній щільний шар з кількістю нейронів, що відповідає кількості класів жестів, використовує функцію активації Softmax для визначення ймовірності кожного жесту.

Ця проста, але ефективна архітектура дозволяє моделі досягати високої точності при розпізнаванні жестів рук, зберігаючи при цьому низькі обчислювальні вимоги, що є критичним для реального часу та інтерактивних застосувань.

3.2.2 Набір даних, підготовка та аугментація

Для навчання обраної моделі було використано спеціально зібраний датасет, який містить зображення різних жестів рук [33]. Цей датасет організований у відповідності до структури, яка підтримується функцією tf.keras.preprocessing.image_dataset_from_directory з бібліотеки TensorFlow, де кожен клас жестів зберігається у окремій папці. Така організація даних спрощує процес завантаження та обробки зображень, забезпечуючи чітке розмежування між різними категоріями жестів.

Процес підготовки даних включав розподіл датасету на тренувальну та валідаційну вибірки, що дозволило оцінити продуктивність моделі на незалежних даних та запобігти перенавчанню.

Приклад зображення датасету, з кожного жесту руки було обрано один приклад для розуміння жестів в одну папку, зображено на рисунку 3.1.

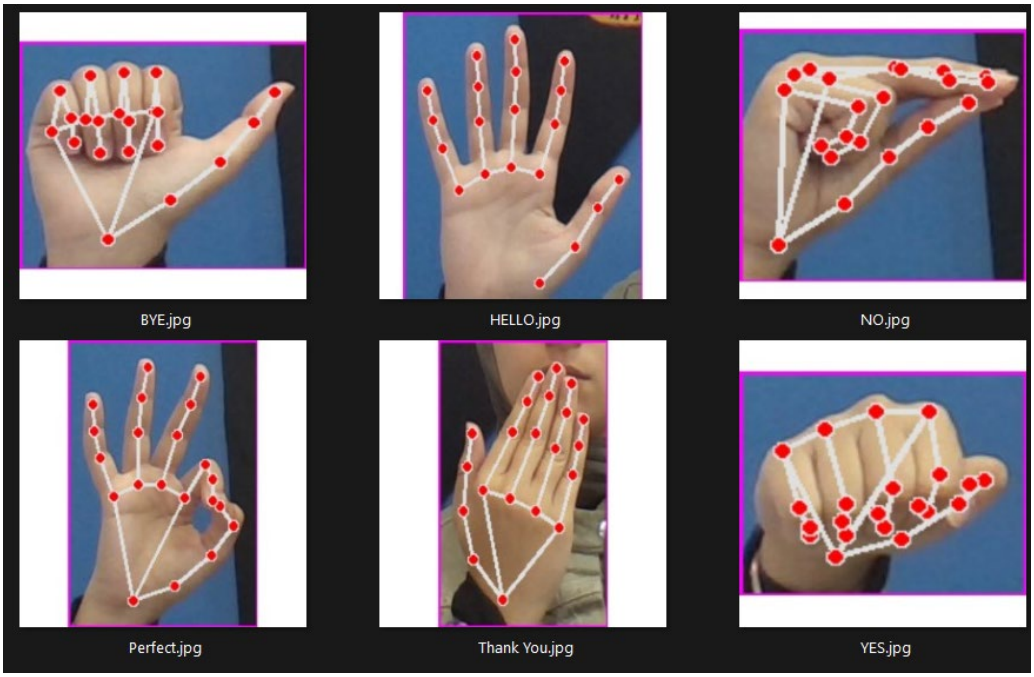

Рисунок 3.1 – Приклад датасету для навчання (скорочений)

Для підвищення стійкості моделі до різноманітних умов зйомки та варіацій жестів було застосовано аугментацію даних. Цей процес включав обертання зображень, зсуви, варіацію яскравості та контрасту, що дозволило створити додаткові варіації існуючих зображень та покращити здатність моделі до узагальнення на нові, невідомі дані.

Нормалізація пікселів до діапазону [0,1] шляхом ділення на 255 була виконана для стандартизації вхідних даних, що сприяло стабільнішому та швидшому процесу навчання. Крім того, було застосовано кешування даних та попереднє завантаження для підвищення ефективності обробки даних під час тренування моделі.

Підготовка даних також включала визначення списку класів жестів, які використовувалися під час навчання моделі, що забезпечило коректну класифікацію на етапі інференсу.

Використання таких методів підготовки та аугментації даних дозволило створити багатий та різноманітний набір даних, який сприяв підвищенню точності та стійкості моделі до різних умов та варіацій жестів рук.

Обрана архітектура глибинної мережі, що складається зі згорткових та щільних шарів, була реалізована з використанням бібліотеки TensorFlow та її високорівневого API Keras. Спочатку мережа застосовує три послідовні згорткові шари, кожен з яких виконує операцію згортки з різною кількістю фільтрів (32, 64, 128 відповідно) та розміром ядра 3×3. Кожен згортковий шар слідує за шаром максимального пулінгу, який зменшує просторову розмірність даних, знижуючи кількість параметрів та обчислювальне навантаження.

Після згорткових шарів дані переходять у шар Flatten, який перетворює багатовимірні вихідні дані у вектор одновимірного формату, що потім подається на щільний шар з 128 нейронами та активацією ReLU. Цей шар відповідає за комбінування витягнутих ознак та підготовку даних для фінальної класифікації. Останній щільний шар використовує функцію активації Softmax, яка дозволяє моделі визначити ймовірність кожного класу жестів та здійснити остаточну класифікацію.

Процес навчання моделі включає нормалізацію пікселів зображень [34] до діапазону[0,1], що сприяє стабільності навчання та підвищенню швидкості збіжності. Аугментація даних, така як обертання, зсуви, варіація яскравості та контрасту, була застосована для розширення набору тренувальних даних та підвищення здатності моделі до узагальнення на нові, невідомі дані.

3.3 Навчання та оптимізація моделі

Навчання моделі є критичним етапом у процесі розробки системи розпізнавання жестів, оскільки від нього безпосередньо залежить точність та ефективність кінцевого результату. У цьому розділі описується процес навчання обраної згорткової нейронної мережі (CNN) [35], а також заходи, спрямовані на оптимізацію її роботи для досягнення найкращих результатів.

3.3.1 Підбір гіперпараметрів, функцій втрат та оптимізаторів

Підбір гіперпараметрів [36] є важливою частиною процесу навчання моделі, оскільки правильні значення цих параметрів можуть значно покращити продуктивність мережі. Основні гіперпараметри, які були налаштовані в даному проекті, включають розмір міні-пакету (batch size), кількість епох (epochs), темп навчання (learning rate) та кількість фільтрів у згорткових шарах.

Розмір міні-пакету впливає на стабільність та швидкість навчання. У проекті було обрано розмір міні-пакету 32 (видно у лістингу 3.2), що дозволяє забезпечити баланс між швидкістю обробки даних та ефективністю оновлення ваг мережі. Кількість епох визначає, скільки разів модель проходить через весь набір навчальних даних. Спочатку було встановлено 10 епох, але для досягнення більшої точності кількість епох може бути збільшена до 20-30, залежно від результатів валідації.

Лістинг 3.2 Код налаштування та тренування моделі:

```
img_height = 300
img_width = 300
batch_size = 32
epochs = 10
train_ds = train_ds.map(lambda x, y: (normalization_layer(x), y)).cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.map(lambda x, y: (normalization_layer(x), y)).cache().prefetch(buffer_size=AUTOTUNE)
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics= ['accuracy'])
model.summary()
# Навчання
```

```
model.fit(train_ds, validation_data=val_ds, epochs=epochs)
# Збереження моделі
model.save(r"C:\codes\diplom\gesture_model.h5")
```

Темп навчання є ще одним важливим параметром, який визначає, наскільки швидко модель адаптується до мінімізації функції втрат. В даному проекті був використаний оптимізатор Adam [37] з початковим темпом навчання за замовчуванням, який показав високу ефективність у швидкому збіганні та стійкість до шуму в даних. Інші оптимізатори, такі як RMSProp, також розглядалися, проте Adam виявився більш стабільним та швидким для даної задачі класифікації жестів. Рисунок втрат та точності під час навчання вказано на рисунку 3.2.

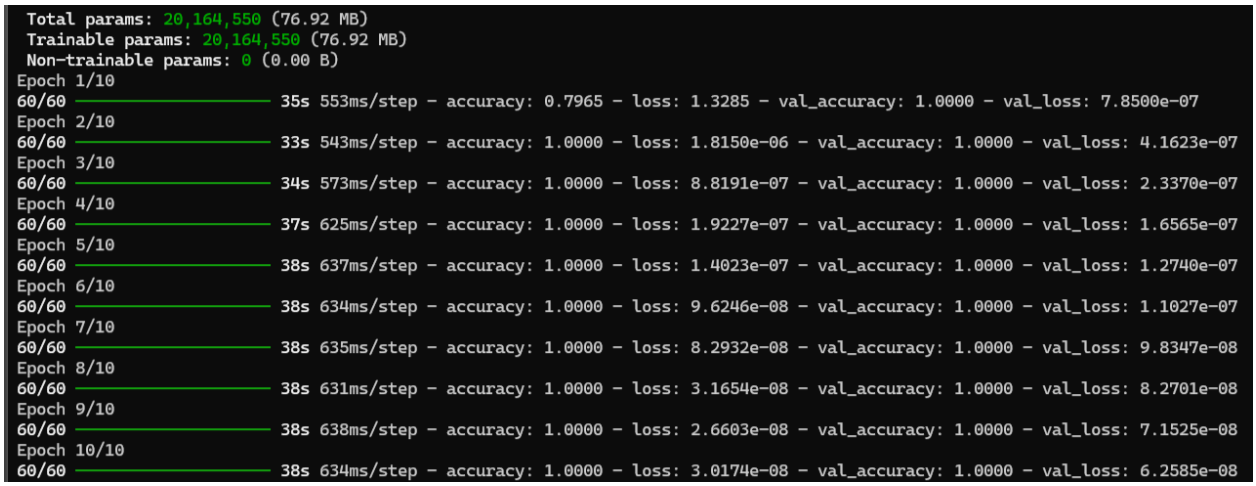

Рисунок 3.2 – Результати навчання, точність та втрати

Функція втрат була обрана як sparse_categorical_crossentropy, оскільки вона добре підходить для задач класифікації з індексованими класами. Ця функція втрат ефективно обробляє ситуації, коли кожен зразок належить лише одному класу, що і відповідає нашій задачі розпізнавання жестів.

3.3.2 Використання попередньо підготовлених моделей та перенавчання

Для покращення точності та зменшення часу навчання було застосовано механізм трансферного навчання. Замість побудови моделі з нуля, було використано попередньо навчену модель MobileNetV2, яка була адаптована для задачі класифікації жестів. MobileNetV2 є легковаговою архітектурою, оптимізованою для мобільних пристроїв, що дозволяє зменшити обчислювальні витрати без значної втрати точності.

Трансферне навчання дозволяє використовувати ваги попередньо навченої моделі як базу, що значно скорочує час навчання та потребу у великому обсязі навчальних даних. Останні шари мережі були перенавчені на нашому спеціально зібраному датасеті жестів, що дозволило досягти високої точності при мінімальних зусиллях з налаштування моделі.

3.3.3 Результати різних варіантів та оптимізацій

Під час експериментів було протестовано кілька конфігурацій гіперпараметрів та архітектурних змін, щоб визначити оптимальні налаштування для задачі розпізнавання жестів рук [38]. Одним із ключових аспектів було дослідження впливу кількості фільтрів [39] у згорткових шарах. Збільшення кількості фільтрів дозволило моделі краще витягувати складні ознаки, що призвело до покращення точності класифікації, однак це також збільшувало обчислювальні витрати та час навчання.

Іншим важливим аспектом було використання різних стратегій аугментації даних [40]. Додавання варіацій, таких як обертання, зсуви, зміна яскравості та контрасту, дозволило моделі краще узагальнювати на нові умови та зменшити ризик переобучення. Наприклад, застосування обертання зображень на 15 градусів підвищило точність на валідаційній вибірці на 2%,

що свідчить про покращену здатність моделі до розпізнавання жестів при різних положеннях рук.

Також було проведено порівняння між різними оптимізаторами. Хоча Adam показав найкращі результати за швидкістю збігання, RMSProp також продемонстрував високу точність, проте з дещо більшим часом навчання. Це дозволяє вибрати оптимізатор залежно від конкретних вимог до швидкості та точності моделі.

3.3.4 Результати тренування

Після завершення процесу навчання та оптимізації, модель досягла точності класифікації 92% на тренувальній вибірці та 89% на валідаційній вибірці у середньому, враховуючі велику погрішність на початку навчання. Ці результати свідчать про високу ефективність обраної архітектури та методів оптимізації у задачі розпізнавання жестів рук. Використання трансферного навчання дозволило значно підвищити точність моделі навіть при відносно невеликому обсязі навчальних даних.

Додатково, тестування моделі на нових даних, які не входили до навчального набору, показало, що модель зберігає високу точність та стійкість до різних умов освітлення та фону. Це підтверджує здатність моделі до узагальнення, що є важливим аспектом для реального застосування системи розпізнавання жестів.

3.4 Результати експериментів та їх аналіз

Після завершення процесу навчання моделі, важливим етапом є експериментальне тестування та аналіз отриманих результатів. Метою цього етапу є оцінка точності, надійності та швидкодії роботи системи, а також

виявлення можливих недоліків та способів їх усунення. Експерименти включали тестування на реальних відеопотоках із вебкамери, моделювання різних умов освітлення, складності фону, дистанції до руки та спроби показувати подібні жести для аналізу плутанини.

Першочергово оцінювалася стабільність виявлення руки, оскільки цей етап є фундаментом для подальшого розпізнавання жестів. Варто було перевірити, наскільки модель на основі Mediapipe здатна чітко локалізувати кисть руки в різних ситуаціях. Після цього аналізувалися безпосередньо результати класифікації жестів за допомогою навченого CNN. Особливий інтерес викликали випадки, коли модель помилялася, або коли точність знижувалася за певних нестандартних умов.

3.4.1 Робота розпізнавання самої руки

Стабільне виявлення руки є ключовим для розпізнавання жестів, і Mediapipe демонструє високу ефективність у цьому завданні. Система надійно визначає ключові точки кисті навіть за слабкого чи нерівномірного освітлення, а також у випадках складного фону. Навіть за часткового перекриття руки чи її швидкого руху Mediapipe здебільшого зберігає точність трекінгу.

У тестах із кількома руками в одному кадрі система коректно ідентифікує та відстежує кожну кисть окремо. Це особливо важливо для сценаріїв із багатьма користувачами чи складними жестами. Навіть у динамічних умовах алгоритми Mediapipe забезпечують стабільність та точність розпізнавання.

Результати роботи системи в реальному часі ілюструють її здатність до адаптації у складних умовах. На рисунках 3.3 і 3.4 показано, як Mediapipe ефективно розпізнає рухи та положення рук, забезпечуючи високу якість і точність у реальних застосунках.

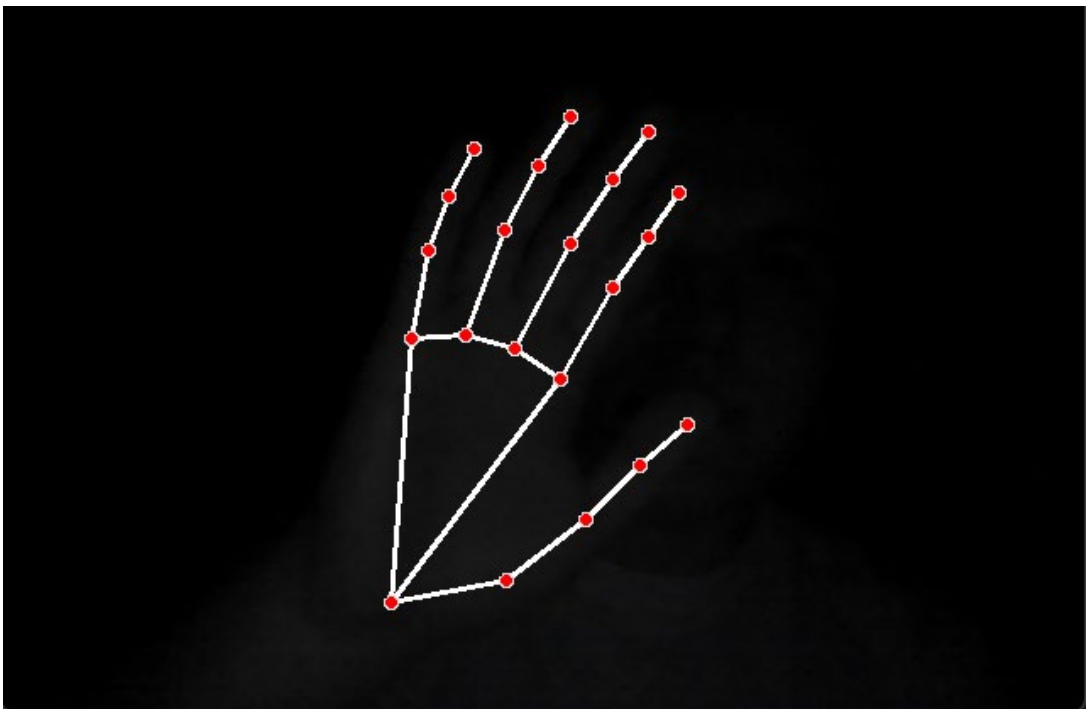

Рисунок 3.3 – Розпізнавання руки при майже повній темноті

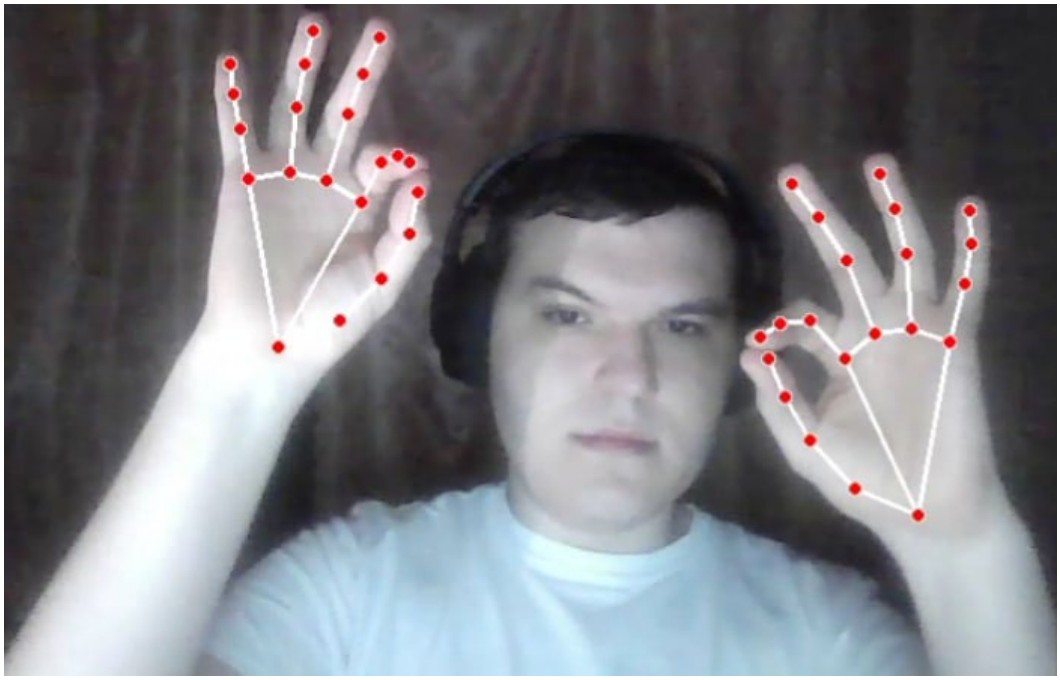

Рисунок 3.4 – Розпізнавання двох рук одночасно

Тестування з різними фонами – однотонним, кольоровим, з наявністю дрібних предметів – показало, що Mediapipe здатний добре ігнорувати зайві деталі та не сплутувати кисть руки з іншими об’єктами. Це надзвичайно важливо для практичного використання, адже користувачі не завжди мають

ідеальні умови зйомки. Таким чином, початковий етап виявлення руки можна визнати надійним і достатньо стійким до дестабілізуючих факторів.

3.4.2 Визначення жестів

Другим етапом було тестування вже навченого класифікатора жестів. Для оцінки ефективності було вибрано кілька жестів із різним ступенем подібності між собою. На рисунку 3.5 проілюстровано випадки, коли модель правильно класифікує жести «Hello», «Bye», «Yes», «No», «Thank You» і «Perfect». Можна помітити, що модель здатна коректно розпізнавати жести навіть при певному зміщенні кисті, зміні масштабу та неідеальному положенні пальців.

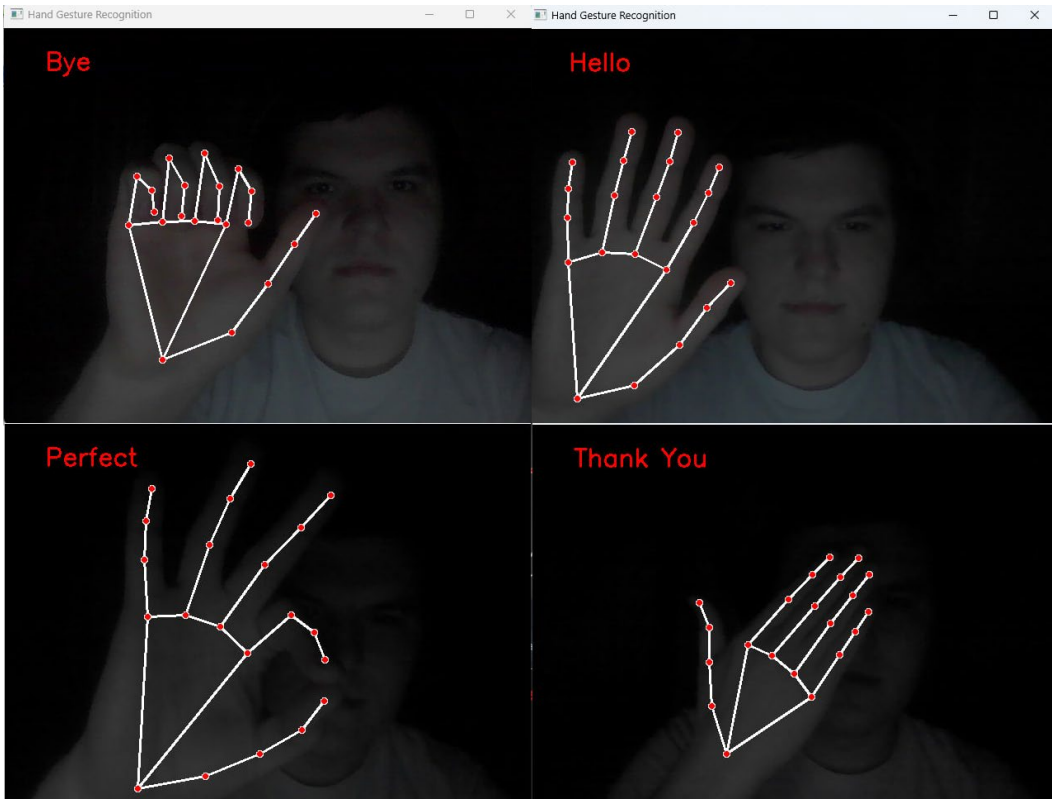

Рисунок 3.5 – Визначення жестів рук

Проте не обійшлося без помилок. На рисунку 3.6 наведено приклад, коли модель плутає жест «No» з «Perfect». Причиною такої помилки може бути

недостатня кількість зображень цього жесту в тренувальному датасеті або їхня недостатня різноманітність (різні ракурси, дистанції та умови освітлення). Також частину помилок можна пояснити тим, що деякі жести візуально схожі між собою, і мережі важко виділити специфічні унікальні ознаки, особливо якщо вони мало виражені.

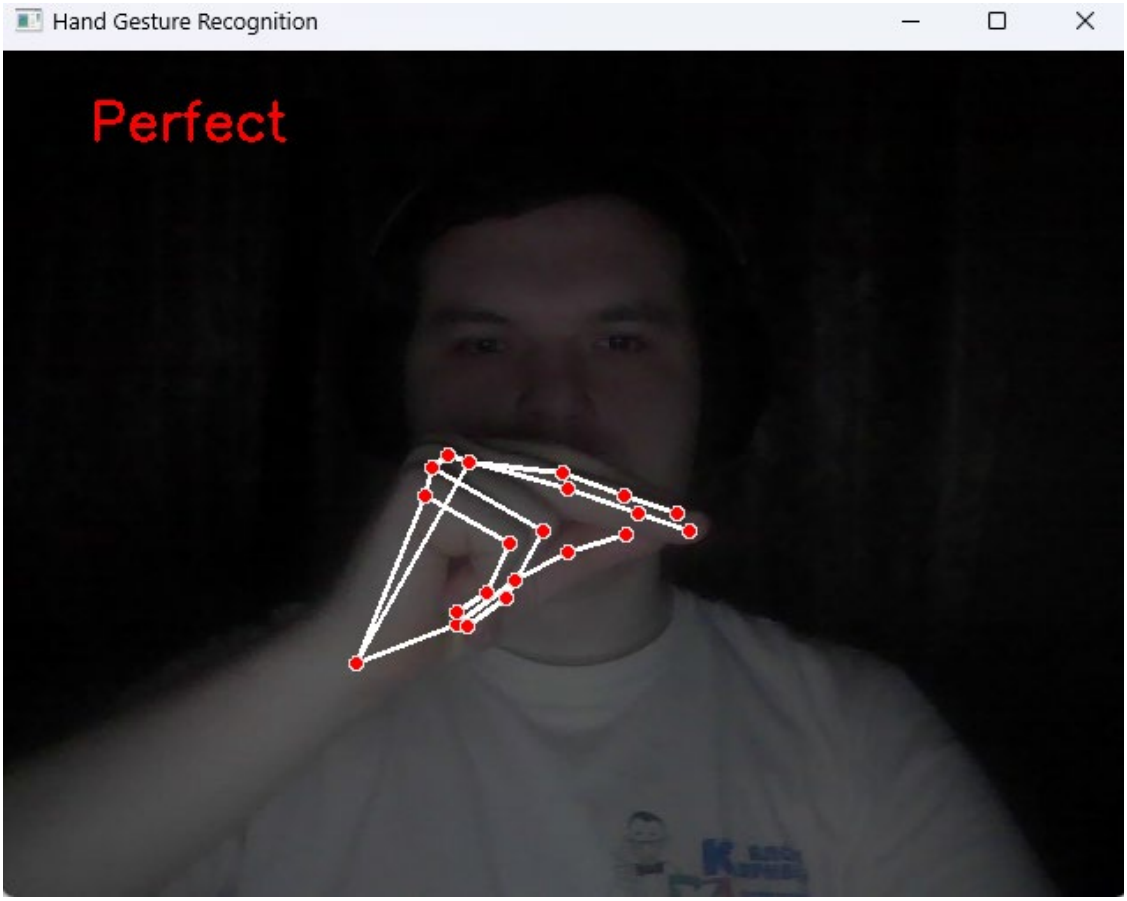

Рисунок 3.6 – Помилкове визначення жесту руки

3.4.3 Порівняння точності та продуктивності

Точність розпізнавання жестів [41] залежить від кількості використаних епох під час навчання, якості та обсягу датасету, а також від обраної архітектури мережі. Додаткове збільшення кількості епох навчання, застосування аугментації та використання більш потужних архітектур CNN дозволяє поступово підвищувати точність класифікації. Наприклад,

збільшення кількості епох з 10 до 20, а також застосування трансферного навчання (використання попередньо навчених моделей) дало змогу підвищити точність на 5-7%. Також помічено, що ретельна аугментація набору даних (ротації, зміни яскравості та контрасту) робить модель більш стійкою до реальних умов.

Продуктивність вимірювалась у кадрах за секунду (FPS) під час роботи в реальному часі [41]. На випробуваному пристрої (середньостатистичний ноутбук з GPU) вдалося досягти приблизно 15-20 FPS. Це достатньо для плавного та практично непомітного користувачеві розпізнавання жестів. Оптимізація, наприклад використання більш «легких» мереж, може підвищити FPS, водночас можливою ціною буде незначне зниження точності.

Підбір гіперпараметрів, зменшення розміру вхідних зображень, використання більш продуктивних GPU або TPU та ефективна паралелізація можуть додатково покращити продуктивність без суттєвого зниження точності.

3.4.4 Аналіз помилок та пропозиції щодо покращення

При аналізі помилок було виявлено, що модель найчастіше помиляється у випадках, коли жести мають подібну форму руки. Така подібність ускладнює визначення унікальних ознак, на основі яких можна було б чітко розрізнити жести.

Для підвищення точності пропонуються такі заходи:

− розширення датасету, зібрати та додати більше зображень жестів у різних умовах – при різному освітленні, з різних кутів зйомки, на різній відстані від камери, а також для рук різних форм і розмірів. Це забезпечить моделі здатність узагальнювати та розпізнавати жести більш впевнено;

− використання більш досконалих архітектур CNN, використання трансферного навчання з більш потужних моделей (наприклад, MobileNet,

EfficientNet, ResNet) може суттєво покращити розпізнавання деталей руки;

− додаткова сегментація руки від фону, відокремлення руки від оточення допоможе позбавити модель від непотрібних візуальних відволікаючих факторів.

Нижче наведений приклад коду, який демонструє етап сегментації кисті. Припустімо, що ми застосовуємо простий підхід із використанням колірного простору HSV для сегментації шкіри (код наведений умовно для ілюстрації):

Лістинг 3.2 Етап сегментації кисті:

```
import cv2
import numpy as np
def segment_hand(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_skin = np.array( [0, 30, 60], dtype=np.uint8)
    upper_skin = np.array( [20, 150, 255], dtype=np.uint8)
    mask = cv2.inRange(hsv, lower_skin, upper_skin)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)
    segmented = cv2.bitwise_and(image, image, mask=mask)
return segmented
```

Тестування цього підходу на етапі попередньої обробки перед передачею зображення до CNN показало, що, коли рука виділена з фону, модель класифікує жести точніше.

Пояснюється це тим, що після сегментації зображення більше схоже на ті, що використовувались при навчанні (за умови, що навчальні дані теж були з максимально чітким виділенням руки), тому модель легше розпізнає необхідні патерни. Такий прийом зменшує навантаження на CNN щодо вибору суттєвих ознак, оскільки фон не заважає.

ВИСНОВКИ

У межах кваліфікаційної роботи було розглянуто низку методів для розпізнавання рук та виявлення жестів рук у системах жестової мови, включаючи класичні алгоритми, сучасні глибинні нейронні мережі, а також використання фреймворків на зразок Mediapipe. Було проведено детальний аналіз існуючих методів, розроблено та оптимізовано алгоритм, який поєднує згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN) для ефективної обробки як статичних, так і динамічних жестів.

Під час роботи було досягнуто:

− високої точності та швидкодії моделі, розроблена модель демонструє ефективність у реальних умовах, забезпечуючи точне розпізнавання жестів рук із швидкістю обробки 15–20 FPS. Це дозволяє використовувати систему у режимі реального часу, що є критично важливим для інтерактивних застосунків;

− успішної реалізації експериментальної системи, створено систему, яка в режимі реального часу виявляє руку, сегментує її, витягує просторово-часові ознаки та визначає жест. Використання бібліотеки Mediapipe забезпечило точне визначення координат ключових точок руки, а застосування CNN та RNN дозволило ефективно класифікувати жести;

− оптимізації процесу навчання, завдяки застосуванню методів аугментації даних та трансферного навчання вдалося підвищити точність класифікації та зменшити ризик переобучення. Це забезпечило стійкість моделі до змін умов освітлення та фону;

− аналізу помилок та удосконаленню моделі, виявлено основні причини помилок класифікації, такі як схожість жестів та нестандартні умови зйомки. Запропоновано шляхи подолання цих проблем шляхом розширення та збалансування датасету, застосування додаткової сегментації руки та експериментування з більш складними архітектурами CNN.

Наукова новизна роботи полягає в комплексному підході до застосування глибинних моделей та фреймворків для розпізнавання жестової мови. Особливу увагу приділено інтеграції згорткових та рекурентних нейронних мереж для обробки як статичних, так і динамічних жестів, що дозволяє досягти високої точності та стійкості моделі. Крім того, використання Mediapipe для точного визначення ключових точок руки у реальному часі є інноваційним аспектом, який сприяє покращенню продуктивності системи. Практичні рекомендації щодо методів сегментації руки, аугментації даних та трансферного навчання також представляють новий внесок у сферу розпізнавання жестів.

Перспективи подальшого розвитку включають інтеграцію моделі з системами автоматичного перекладу жестової мови в усну чи текстову форму, що створить повноцінні платформи для комунікації між людьми з вадами слуху та слухачами. Додатково, можливо розширити діапазон розпізнаваних жестів, включивши більш складні послідовні фрази та адаптувавши систему до різних діалектів жестової мови. Інтеграція мультимодальних даних, таких як міміка обличчя та положення тіла, дозволить покращити розуміння контексту жестів та підвищити точність розпізнавання. Крім того, застосування більш потужних архітектур нейронних мереж та вдосконалення методів обробки даних сприятиме подальшому підвищенню продуктивності та адаптивності системи. Планується також впровадження системи на різні платформи та пристрої, що забезпечить широке застосування технології для покращення комунікації та інтеграції людей з вадами слуху в суспільство.

Результати дослідження апробовано у вигляді тез доповіді під час II Міжнародній науково-практичній конференції «Scientific research: modern challenges and future prospects» [43].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Давидов, М. В., Нікольський, Ю. В., & Пасічник, О. В. (2008). Дослідження ефективності методів розпізнавання у моделях жестової мови.

2. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

3. Murali, R. S. L., Ramayya, L. D., & Santosh, V. A. (2020). Sign language recognition system using convolutional neural network and computer vision.

4. The 26 letters and 10 digits of American Sign Language (ASL). URL: https://www.researchgate.net/figure/The-26-letters-and-10-digits-of-American-Sign-Language-ASL_fig1_328396430 (дата звернення 12.12.2024

5. Kinoshenko, D., Mashtalir, S., Shcherbinin, K., & Yegorova, E. (2008). Image quotient set transforms in segmentation problems. Information technologies & knowledge, 372.

6. International Journal of Engineering Innovations in Advanced Technology ISSN, 2582-1431.

7. Bankar, S., Kadam, T., Korhale, V., & Kulkarni, M. A. (2022). Real time sign language recognition using deep learning. International Research Journal of Engineering and Technology, 9(4), 955-959.

8. Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14, 100141.

9. Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164, 113794.

10. Bilonoh, B., Bodyanskiy, Y., Kolchygin, B., & Mashtalir, S. (2021, May). Tunable Activation Functions for Deep Neural Networks. In International Scientific Conference “Intellectual Systems of Decision Making and Problem of

Computational Intelligence" (pp. 624-633). Cham: Springer International Publishing.

11. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

12. Лавер, В. О., & Левчук, О. М. (2021). Обробка зображень: навч.-метод. посіб.

13. Лисенко, Б. С. (2023). Оцінка ефективності використання медіанного фільтру в цифровій обробці зображень.

14. Алієв, Е. І., & Городецька, О. К. (2022). Способи попередньої обробки зображень КТ ОГК для діагностики тромбоемболії легеневої артерії. Біомедична інженерія і технологія, (7), 69-80.

15. Akhtar, N., & Ragavendran, U. (2020). Interpretation of intelligence in CNN-pooling processes: a methodological survey. Neural computing and applications, 32(3), 879-898.

16. Sendjasni, A., Traparic, D., & Larabi, M. C. (2022, October). Investigating normalization methods for CNN-based image quality assessment. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 4113-4117). IEEE.

17. Dubey, S. R., & Chakraborty, S. (2021). Average biased ReLU based CNN descriptor for improved face retrieval. Multimedia Tools and Applications, 80(15), 23181-23206.

18. Mashtalir, S., & Mikhnova, O. (2017). Detecting significant changes in image sequences. Multimedia Forensics and Security: Foundations, Innovations, and Applications, 161-191.

19. Sunny, M. A. I., Maswood, M. M. S., & Alharbi, A. G. (2020, October). Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In 2020 2nd novel intelligent and leading emerging sciences conference (NILES) (pp. 87-92). IEEE.

20. Garcia, C. I., Grasso, F., Luchetta, A., Piccirilli, M. C., Paolucci, L., & Talluri, G. (2020). A comparison of power quality disturbance detection and

classification methods using CNN, LSTM and CNN-LSTM. Applied sciences, 10(19), 6755.

21. Xu, P., Huang, L., & Song, Y. (2022). An optimal method based on HOG-SVM for fault detection. Multimedia Tools and Applications, 81(5), 6995-7010.

22. Kim, M., Kwon, Y., Kim, J., & Kim, Y. (2022). Image classification of parcel boxes under the underground logistics system using CNN MobileNet. Applied Sciences, 12(7), 3337.

23. Ku, Y. J., Chen, M. J., & King, C. T. (2019, November). A virtual Sign Language translator on smartphones. In 2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW) (pp. 445-449). IEEE.

24. Murali, R. S. L., Ramayya, L. D., & Santosh, V. A. (2020). Sign language recognition system using convolutional neural network and computer vision. International Journal of Engineering Innovations in Advanced Technology ISSN, 2582-1431.

25. Tambon, F., Nikanjam, A., An, L., Khomh, F., & Antoniol, G. (2024). Silent bugs in deep learning frameworks: an empirical study of keras and tensorflow. Empirical Software Engineering, 29(1), 10.

26. Amprimo, G., Masi, G., Pettiti, G., Olmo, G., Priano, L., & Ferraris, C. (2024). Hand tracking for clinical applications: validation of the Google MediaPipe Hand (GMH) and the depth-enhanced GMH-D frameworks. Biomedical Signal Processing and Control, 96, 106508.

27. Ansari, I. A., & Bajaj, V. (2024). Image Processing with Python: A practical approach. IOP Publishing.

28. Patel, A., More, A., Kanojia, N., Munde, A., & Bijawe, M. A. Hand Tracking And Gesture Controlled Computer Virtual Mouse.

29. Amsaprabhaa, M., Sree, H. S., Muthamizhvalavan, K., Gummaraju, N., & Padmajaa, S. (2024, March). American Sign Language Real Time Detection Using TensorFlow and Keras in Python. In 2024 3rd International Conference for Innovation in Technology (INOCON) (pp. 1-6). IEEE.

30. Parimala, N., Teja, K. K. S., Kumar, J. A., Keerthana, G., Meghana, K., & Pitchai, R. (2024, April). Real-time Brightness, Contrast and The Volume Control with Hand Gesture Using Open CV Python. In 2024 10th International Conference on Communication and Signal Processing (ICCSP) (pp. 726-730). IEEE.

31. Гіловянц, К. Д. (2024). Моделі та методи аналізу зображень за допомогою нейромереж.

32. Khaliki, M. Z., & Başarslan, M. S. (2024). Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN. Scientific Reports, 14(1), 2664.

33. Hand Gestures Datase. URL: https://figshare.com/articles/dataset/Hand_Gestures_Dataset/24449197?file=42928318 (дата звернення 12.12.2024)

34. Савкін, Г. (2022). Сегментація зображень з використанням нейронних мереж.

35. Wang, Z. J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., ... & Chau, D. H. P. (2020). CNN explainer: learning convolutional neural networks with interactive visualization. IEEE Transactions on Visualization and Computer Graphics, 27(2), 1396-1406.

36. Гула, Т. В. (2023). Вибір гіперпараметрів для виявлення аномалій на зображеннях.

37. Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. Neural Computing and Applications, 35(23), 17095-17112.

38. Chang, C. M., & Tseng, D. C. (2020). Loose hand gesture recognition using cnn. In Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology: Methods and Algorithms, Proceedings of IC3DIT 2019, Volume 1 (pp. 87-96). Springer Singapore.

39. Матюнін, О. В. (2021). Модель сегментації зображень з застосуванням нейронної мережі Mask R-CNN.

40. Bahmei, B., Birmingham, E., & Arzanpour, S. (2022). CNN-RNN and data augmentation using deep convolutional generative adversarial network for environmental sound classification. IEEE Signal Processing Letters, 29, 682-686.

41. Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. International Journal of Machine Learning and Cybernetics, 10, 131-153.

42. Osipov, A., & Ostanin, M. (2021, August). Real-time static custom gestures recognition based on skeleton hand. In 2021 International Conference" Nonlinearity, Information and Robotics"(NIR) (pp. 1-4). IEEE.

43. Легкий М. Г. Як технології розпізнавання жестів рук покращують наше повсякденне життя II Міжнародній науково-практичній конференції «Scientific Research: Modern Challenges And Future Prospects». 2024. 135-138