

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем та програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Дослідження здатності штучного
інтелекту до розпізнавання жестової мови

Виконав: студент 6 курсу, групи СПм-61
спеціальності 121 «Інженерія програмного

забезпечення»

(шифр і назва спеціальності)

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М.Р.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль 2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет _____
(повна назва факультету)

Кафедра _____
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) _____
(прізвище та ініціали)
« » 20__ р.

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня _____
(назва освітнього ступеня)

за спеціальністю _____
(шифр і назва спеціальності)

студенту _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____

Керівник роботи _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка

Студент _____
(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____
(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота магістра за спеціальністю «Інженерія програмного забезпечення» на тему «Дослідження здатності штучного інтелекту до розпізнавання жестової мови». Пояснювальна записка до випускної роботи містить 67 сторінок, 14 ілюстрацій, 15 бібліографій, 1 таблиця та 4 додатки.

Метою кваліфікаційної роботи є розробка та навчання моделі штучного інтелекту, що зможе розпізнавати знаки жестової мови.

В роботі було проведено аналіз предметної області та визначено основні етапи для розробки моделі машинного навчання, обґрунтовано вибір технологій, описано процеси та принципи роботи нейронних мереж та продемонстровано наявні результати роботи навченої моделі.

Результатом виконання кваліфікаційної роботи є розроблена модель машинного навчання, що розпізнає всі знаки американської жестової мови.

Ключові слова: жестова мова, штучний інтелект, розпізнавання жестів, нейронна мережа, глибоке навчання, TensorFlow, Python

SUMMARY

Master's thesis in Software Engineering on the topic "Research of artificial intelligence ability to recognize sign language". The explanatory note to the thesis consists of 67 pages, 14 illustrations, 15 bibliographies, 1 table and 4 appendices.

The purpose of the qualification work is to develop and train an artificial intelligence model that can recognize sign language signs.

The work analyzes the subject area and identifies the main stages for developing a machine learning model, justifies the choice of technologies, describes the processes and principles of neural networks, and demonstrates the existing results of the trained model.

The result of the qualification work is a developed machine learning model that recognizes all signs of American Sign Language.

Keywords: sign language, artificial intelligence, gesture recognition, neural network, deep learning, TensorFlow, Python

ЗМІСТ

АНОТАЦІЯ	4
SUMMARY	5
ЗМІСТ	6
ВСТУП.....	8
1 АНАЛІТИЧНА ЧАСТИНА	10
1.1. Актуальність теми та аналіз найпопулярніших жестових мов	10
1.2. Технології розпізнавання жестів	12
1.3. Алгоритми машинного навчання в розпізнаванні жестів	14
1.3.1. Згорткові нейронні мережі (CNN)	14
1.3.2. Рекурентні нейронні мережі (RNN)	15
1.3.3. Метод опорних векторів (SVM)	16
1.3.4. Метод головних компонент (PCA)	17
1.3.5. Метод опорних точок (Landmark Detection)	18
1.3.6. Каскади класифікаторів від Адабуста	19
1.4. Підбір даних для навчання моделі ШІ	20
1.5. Обробка даних перед навчанням моделі ШІ	21
1.6. Постановка задачі	22
2 ПРОЕКТУВАННЯ ТА ОГЛЯД ТЕХНОЛОГІЙ.....	23
2.1. Аналіз предметної області	23
2.1.1. Значення та важливість розпізнавання жестів у комунікації	23
2.1.2. Перспективи використання штучного інтелекту для покращення розпізнавання жестів	24
2.2. Визначення вимог	25
2.3. Проектування програмної системи	26
2.3.1. Пошук акторів та варіантів використання	26
2.3.2. Огляд ключових варіантів використання	27
2.3.3. Обґрунтування вибору методології RUP для виконання проекту	29
2.3.4. Методологія RUP для розробки моделі штучного навчання	30

2.3.5. Функціональна архітектура системи	31
2.4. Порівняльна характеристика TensorFlow та PyTorch	32
2.5. Допоміжні бібліотеки використані у кваліфікаційній роботі	34
2.5.1. Опис та основні переваги використання бібліотеки OpenCV	34
2.5.2. Опис та основні переваги використання бібліотеки Numpy	34
2.5.3. Опис та основні переваги використання бібліотеки Mediarpipe	35
2.6. Основні переваги, сфери застосування та механізми роботи глибокого навчання	36
2.6.1. Як глибоке навчання досягає вражаючих результатів	36
2.6.2. Приклади глибокого навчання в роботі	37
2.6.3. Основний механізм роботи глибокого навчання	38
2.6.4. Різниця між машинним та глибоким навчанням	40
2.6.5. Вибір між машинним та глибоким навчанням	42
3 РЕЗУЛЬТАТИ РОЗРОБКИ ТА ТЕСТУВАННЯ МОДЕЛІ	43
3.1. Розподіл даних та навчання моделі	43
3.2. Тестування моделі у режимі реального часу	47
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	50
4.1. Охорона праці	50
4.2. Безпека в надзвичайних ситуаціях	54
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТКИ.....	62
ДОДАТОК А.....	63
ДОДАТОК Б	66
ДОДАТОК В	71

ВСТУП

Штучний інтелект (ШІ) сьогодні знаходиться на передових рубежах технологічного розвитку, розкриваючи безмежні можливості в різних сферах життя. Однією з ключових областей застосування штучного інтелекту є розпізнавання мови, де значна увага приділяється вдосконаленню систем розпізнавання текстових та усно-мовленнєвих виразів. Проте, наряд з цим, сучасний розвиток штучного інтелекту визначає нову галузь досліджень — розпізнавання мови жестів.

Мова жестів, використовувана людьми для вираження емоцій, комунікації та передачі інформації, є важливим компонентом нашого спілкування. Цей унікальний спосіб вираження може бути важливим каналом взаємодії, особливо у тих випадках, коли словесна мова недостатня або неможлива. Розуміння мови жестів вимагає не лише великої кількості інформації, що включає в себе рухи рук, обличчя та інші частини тіла, але й враховує контекст і ситуаційний аспект.

З урахуванням цього, ця магістерська робота присвячена глибокому аналізу та експериментальному дослідженню здатностей штучного інтелекту до розпізнавання мови жестів. Метою є не лише розвиток та оптимізація алгоритмів, спроможних адекватно визначати та інтерпретувати жести, але і створення моделей, що здатні враховувати контекст та адаптуватися до різних ситуацій.

В ході роботи планується розглядати різноманітні аспекти, пов'язані із розпізнаванням мови жестів, включаючи технологічні виклики, що виникають у зв'язку із складністю аналізу рухів та виразів, а також питання етики, пов'язані зі збором та використанням інформації, що стосується індивідуального жестового спілкування.

У цьому контексті робота визначається як спроба визначити та розібрати ключові аспекти проблематики розпізнавання мови жестів з використанням

штучного інтелекту, а також вирішити технічні та етичні виклики, які можуть виникнути в цьому контексті.

Технологічний прогрес у сфері розпізнавання мови жестів привертає увагу науковців та індустрії з різних галузей, включаючи комп'ютерні науки, психологію, технології віртуальної реальності та медицину. Сучасні системи штучного інтелекту стають все більш здатними до аналізу та інтерпретації складно-структурованих даних, але визнання і розуміння жестів залишається викликом через їх варіативність та суб'єктивність в сприйнятті.

Дослідження в області розпізнавання мови жестів має великий потенціал в сферах віртуальної та змішаної реальності, де можливість взаємодії користувача з об'єктами та іншими користувачами через жести може збільшити ефективність та натуральність комунікації. Вдосконалення таких систем також може мати важливе значення у сферах, де спілкування словесною мовою є обмеженим або недоступним, таких як мовчазна комунікація та взаємодія з людьми з обмеженими можливостями.

Для досягнення цих цілей важливо розглядати технічні виклики, такі як точність розпізнавання, адаптивність до різних особливостей жестів, а також обробка великої кількості даних в реальному часі. Крім того, робота повинна враховувати питання конфіденційності та етики в обробці інформації, пов'язаної з рухами та виразами обличчя. Висвітлення цих аспектів в магістерській роботі має на меті створити повнішу картину не лише з технічної, але і з соціокультурної та етичної перспектив.

Загальна мета роботи полягає в створенні інноваційних підходів та технологій для розпізнавання мови жестів з використанням штучного інтелекту, що сприятиме розвитку сучасних систем взаємодії та комунікації. Результати даного дослідження можуть служити основою для подальших вдосконалень в галузі штучного інтелекту та розширення можливостей міжособистісної та міжкультурної комунікації.

1 АНАЛІТИЧНА ЧАСТИНА

1.1. Актуальність теми та аналіз найпопулярніших жестових мов

Сьогодні глухонімі люди по всьому світу стикаються з рядом викликів та обмежень у сучасному суспільстві, яке здебільшого не забезпечує їм повноцінну участь та рівні можливості. Однією з головних проблем є обмежений доступ до інформації. Багато комунікаційних платформ, включаючи веб-сайти, відеоматеріали та соціальні мережі, зорієнтовані на аудіовізуальні засоби сприйняття, не завжди враховують потреби людей, які сприймають інформацію через жестову мову чи письмово.

Важливим аспектом є також обмежені можливості у сфері навчання та працевлаштування. Глухонімі особи можуть стикатися зі складнощами у доступі до освітніх ресурсів та отриманні якісної професійної підготовки через відсутність адаптованих матеріалів та навчальних середовищ.

Також, у сфері медичного обслуговування, глухонімі іноді відчують відсутність спеціалізованих послуг та фахівців, які б допомагали їм здійснювати доступ до необхідної медичної інформації та послуг.

Соціальна ізоляція і неврахування специфіки життя глухонімих осіб можуть викликати психологічні та емоційні труднощі. Брак адекватного розуміння та підтримки з боку суспільства може призводити до відчуття відстороненості та складностей у встановленні міжособистісних відносин.

Підсумуючи, глухонімі люди у сучасному суспільстві зіштовхуються з багатьма викликами, які включають в себе обмежений доступ до інформації, учбових та трудових можливостей, а також соціальну ізоляцію та відчуття неприйняття з боку оточуючого середовища. Покращення цих аспектів вимагає комплексних зусиль у різних сферах, спрямованих на створення більш інклюзивного та рівноправного суспільства для всіх його членів.

У теперішній час існує безліч різноманітних мов жестів, які використовуються різними групами людей та в різних культурних та

соціальних контекстах. Однією з найбільш відомих є мова жестів для глухих, яка використовується в спілкуванні між глухими та наглими.

Поза цими мовами жестів, існують також жести та мовні знаки, що використовуються в різних сферах життя, таких як спорт, бізнес, мистецтво чи техніка. Наприклад, жести, використовувани в аеропортах для керування рухом літаків, виглядають іншим чином в порівнянні із жестами, які використовують ведучі новин або люди в інших сферах взаємодії.

Загалом, різноманітність мов жестів свідчить про те, як цей спосіб комунікації адаптується та розвивається в різних культурних, соціальних і технологічних контекстах.

American Sign Language (ASL). American Sign Language (ASL) - це мова жестів, що використовується глухими спільнотами в Сполучених Штатах. ASL є повноправною мовою з власною граматиною та лексикою. Спілкування в ASL включає жести рук, міміку та рухи тіла. Ця мова знаків має свої варіації в різних регіонах США.

British Sign Language (BSL). British Sign Language (BSL) - це мова жестів, яка використовується глухими в Великобританії та в деяких частинах Австралії та Нової Зеландії. BSL відрізняється від ASL та має власну лексику та граматику. Ця мова є важливою частиною культурного ідентитету глухих спільнот у Великобританії.

Deutsche Gebärdensprache (DGS). Deutsche Gebärdensprache (DGS) - це мова жестів, яка використовується в глухих спільнотах у Німеччині. DGS є відокремленою мовою з унікальною лексикою та граматиною. Ця мова використовується в різних аспектах життя, включаючи освіту та масові заходи.

Langue des Signes Française (LSF). Langue des Signes Française (LSF) - це мова жестів, що використовується глухими спільнотами во Франції та частинах Бельгії. LSF є важливим засобом комунікації для французьких глухих людей та має свої власні особливості, які роблять її унікальною серед світових мов жестів.

Ці мови жестів відіграють ключову роль у спілкуванні та культурній ідентифікації глухих спільнот у своїх країнах, створюючи багатозарові та динамічні системи взаємодії та вираження емоцій та думок.

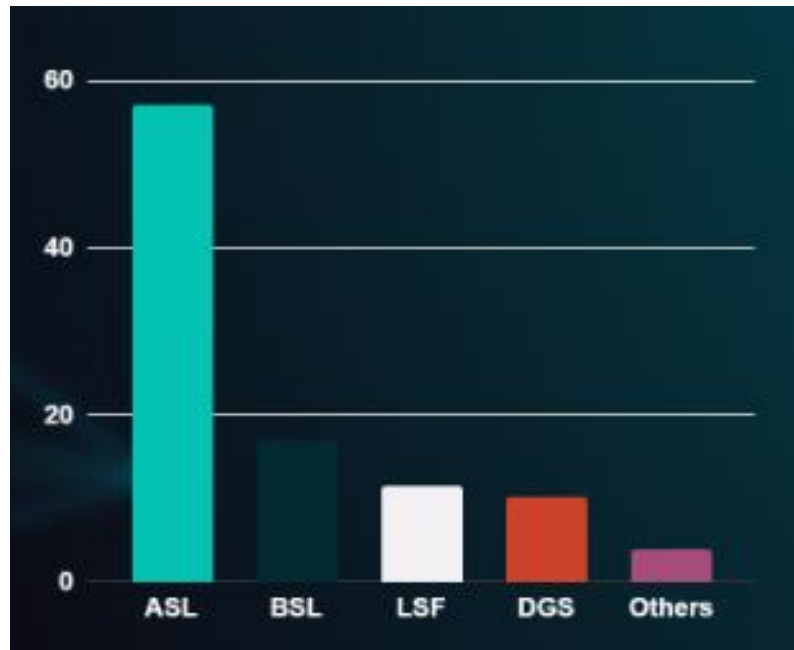


Рис. 1.1 – Графік популярності різних мов жестів

В подальшому ASL (American Sign Language) буде основою для навчання системи штучного інтелекту, адже вона є найбільш популярною.

1.2. Технології розпізнавання жестів

Технології розпізнавання жестів включають в себе різноманітні методи та інструменти, які дозволяють системам і пристроям інтерпретувати та розуміти жести користувачів. Ось деякі ключові аспекти та технології, які використовуються для розпізнавання жестів:

1. Камери та відео-системи:

- *RGB та інфрачервоні камери:* Використовуються для запису відео та отримання зображень, що використовуються для розпізнавання жестів.
 - *Триразові та глибинні камери:* Дозволяють визначати глибину та розміщення об'єктів, поліпшуючи точність розпізнавання.
2. Сенсорні технології:
- *Сенсорні панелі:* Реагують на торкання, жести та рухи, забезпечуючи можливості вводу, які можуть бути використані для розпізнавання жестів.
 - *Гіроскопи та акселерометри:* Здатні виявляти рухи та нахил пристроїв, використовуються для визначення жестів, таких як крутіння або нахил.
3. Комп'ютерний зір та обробка зображень:
- *Алгоритми розпізнавання образів:* Використовують машинне навчання для ідентифікації та класифікації жестів на зображеннях чи відео.
 - *Техніки відслідковування обличчя та руху:* Використовуються для визначення положення та рухів обличчя та рук для розпізнавання жестів.
4. Носії та віртуальна реальність:
- *Спеціальні датчики:* Використовуються на носіях, таких як рукавички або костюми, для точного відслідковування рухів та жестів користувача.
 - *Віртуальна реальність та кінект:* Використовують технології, які дозволяють інтерактивну взаємодію на основі розпізнавання рухів та жестів.
5. Системи штучного інтелекту та машинного навчання:
- *Класифікація та навчання:* Алгоритми машинного навчання використовуються для навчання систем розпізнавання жестів на основі великої кількості даних.
 - *Нейронні мережі:* Глибокі нейронні мережі можуть вдосконалювати точність розпізнавання та реагувати на складніші жести.

6. Звукові технології:

- *Розпізнавання голосу:* Звукові команди та жести можуть використовуватися спільно для збільшення можливостей вводу та взаємодії.

Технології розпізнавання жестів постійно розвиваються, використовуючи сучасні методи та інновації для поліпшення точності та реактивності систем у різних областях, включаючи віртуальну реальність, медицину, ігрову індустрію та багато інших.

1.3. Алгоритми машинного навчання в розпізнаванні жестів

1.3.1. Згорткові нейронні мережі (CNN)

Згорткові нейронні мережі (CNN) - це клас нейронних мереж, спеціально призначених для обробки зображень. Вони використовуються в задачах комп'ютерного зору та розпізнавання образів. Основна ідея полягає в тому, щоб автоматично вивчати характеристики та патерни в зображеннях без явного програмування.

Центральні елементи CNN включають згорткові шари, які використовують фільтри для виділення різних ознак зображення. Шари пулінгу використовуються для зменшення розмірності зображення та зменшення обчислювальних витрат.

CNN є потужним інструментом для класифікації жестів у визначеному контексті. Вони використовуються для автоматичного вивчення важливих ознак і патернів на зображеннях рухів або поз в реальному часі. Згорткові шари в мережі дозволяють взяти до уваги локальні особливості жестів, а шари пулінгу допомагають зменшити розмірність даних та вилучити ключові ознаки. Функції активації, такі як ReLU, додають нелінійність до моделі, а повністю

з'єднані шари використовуються для класифікації розпізнаних жестів. Цей підхід дозволяє моделі ефективно вивчати та розпізнавати рухи рук та жести, роблячи його ідеальним для застосувань, пов'язаних із введенням жестів в інтерфейси або системи управління.

1.3.2. Рекурентні нейронні мережі (RNN)

Рекурентні нейронні мережі (RNN) - це клас нейронних мереж, які призначені для роботи з послідовними даними та динамічними вхідними сигналами. Основна особливість RNN полягає в їхній здатності запам'ятовувати інформацію з попередніх моментів часу та використовувати цю інформацію для обробки нових вхідних даних.

Основні компоненти RNN включають в себе ваги, які змінюються від кроку до кроку, та функції активації, які дозволяють вводити нелінійність. Однак у класичних RNN може виникати проблема з втратою інформації на великих відстанях в часі, що може ускладнити вивчення довгих залежностей.

Деякі варіанти RNN, такі як Long Short-Term Memory (LSTM) та Gated Recurrent Unit (GRU), були створені для вирішення цієї проблеми швидкою заборною або фільтрацією інформації, що проходить через мережу. Ці покращені архітектури RNN знаходять застосування в багатьох завданнях обробки послідовних даних.

Рекурентні нейронні мережі (RNN) використовуються для класифікації жестів, забезпечуючи здатність аналізувати послідовності даних, такі як рухи та пози рук. Мережі цього типу мають рекурентні зв'язки, що дозволяють передавати інформацію від одного кроку часу до іншого. Такий підхід особливо корисний в задачах, пов'язаних із варіативним контекстом жестів. Ваги в мережі оновлюються залежно від попередніх взаємодій та подій, дозволяючи

моделі враховувати довгострокові залежності в динаміці жестів. Процес тренування включає в себе адаптацію ваг та використання функцій втрат для оптимізації навчання моделі. В результаті, RNN стає ефективним інструментом для класифікації жестів, адаптованим до варіативного характеру послідовних даних.

1.3.3. Метод опорних векторів (SVM)

Метод опорних векторів (SVM) — це алгоритм машинного навчання, використовуваний для завдань класифікації та регресії. Основна ідея полягає в тому, щоб знайти оптимальну гіперплощину, яка найкращим чином розділяє дані різних класів у просторі ознак. SVM спробує максимізувати відстань між цією гіперплощиною та найближчими точками даних кожного класу, визначаючи їхні опорні вектори.

Для досягнення цього, SVM використовує ядро, яке дозволяє відобразити дані в багатовимірний простір, де вони можуть бути лінійно розділені. Різні ядра можуть використовуватися для різних типів даних.

Крім того, SVM добре справляється з управлінням високо вимірних даних і є відомим своєю ефективністю в просторах великої розмірності. Він також може бути використаний для регресії та вирішення задач ансамблювання.

Метод опорних векторів (SVM) в контексті класифікації жестів використовується для ефективного визначення границі розділення між різними класами жестів у просторі ознак. Основна мета полягає в тому, щоб знайти оптимальну гіперплощину, яка максимізує відстань між цією гіперплощиною та найближчими точками даних кожного класу, визначаючи їхні опорні вектори.

SVM може бути використаний для класифікації жестів, визначаючи їхні просторові характеристики та патерни. За допомогою ядер, SVM може

ефективно враховувати складні залежності в даних жестів, а також пристосовуватися до різних типів жестів та їх варіацій.

SVM вигідно використовувати в задачах класифікації жестів, оскільки цей метод добре працює з високо вимірними даними та може надійно розділяти простори різних класів, забезпечуючи ефективне визначення та класифікацію жестів в реальному часі.

1.3.4. Метод головних компонент (PCA)

Метод головних компонент (PCA) - це статистичний метод зменшення розмірності даних, який використовується для виявлення та виділення основних змінних у наборі даних. Головна ідея полягає в тому, щоб знайти нові ортогональні вектори (головні компоненти), які захоплюють найбільший розкид в даних. Ці головні компоненти використовуються для перетворення початкових даних в новий простір, в якому зберігається найбільша частина варіації.

PCA є потужним інструментом для зменшення розмірності даних та видалення зайвих кореляцій між ознаками. Цей метод застосовується в різних областях, включаючи аналіз зображень, розпізнавання обличчя, обробку сигналів та інші завдання, де важлива робота з великими об'ємами даних. При використанні PCA важливо враховувати, що він оптимально працює, коли дані є лінійно залежними.

Метод головних компонент (PCA) може бути використаний для класифікації жестів як ефективний засіб для аналізу та виділення основних змінних у великому обсязі даних жестів. Використовуючи PCA, можна визначити головні компоненти, які найкраще описують варіації в жестах та визначають ключові характеристики рухів.

Цей метод дозволяє зменшити розмірність даних, зберігаючи при цьому важливі інформаційні змінні. Відповідно, використання PCA для класифікації жестів може полегшити аналіз та розпізнавання рухів рук, забезпечуючи більш ефективну обробку та відокремлення різних класів жестів в просторі основних компонент.

1.3.5. Метод опорних точок (Landmark Detection)

Метод опорних точок, або Landmark Detection, є технікою в комп'ютерному зорі та обробці зображень, яка використовується для виявлення та локалізації ключових точок або особливих місць на обличчі чи об'єкті.

Метод опорних точок грає важливу роль в задачах розпізнавання обличчя, анімації обличчя, а також у ряді медичних застосувань. Використання цього методу дозволяє системам автоматичного розпізнавання визначати анатомічні особливості та рухи обличчя, що може бути корисним у різних сценаріях, включаючи відновлення емоцій, анімацію персонажів або визначення певних властивостей обличчя в медичних дослідженнях.

Метод опорних точок, в контексті класифікації жестів використовується для точного виявлення та локалізації ключових точок чи опорних точок на руках або обличчі під час виконання жестів. Ці опорні точки можуть включати місця на долонях, кінчиках пальців або інші анатомічні особливості.

Використання методу опорних точок для класифікації жестів дозволяє системам визначати конкретні пози та рухи рук, що є важливим для розпізнавання та інтерпретації жестів. Цей підхід може бути використаний у системах розуміння жестів для введення команд, інтерфейсів взаємодії або в контексті розробки технологій управління за допомогою жестів в реальному часі.

1.3.6. Каскади класифікаторів від Адабуста

Каскади класифікаторів від Адабуста - це метод машинного навчання, який базується на адаптивному бустінгу та використовує послідовність класифікаторів для виявлення об'єктів або особливостей в зображеннях. Алгоритм працює шляхом тренування послідовності слабких класифікаторів, кожен з яких намагається виправити помилки попереднього.

Основна ідея полягає в тому, щоб створити каскад, де кожен класифікатор використовується для швидкого відсіювання негативних областей, тобто місць, де об'єкт відсутній. Це дозволяє прискорити процес виявлення об'єктів, зосереджуючись на потенційно позитивних областях. Кожен класифікатор в каскаді навчається враховувати найбільш важливі ознаки для виявлення конкретного об'єкта.

Каскади класифікаторів від Адабуста в контексті класифікації жестів використовуються для ефективного виявлення та розпізнавання рухів рук та поз. Алгоритм базується на принципі послідовної обробки зображення серією класифікаторів, кожен з яких спроектований для швидкого відсіювання негативних областей та фокусу на потенційно позитивних регіонах.

Цей підхід дозволяє прискорити процес класифікації, забезпечуючи швидше виявлення жестів у відеопотоці чи зображенні. Кожен класифікатор в каскаді покликаний виявити конкретні ознаки, що вказують на наявність певного жесту. Ітеративний процес тренування дозволяє адаптувати класифікатори до конкретних особливостей жестів.

Каскади класифікаторів від Адабуста для класифікації жестів застосовуються в сучасних системах розпізнавання жестів, де важливо досягти високої точності та ефективності у реальному часі, наприклад, в інтерфейсах взаємодії чи системах управління.

1.4. Підбір даних для навчання моделі ШІ

Для навчання ШІ необхідна велика кількість вхідних даних. Джерела даних для штучного інтелекту можуть бути різноманітними і включають в себе різні джерела інформації, які відображають реальний світ та дозволяють моделям навчатися та вирішувати завдання в різних галузях. Найрозповсюдженіші джерела для отримання навчальних даних:

1. Бази даних. Структуровані дані, які можна витягти з реляційних баз даних. Це може включати дані з електронних таблиць, баз даних клієнтів, фінансових систем тощо.

2. Сенсори та Інтернет речей (IoT). Дані, зібрані з сенсорів та пристроїв Інтернету речей. Це може включати дані з смарт-домів, медичних пристроїв, автомобілів та інших підключених пристроїв.

3. Соціальні мережі. Інформація, зібрана з соціальних платформ, таких як Facebook, Twitter, LinkedIn. Дані з соціальних мереж можуть використовуватися для аналізу поведінки користувачів, прогнозування трендів тощо.

4. Текстові документи та інтернет-ресурси. Дані з текстових джерел, таких як веб-сторінки, блоги, новинні статті.

5. Зображення та відео. Дані від камер та відеосистем, що використовуються для розпізнавання образів, аналізу відображення тощо.

6. Аудіо та мовлення. Дані, які збираються з мікрофонів та інших аудіо-джерел. Використовуються для розпізнавання мови, голосових асистентів тощо.

7. Геопросторові дані. Дані, пов'язані з місцезнаходженням. Це може включати дані GPS, географічні інформаційні системи (ГІС) та інші дані, які вказують на просторовий контекст.

Ці джерела входять у склад великих обсягів даних, від яких моделі штучного інтелекту можуть вивчати та робити прогнози. Правильний вибір та ефективна обробка цих даних грає ключову роль у вдосконаленні точності та ефективності моделей.

1.5. Обробка даних перед навчанням моделі ШІ

Обробка даних перед навчанням штучного інтелекту - це важливий етап у створенні моделей машинного навчання. Цей процес включає в себе кілька кроків, щоб гарантувати, що дані є придатними для навчання та використання моделі. Ось детальний огляд цього процесу:

1. Очищення даних. Очищення даних включає видалення аномальних значень, обробку відсутніх даних, корекцію помилок та стандартизацію форматів. Це допомагає уникнути спотворення результатів та покращити якість даних.

2. Експлоративний аналіз даних. Аналізуються основні статистики та властивості даних для кращого розуміння їх розподілу. Це допомагає виявити особливості та важливі аспекти, які можна використовувати під час навчання моделі.

3. Нормалізація. Деякі алгоритми машинного навчання краще працюють, коли дані нормалізовані. Це може включати у себе стандартизацію числових значень або перетворення категоріальних даних в числові представлення.

4. Вибір ознак. Визначення, які ознаки (змінні) будуть використовуватися для навчання моделі. Важливо обрати ті ознаки, які мають найбільший вплив на цільову змінну та зменшують складність моделі.

5. Розбиття даних. Дані зазвичай розбиваються на тренувальний, валідаційний та тестовий набори. Тренувальний використовується для навчання моделі, валідаційний для налаштування гіперпараметрів, а тестовий для оцінки її ефективності.

6. Оптимізація та узагальнення. Під час навчання моделі важливо уникати перенавчання (overfitting) та піднавчання (underfitting). Для цього використовують різні методи регуляризації та перехресної перевірки.

7. Кодування категоріальних даних. Якщо в наборі даних є категоріальні ознаки, їх необхідно закодувати в числовий формат. Це може бути зроблено за допомогою методів, таких як one-hot encoding чи label encoding.

8. Балансування класів. У випадках, коли дані мають нерівномірний розподіл між класами (наприклад, в задачах класифікації), важливо використовувати методи для балансування класів, щоб уникнути перекоосу в бік домінуючих класів.

9. Захист приватності та безпека. При обробці даних також слід враховувати аспекти захисту приватності та безпеки. Це може включати в себе анонімізацію, шифрування та інші заходи для забезпечення конфіденційності даних.

Обробка даних перед навчанням моделей ШІ - це трудомісткий, але необхідний процес, що може суттєво покращити якість та ефективність моделі.

1.6. Постановка задачі

Проаналізувавши предметну область можна визначити основні задачі для виконання кваліфікаційної. Отже для поставленого завдання, основними етапами є:

- підбір даних для навчання нейронної мережі на основі жестової мови ASL;
- обробити вихідні дані для коректного навчання нейронної мережі;
- навчити нейронну мережу за допомогою відповідної бібліотеки машинного навчання;
- перевірити точність навченої мережі;
- перевірити здатність навченої мережі розпізнавати знаки жестової мови в реальному часі;

2 ПРОЕКТУВАННЯ ТА ОГЛЯД ТЕХНОЛОГІЙ

2.1. Аналіз предметної області

2.1.1. Значення та важливість розпізнавання жестів у комунікації

Розпізнавання жестів у комунікації відіграє ключову роль, оскільки воно допомагає людям виражати свої думки, емоції та наміри не лише словами, а й невербальними елементами. Емоційний вираз через жести дозволяє передавати настрій, додавати емоційну глибину до комунікативних ситуацій. Жести підсилюють вербальні повідомлення, роблячи їх більш виразними та запам'ятовуваними.

Також, жести можуть слугувати сигналами узгодження чи неузгодження, вказуючи на рівень згоди або незгоди з висловлюваннями співрозмовника. Крім того, вони відіграють важливу роль у культурному контексті, де різні жести можуть мати різне значення в різних культурах.

Позиція тіла та рухи також можуть вказувати на відкритість чи закритість людини для спілкування. Розпізнавання жестів є ефективним інструментом для зменшення бар'єрів мови, допомагаючи взаєморозумінню у випадках, коли спілкування відбувається між людьми, що користуються різними мовами.

Всі ці аспекти допомагають створити повноцінне та ефективне спілкування, де комбінація вербальних та невербальних елементів розширює можливості виразності та взаєморозуміння. Розпізнавання жестів є важливою навичкою в міжособистісній комунікації, сприяючи взаємодії та розумінню в різноманітних ситуаціях.

Поміж основних функцій розпізнавання жестів у комунікації слід визначити їхню здатність передавати інформацію без використання мовних засобів. Це стає корисним в тих ситуаціях, коли мовний обмін неможливий або обмежений, таких як зустрічі в гучних місцях, на різних культурних заходах або у випадках, коли мови розмовників різняться.

Розпізнавання жестів також сприяє підвищенню ефективності комунікації в забруднених або небезпечних середовищах, де гучність чи інші фактори можуть заважати сприйняттю вербальних повідомлень. У таких обставинах невербальні засоби, зокрема жести, стають важливими засобами обміну інформацією.

Значення розпізнавання жестів визначається також у можливості підтримувати невербальні ритуали та обряди, що є частиною культурного спадку. Вони можуть включати ритуальні жести, які використовуються для вираження поваги, привітання або підтримки у різних культурах.

Крім того, жести важливі в бізнес-комунікації, де вони можуть вказувати на різноманітні стилі лідерства, рішення або ставлення до обговорюваної теми. Ефективне використання жестів може сприяти встановленню позитивного враження та покращити взаємодію в професійних взаємовідносинах.

Отже, розпізнавання жестів визначається не лише їхньою роллю у вираженні емоцій чи підсиленні слів, а й їхньою універсальністю у передачі інформації та участі у культурних та соціальних аспектах комунікації.

2.1.2. Перспективи використання штучного інтелекту для покращення розпізнавання жестів

Використання штучного інтелекту (ШІ) для поліпшення розпізнавання жестів відкриває широкі перспективи у різних областях. Розвиток алгоритмів машинного навчання, зокрема нейронних мереж, дозволяє точніше та ефективніше розпізнавати жести на основі обширних наборів даних.

Це особливо актуально у сферах віртуальної реальності та розширеної реальності, де можливості взаємодії систем розпізнавання з жестами стають важливим елементом створення інтерактивних середовищ. Інтеграція ШІ із сенсорами, такими як камери та мікрофони, розширює можливості отримання

комплексної інформації про навколишнє середовище та взаємодії з користувачем.

Крім того, можливість використання ШІ для адаптації до індивідуальних особливостей користувачів через методи машинного навчання робить системи розпізнавання більш ефективними та персоналізованими.

В цілому, ці технологічні тенденції у розпізнаванні жестів відкривають нові перспективи для створення більш інтуїтивних і ефективних систем взаємодії між людьми та комп'ютерами. Розширення застосувань ШІ у цій області може призвести до подальшого вдосконалення технологій і поліпшення якості комунікації у різних сферах життя.

2.2. Визначення вимог

На основі аналізу предметної області та постановки задачі. Система повинна мати наступний перелік вимог:

Функціональні вимоги:

- тестування моделі в режимі реального часу;
- зміна параметрів моделі;
- зміна навчальних даних;
- зміна способу представлення даних.

Нефункціональні вимоги:

- час навчання моделі до 4 годин (в залежності від кількості навчальних даних);
- мінімальна кількість навчальних епох – 5;
- нижня допустима точність розпізнавання символів – 80%.

2.3. Проектування програмної системи

2.3.1. Пошук акторів та варіантів використання

Перед початком безпосередньої роботи на програмним забезпеченням необхідно визначити акторів та варіанти використання системи. Для демонстрації даних сценаріїв можна скористатися UML діаграмами. UML діаграма – це графічний інструмент для моделювання систем та їх компонентів, який надає структуроване представлення взаємодій між різними частинами системи. Unified Modeling Language (UML) надає стандартний набір графічних символів та правил для створення діаграм, що дозволяє спростити розуміння та спілкування щодо архітектури та функціональності системи.

При аналізі предметної області я визначив таких акторів:

- Звичайний користувач;
- Дослідник.

При визначенні функціональних вимог даної системи я визначився з такими варіантами використання:

- Розпізнавання символів, жестової мови;
- Зміна параметрів навчання моделі;
- Зміна навчальних даних для навчання моделі;
- Регулювання представлення результатів;

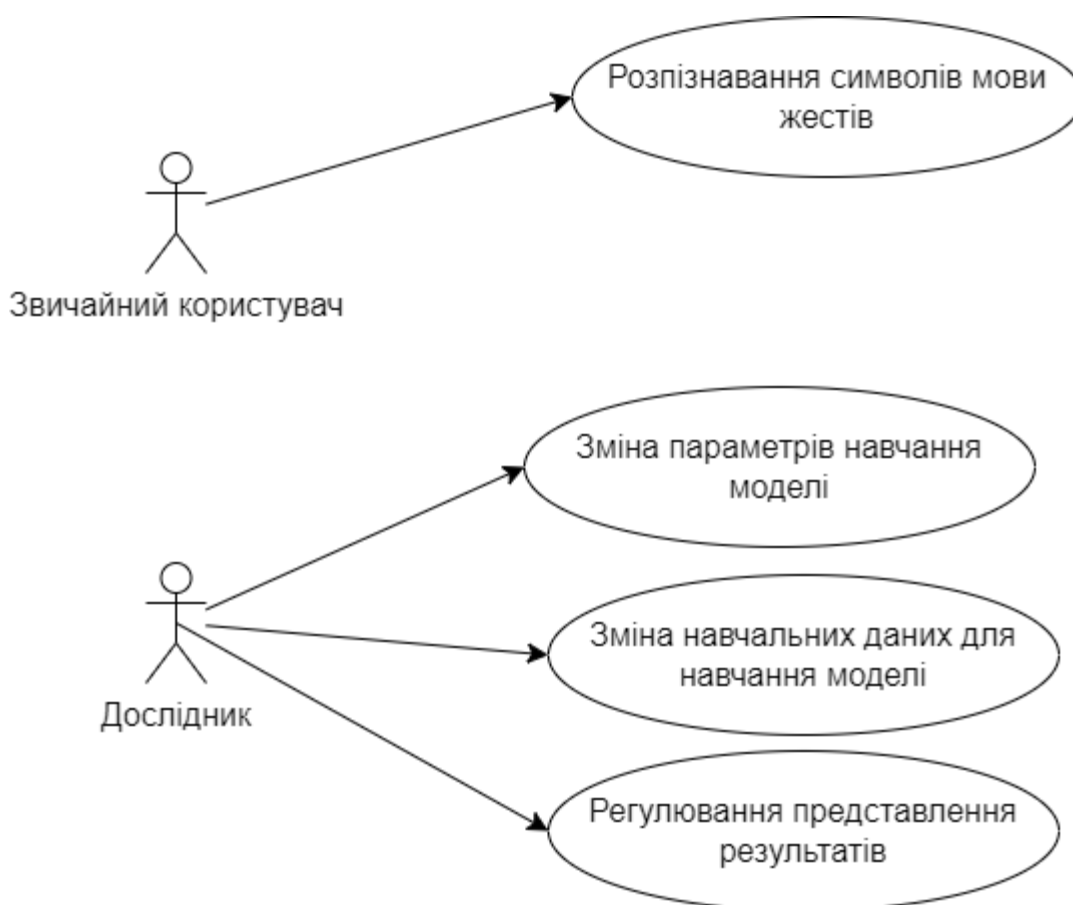


Рис. 2.1 –.Варіанти використання системи

2.3.2. Огляд ключових варіантів використання

Визначившись із основними акторами та варіантами використання варто зазначити потік роботи основних варіантів використання нашої системи. Всі основні та альтернативні варіанти використання описані в таблиці 2.1

Таблиця 2.1 – Опис основних прецедентів системи

Код	Найменування	Основний сценарій	Альтернативний сценарій
ЗК1	Розпізнавання символів жестової мови	<ol style="list-style-type: none"> 1. Звичайний користувач відкриває проект. 2. Користувач запускає виконання першої частини додатку із завантаженням необхідних бібліотек 3. Користувач запускає проект від моменту завантаження збереженої моделі. 4. Система запускає вікно із передбаченнями демонстрованих користувачем жестів. 	<ol style="list-style-type: none"> 2.a. Відсутні певні бібліотеки 2.a.1. Система виводить повідомлення про відсутність бібліотек 3.a. Неправильний шлях до збереженої моделі 3.a.1. Система виводить повідомлення про неправильний шлях до моделі
ЗК2	Зміна параметрів моделі навчання	<ol style="list-style-type: none"> 1. Дослідник відкриває проект. 2. Дослідник змінює потрібні йому параметри у розділі навчання моделі. 3. Дослідник запускає навчання моделі. 4. Дослідник зберігає навчену модель 	<ol style="list-style-type: none"> 3.a. Некоректні налаштування моделі 3.a.1. Система виводить повідомлення про некоректні налаштування 4.a. Несумісний формат збереження моделі 4.a.1. Система виводить повідомлення про несумісний формат збереження моделі
ЗК3	Зміна навчальних даних для навчання моделі	<ol style="list-style-type: none"> 1. Дослідник відкриває проект. 2. Дослідник завантажує потрібні дані у папку з проектом. 3. Дослідник вказує змінений шлях до даних у секції підготовка даних. 4. Дослідник зберігає навчену модель 	<ol style="list-style-type: none"> 3.a. Неправильний шлях до збереженої моделі 3.a.1. Система виводить повідомлення про неправильний шлях до даних

Продовження таблиці 2.1

ЗК4	Регулювання представлення результатів	1. Дослідник відкриває проект. 2. Дослідник додає потрібні графіки у секції демонстрація статистичних результатів 3. Дослідник запускає створені графіки. 4. Системи виводить графічні результати	3.a. Некоректно налаштовані графіки 3.a.1. Система виводить повідомлення про конкретну помилку
-----	---------------------------------------	--	---

2.3.3. Обґрунтування вибору методології RUP для виконання проекту

Rational Unified Process (RUP) представляє собою методологію розробки програмного забезпечення, яка охоплює повний життєвий цикл проекту. Починаючи з визначення бізнес-вимог, на першому етапі, визначаються основні цілі та завдання проекту, а також ідентифікуються зацікавлені сторони та їх очікування.

На етапі аналізу вимог бізнес-вимоги деталізуються, розкладаючись на конкретні функціональні та нефункціональні вимоги, включаючи обсяг робіт, ресурси та графік проекту. Після чого настає етап архітектури, де розробляється загальна структура системи, обираються архітектурні стилі та технології, що визначають подальший розвиток проекту.

Реалізація системи відбувається на етапі розробки, де команда розробників застосовує архітектурні рішення для створення функціональності. Тестування, наступний етап, включає в себе різноманітні тести для перевірки працездатності та відповідності вимогам.

Процес розвитку передбачає додавання нових функціональностей та виправлення помилок, а управління змінами забезпечує ефективний контроль

над змінами в проєкті. Управління якістю включає в себе тестування, використання кращих практик та виправлення дефектів.

Після успішного завершення всіх етапів впровадження системи в реальне середовище, наступає етап супроводу. На цьому етапі забезпечується підтримка та вдосконалення системи в процесі експлуатації, з урахуванням відгуків користувачів та необхідних модифікацій. Таким чином, RUP забезпечує структурований та повний підхід до розробки програмного забезпечення, покриваючи всі етапи проєкту від визначення вимог до супроводу в експлуатації.

2.3.4. Методологія RUP для розробки моделі штучного навчання

Розглядаючи Rational Unified Process (RUP) у контексті розробки моделі машинного навчання для розпізнавання символів жестової мови, першим етапом буде визначення бізнес-вимог, пов'язаних із системою розпізнавання жестів. Це включатиме в себе розуміння точних символів, які слід розпізнавати, а також очікування щодо продуктивності та точності системи.

На етапі аналізу вимог, бізнес-вимоги будуть конкретизовані в функціональні та нефункціональні вимоги моделі машинного навчання. Визначатимуться параметри, які впливають на ефективність розпізнавання жестів, такі як швидкість відповіді та рівень точності.

Архітектурний етап буде спрямований на створення оптимальної архітектури моделі машинного навчання, вибір відповідних алгоритмів навчання та визначення взаємодії з іншими компонентами системи, такими як інтерфейс користувача.

Реалізація моделі включатиме в себе програмування та конфігурацію алгоритмів машинного навчання, а також інтеграцію їх у велику систему розпізнавання жестів.

На етапі тестування буде проведено ретельне валідацію та верифікацію роботи моделі на різних тестових наборах даних, включаючи реальні сценарії використання символів жестової мови.

Процес розвитку буде включати постійне оновлення та вдосконалення моделі відповідно до нових вимог або виявлених недоліків, зокрема враховуючи особливості жестової мови та можливість розширення архітектури для підтримки нових жестів.

Управління змінами важливо для ефективного впровадження змін в модель машинного навчання, особливо при адаптації до реальних умов або змінених вимог користувачів.

Оцінка якості моделі буде проводитися за допомогою метрик, таких як точність, чутливість та специфічність, з метою забезпечення високої продуктивності та точності системи.

Супровід означає постійну підтримку та вдосконалення системи під час експлуатації, враховуючи відгуки користувачів та можливість автоматизованого оновлення моделі машинного навчання для покращення її продуктивності та розпізнавання символів жестової мови. Таким чином, RUP надає структурований та гнучкий підхід до розробки моделі машинного навчання, забезпечуючи високу якість та ефективність продукту.

2.3.5. Функціональна архітектура системи

Функціональна архітектура програмного забезпечення - це ключовий аспект розробки, який визначає спосіб виконання функцій та взаємодії між їх компонентами. Одним із основних аспектів є модульність, де програма розбивається на незалежні модулі, що сприяє підтримці, розширюваності та тестуванню коду. Визначення ефективної комунікації між компонентами включає в себе визначення інтерфейсів та протоколів передачі даних.

Також важливим є структура даних та їх організація в системі, що включає в себе бази даних та алгоритми обробки інформації. Для досягнення конкретних цілей, визначаються алгоритми та логіка роботи системи, а також стратегії обробки помилок та відновлення.

У функціональній архітектурі враховується безпека та доступність системи, включаючи заходи безпеки, такі як автентифікація та шифрування, а також доступність для користувачів та стратегії для неперервної роботи системи. Проектування інтерфейсів користувача та їх взаємодія з системою також увійшли в функціональну архітектуру.

Найсуттєвішим є розгляд можливості масштабування системи, щоб ефективно відповідати на зростання обсягу даних чи користувачів, включаючи стратегії розподіленої обробки та кешування. Ці аспекти разом формують функціональну архітектуру, яка визначає основні принципи організації та функціонування системи, сприяючи створенню ефективних, масштабованих та надійних програмних рішень.

2.4. Порівняльна характеристика TensorFlow та PyTorch

Вибір бібліотек і фреймворків може значно полегшити впровадження моделі. TensorFlow і PyTorch є двома з найбільш популярних фреймворків для навчання глибоких нейронних мереж. Scikit-learn, з іншого боку, служить бібліотекою для класичного машинного навчання. Важливо обрати ті технології, які як найкраще спосіб відповідають конкретній задачі.

TensorFlow і PyTorch – це два найпопулярніших відкритих фреймворки для реалізації моделей машинного навчання та глибокого навчання. Кожен з них має свої переваги та особливості, і вибір між ними зазвичай залежить від конкретних потреб та вподобань розробника чи дослідника. Нижче подано порівняльний огляд TensorFlow і PyTorch за різними аспектами:

1. Архітектура та граф обчислень:

TensorFlow використовує статичний обчислювальний граф, що спочатку визначає структуру графу, а потім проводить обчислення. Це сприяє оптимізації та розпаралеленню обчислень.

PyTorch використовує динамічний обчислювальний граф, що створює граф "на льоту" під час виконання коду, роблячи його більш читабельним та легко відлагоджуваним.

2. Спільнота та екосистема:

TensorFlow має велику та активну спільноту, а також багато готових бібліотек та інструментів для різних завдань машинного навчання.

PyTorch також має активну спільноту та швидкий розвиток, славиться своєю простотою та зручністю використання.

3. Навчання та документація:

TensorFlow має обширну офіційну документацію та онлайн-курси, спрямовані на початківців, проте в статичному графі може виникати більше складнощів у розробці коду для різних операцій.

PyTorch славиться легкістю навчання та дружньою динамічною обчислювальною графікою, а його документація вважається доброю.

4. Складність використання та продуктивність:

TensorFlow може бути складнішим для початківців через вивчення статичного графа, проте це може призвести до більшої продуктивності при роботі з великими моделями.

PyTorch відомий своєю простотою та інтуїтивністю, особливо для прототипування та досліджень.

У результаті порівняння, ви вирішили віддати перевагу фреймворку TensorFlow, враховуючи його варіативність і швидкість освоєння.

2.5. Допоміжні бібліотеки використані у кваліфікаційній роботі

2.5.1. Опис та основні переваги використання бібліотеки OpenCV

Бібліотека `cv2`, також відома як OpenCV (Open Source Computer Vision Library), є однією з найпопулярніших бібліотек для роботи з зображеннями та відео у світі комп'ютерного зору та комп'ютерного бачення. Вона написана на мові програмування C++ і має багато інтерфейсів для мов програмування, таких як Python, Java і інші. Основна мета OpenCV - це надання інструментів для розробки різноманітних комп'ютерних зорових застосувань.

Основні функції:

- зчитування та запис зображень і відео;
- операції над зображеннями (зміна розмірів, кольорів, відокремлення об'єктів тощо);
- обробка відео;
- додавання різноманітних фільтрів;
- розпізнавання країв різних об'єктів (лиць, предметів тощо);
- машинне навчання;
- визначення факту руху об'єктів.

2.5.2. Опис та основні переваги використання бібліотеки NumPy

Бібліотека NumPy (Numerical Python) – це ключовий інструмент для роботи з числовими даними в мові програмування Python. NumPy забезпечує підтримку високопродуктивних масивів та операцій над ними, що робить його важливим компонентом для розв'язання завдань наукового обчислення та обробки даних.

Основні можливості та концепції NumPy:

- структурування та обробка масивів;
- універсальні функції для роботи з масивами;
- індексація та зрізи масивів;
- векторизація операцій (операції над цілими масивами одразу, замість роботи по окремому індексу);
- бродкастинг (спеціальний механізм NumPy для більш гнучкої роботи з різними формами та розмірами);
- набір різноманітних операцій лінійної алгебри;
- рандомізація (включений модель для генерації випадкових чисел або цілих масивів випадкових чисел).

2.5.3. Опис та основні переваги використання бібліотеки Mediapipe

Mediapipe – це бібліотека для роботи з комп'ютерним зором та аналізу медіаданих, розроблена Google. Вона дозволяє легко використовувати різноманітні алгоритми для виявлення та відстеження ключових точок на обличчі, руках, тілі і навіть визначення різноманітних жестів.

Основні переваги Mediapipe:

- виявлення ключових точок лиця;
- відстеження різноманітних рухів рук;
- відстеження жестів;
- виявлення об'єктів;
- аналіз медіаданих;
- просте API;
- мультиплатформеність.

Приклади роботи бібліотеки будуть приведені в наступному розділі.

2.6. Основні переваги, сфери застосування та механізми роботи глибокого навчання

2.6.1. Як глибоке навчання досягає вражаючих результатів

Висока точність є ключовим аспектом, який визначає значущість глибинного навчання. Ця технологія досягає рівня точності в розпізнаванні, який раніше вважався недосяжним. Це досягнення не лише відкриває можливості для вдосконалення побутової електроніки з високими стандартами користувачів, але також має критичне значення для застосувань у сфері безпеки, зокрема в системах безпілотних автомобілів.

Сучасні досягнення в глибинному навчанні настільки вражаючі, що у деяких завданнях, наприклад, класифікації об'єктів на зображеннях, воно перевищує здатність людини. Незважаючи на те, що ідея глибинного навчання теоретично розглядалася ще у 1980-х роках, лише останнім часом вона стала практично корисною, і це визначається двома основними причинами.

По-перше, глибоке навчання вимагає величезних обсягів анотованих даних, таких як мільйони зображень і тисячі годин відео, щоб розробляти та навчати безпілотні автомобілі. По-друге, для успішної реалізації глибинного навчання необхідна значна обчислювальна потужність. Використання графічних процесорів з високою продуктивністю, особливо з паралельною архітектурою, є ефективним для глибокого навчання. Інтеграція їх у системи з кластерами чи обчисленнями в хмарі дозволяє розробникам значно зменшити час навчання глибоких нейронних мереж, перехід від тижнів до годин або навіть менше.

2.6.2. Приклади глибокого навчання в роботі

Програми глибокого навчання виявляють свою ефективність у різноманітних галузях, розширюючи можливості від автоматизованого водіння до медичних технологій.

У сфері автомобілебудування, дослідники використовують глибоке навчання для автоматичного виявлення знаків зупинки, світлофорів і пішоходів. Це не лише сприяє зменшенню аварійності, а й підвищує рівень безпеки на дорогах.

В аерокосмічній та оборонній галузях глибоке навчання використовується для розпізнавання об'єктів на супутниках, визначення місцезнаходження об'єктів та встановлення безпечних або небезпечних зон для військових. Це технологічне вдосконалення сприяє підвищенню ефективності оборонних систем та забезпечує надійний контроль над важливими територіями.

У галузі медичних досліджень глибоке навчання виявляє клітини раку, сприяючи швидкому та точному виявленню захворювань. Наприклад, науковці з Каліфорнійського університету в Лос-Анджелесі розробили мікроскоп з високо розмірними даними, що дозволяє ідентифікувати ракові клітини з вражаючою точністю.

У галузі промислової автоматизації глибоке навчання сприяє підвищенню безпеки працівників, автоматично визначаючи небезпечні відстані від важкої техніки. Це розширює можливості виробничих процесів, забезпечуючи ефективніше та безпечніше використання технічного обладнання.

У сфері електроніки глибоке навчання стає важливим інструментом для автоматизованого розпізнавання і перекладу мови. Додатки домашніх помічників, які базуються на глибокому навчанні, взаємодіють з користувачем через голосові команди та враховують його вподобання, що полегшує їхнє використання та підвищує зручність взаємодії.

2.6.3. Основний механізм роботи глибокого навчання

Більшість підходів у глибокому навчанні спираються на використання архітектур нейромережі, що часто призводить до того, що ці моделі отримують назву "глибокими нейронними мережами".

В термінології глибокого навчання, поняття "глибокий" зазвичай вказує на кількість прихованих шарів у структурі нейронної мережі. Традиційні нейронні мережі зазвичай обмежені двома або трema прихованими шарами, тоді як глибокі мережі можуть включати до 150 таких шарів, дозволяючи здійснювати більш складні обчислення та вивчати більше абстрактних функцій.

Основною характеристикою моделей глибокого навчання є їх здатність до навчання на великих наборах анотованих даних. Завдяки використанню різноманітних нейромережевих архітектур, ці моделі можуть автоматично вивчати важливі ознаки без необхідності ручного вилучення їх з вхідних даних. Це важливо, оскільки дозволяє розробникам створювати більш гнучкі та адаптивні системи, здатні ефективно вирішувати завдання в різних областях, від розпізнавання об'єктів до мовового розпізнавання.

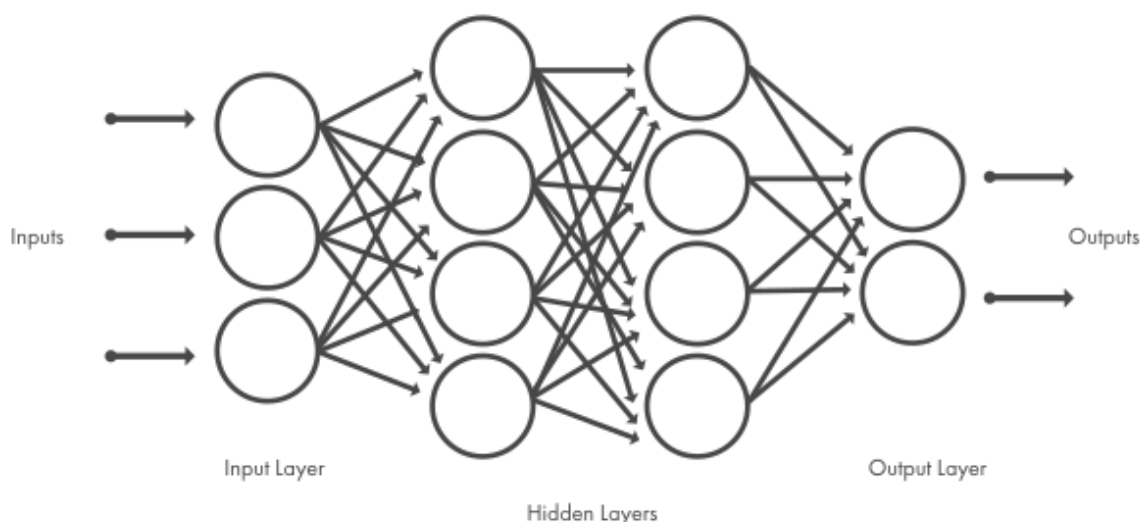


Рис. 2.1 – Нейронні мережі, які організовані у вигляді шарів, що складаються з набору взаємопов'язаних вузлів.

Один з найвизначніших типів глибоких нейронних мереж представляє собою згорткові нейронні мережі, часто відомі як CNN або ConvNet. Ці мережі використовують згорткові шари для аналізу вхідних даних та застосовують двовимірні фільтри, зробивши їх особливо ефективними для обробки 2D-інформації, такої як зображення.

Згорткові нейронні мережі інтегрують в себе унікальний підхід, який усуває необхідність вручного вилучення ознак. Іншими словами, користувачеві не потрібно вручну визначати характеристики, які використовуються для класифікації зображень. Замість цього, CNN автоматично витягає ознаки безпосередньо із зображень. Ці ознаки не є наперед навченими; вони навчаються під час тренування мережі на наборі зображень.

Автоматизоване вилучення ознак робить згорткові нейронні мережі особливо ефективними для завдань комп'ютерного зору, таких як класифікація об'єктів. Ця особливість дозволяє їм адаптуватися до різноманітних вхідних даних та забезпечує високу точність у розпізнаванні та класифікації об'єктів на зображеннях без необхідності попереднього визначення конкретних ознак.

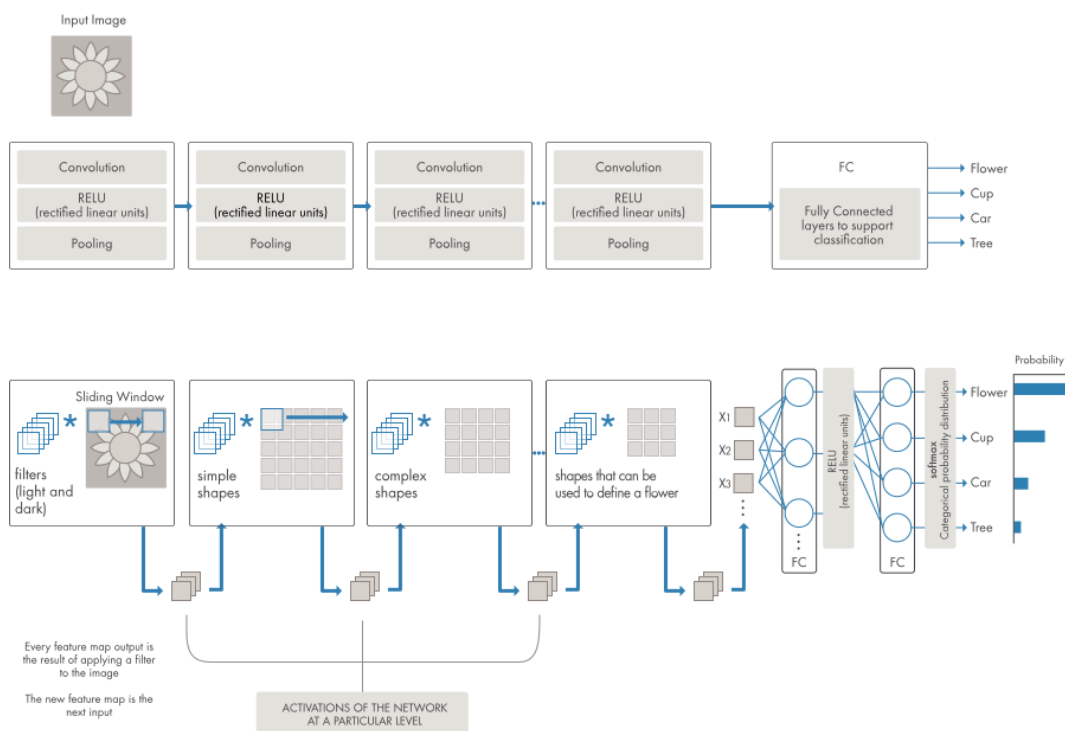


Рис. 2.2 – Приклад мережі з багатьма згортковими шарами

Шари згорткових нейронних мереж вивчають різноманітні характеристики зображення через використання десятків або сотень прихованих шарів. Кожен із цих прихованих шарів вносить свій власний внесок у розуміння та аналіз вхідних даних, додаючи рівень складності до ознак, які вони вивчають.

Для уточнення, можна розглянути приклад: перший прихований шар може бути спеціалізований на виявленні основних контурів та країв у зображенні. У той час як другий прихований шар може фокусуватися на виокремленні абстрактних форм та структур, що можуть вказувати на конкретні об'єкти. Такий підхід дозволяє згортковим нейронним мережам автоматично ієрархізувати та абстрагувати ознаки зображення на різних рівнях, підвищуючи рівень дискретності та розуміння представленого візуального контенту.

Окрім того, останні приховані шари в мережі можуть спеціалізуватися на виявленні більш складних структур та деталей, що відповідають загальній структурі об'єкта, що підлягає розпізнаванню. Цей градієнтний підхід у вивченні ознак робить згорткові нейронні мережі ефективними у завданнях розпізнавання об'єктів, сприяючи автоматизованому та ієрархічному вивченню складних структур на зображеннях.

2.6.4. Різниця між машинним та глибоким навчанням

Глибоке навчання – це окрема галузь машинного навчання, що відрізняється від інших методів. Зазвичай, у традиційному підході до машинного навчання, експерти спочатку ручним чином виділяють відповідні ознаки з зображень, і лише після цього використовують ці ознаки для створення моделі, яка може класифікувати об'єкти на зображеннях. У

глибокому навчанні відповідні ознаки автоматично вивчаються із зображень. Додатково, в глибокому навчанні використовується метод "наскрізного навчання", де мережі отримують необроблені дані та конкретне завдання, таке як класифікація, і самостійно навчаються, як це виконувати.

Ще однією ключовою відмінністю є те, що алгоритми глибокого навчання масштабуються разом із збільшенням обсягу даних. У поверхневому навчанні може виникнути ситуація, коли вони досягають певного рівня продуктивності і перестають покращуватися при додаванні додаткових прикладів та навчальних даних.

Основна перевага мереж глибокого навчання полягає в їхній здатності продовжувати оптимізацію та покращення зі зростанням обсягу даних. Це означає, що, навіть при збільшенні об'єму навчальних даних, глибокі нейронні мережі здатні адаптуватися та постійно вдосконалюватися, що робить їх ефективними для розв'язання різноманітних завдань у світі комп'ютерного зору та машинного навчання.

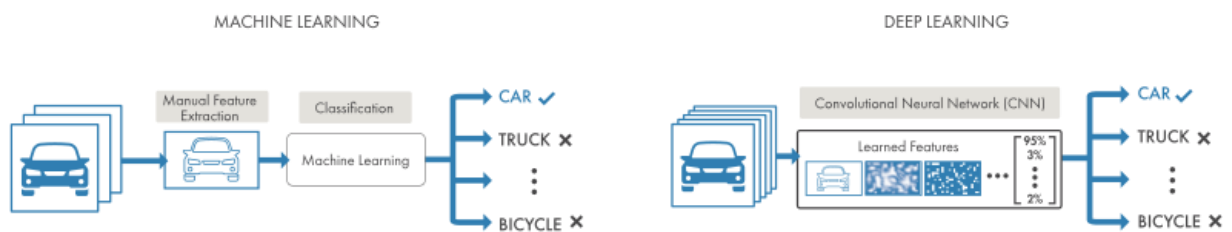


Рис. 2.3 – Порівняння роботи машинного навчання (зліва) та глибокого навчання (справа)

У традиційному підході до машинного навчання вам, як експерту, доводиться вручну обирати ключові ознаки та класифікатори для ефективного сортування та аналізу зображень. Цей процес вимагає глибокого розуміння вхідних даних та досвіду визначення релевантних характеристик для конкретного завдання.

У глибокому навчанні, навпаки, весь цей процес автоматизований. Етапи вилучення ознак і моделювання відбуваються автоматично завдяки потужним

глибоким нейронним мережам. Мережі вивчають представлення ознак безпосередньо з набору даних, самостійно визначаючи ієрархію ознак та їхні відповідності завданню класифікації.

Отже, у глибокому навчанні відсутня необхідність вручну визначати або вибирати ознаки чи класифікатори, оскільки модель здатна автоматично навчатися визначати важливі характеристики з великої кількості даних. Це зробило глибоке навчання надзвичайно потужним і ефективним для різноманітних завдань у сферах комп'ютерного зору, розпізнавання образів та природної мови.

2.6.5. Вибір між машинним та глибоким навчанням

У сфері машинного навчання існує безліч методів та моделей, які можна вибрати, залежно від конкретної задачі, обсягу оброблюваних даних і характеристик поставленої задачі. У той час як глибоке навчання є однією з потужних стратегій, для його успішного використання необхідно мати значну кількість даних, таких як тисячі зображень, для ефективного тренування моделі. Додатково, для швидкої обробки цих великих обсягів даних використання графічних процесорів є важливим аспектом.

При виборі між традиційним машинним навчанням і глибоким навчанням важливо враховувати наявність високопродуктивного графічного процесора та достатню кількість маркованих даних. У випадку відсутності одного чи обох цих елементів, може бути доцільніше звернутися до методів машинного навчання, оскільки глибоке навчання, як правило, вимагає більше ресурсів та зусиль. Для досягнення надійних результатів у глибокому навчанні часто необхідно мати принаймні кілька тисяч маркованих зображень.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ТА ТЕСТУВАННЯ МОДЕЛІ

3.1. Розподіл даних та навчання моделі

Для навчання моделі було використано попередньо розподілені дані з платформи Kaggle.

Після групування даних статистика розподілу наступна:

- Кількість символів для розпізнавання: 29;
- Кількість зображень у тренувальній вибірці: 69600;
- Кількість зображень у тестовій вибірці: 17400.

Ці дані становлять ключовий етап у розвитку та оптимізації нашої моделі, надаючи нам можливість досягти високої точності та надійності у завданнях розпізнавання символів.

```
Total number of symbols: 29  
Number of training images: 69600  
Number of testing images: 17400
```

Рис. 3.1 – Розподіл даних

Після групування даних наступним ключовим етапом стало створення та налаштування моделі для ефективного навчання. Задані параметри відіграють важливу роль у формуванні архітектури моделі та досягненні оптимальних результатів. Ось деякі деталі налаштувань, які були використані:

1. Навчальний розмір зображення 64 x 64 px. Використання цього розміру дозволяє оптимізувати роботу моделі та полегшує процес навчання, забезпечуючи при цьому достатню деталізацію для ефективного розпізнавання символів.

2. Кількість шарів моделі – 5. Задана кількість шарів визначає комплексність моделі та її здатність виявляти внутрішні зв'язки в даних. Оптимально підібрана архітектура забезпечує баланс між точністю та швидкістю навчання.

3. Кількість параметрів – 1660253. Це число вказує на загальну кількість ваг моделі, які піддаються оптимізації під час навчання. Воно є важливим показником складності моделі та може впливати на час навчання та ресурси, необхідні для його проведення.

4. Кількість тренувальних параметрів – 1660253. Це число вказує на кількість параметрів, які насправді оптимізуються під час навчання. Збалансовані значення цього параметру є ключовим елементом для досягнення стабільної та ефективної моделі.

Ці налаштування становлять фундамент для навчання моделі та її успішної адаптації до поставленого завдання розпізнавання символів на зображеннях.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 64, 64, 64)         4864
conv2d_1 (Conv2D)            (None, 64, 64, 64)         102464
max_pooling2d (MaxPooling2D) (None, 16, 16, 64)         0
dropout (Dropout)            (None, 16, 16, 64)         0
conv2d_2 (Conv2D)            (None, 16, 16, 128)        204928
conv2d_3 (Conv2D)            (None, 16, 16, 128)        409728
max_pooling2d_1 (MaxPooling2D) (None, 4, 4, 128)         0
dropout_1 (Dropout)          (None, 4, 4, 128)         0
...
Total params: 1660253 (6.33 MB)
Trainable params: 1660253 (6.33 MB)
Non-trainable params: 0 (0.00 Byte)

```

Рис. 3.2 – Параметри моделі

Навчання моделі зайняв приблизно 2 години 26 хв. Дане значення варіюється в залежності від потужності вашого комп'ютера.

```
Train on 69600 samples
Epoch 1/5
69600/69600 [=====] - 2267s 33ms/sample - loss: 1.5859 - acc: 0.5217
Epoch 2/5
69600/69600 [=====] - 1736s 25ms/sample - loss: 0.2253 - acc: 0.9265
Epoch 3/5
69600/69600 [=====] - 1607s 23ms/sample - loss: 0.1319 - acc: 0.9595
Epoch 4/5
69600/69600 [=====] - 1594s 23ms/sample - loss: 0.1105 - acc: 0.9692
Epoch 5/5
69600/69600 [=====] - 1595s 23ms/sample - loss: 0.0972 - acc: 0.9748
```

Рис. 3.3 – Вивід з процесу навчання моделі

В результаті навчена модель після тестування на виділеній вибірці показала результат точності – 99.707%.

```
Accuracy for test images: 99.707 %
```

Рис. 3.4 – Відсоток точності моделі

Для ілюстрації точності моделі та аналізу її ефективності використовуємо стандартну бібліотеку `scikit-learn` для побудови конфузійної матриці. Ця матриця послужить візуальним інструментом для вивчення процесу прогнозування та визначення символів, в яких модель допустила помилки. Вона дозволяє чітко визначити та відстежувати невірні передбачення, що сприяє подальшому вдосконаленню та налаштуванню моделі для досягнення кращих результатів у завданні розпізнавання символів.

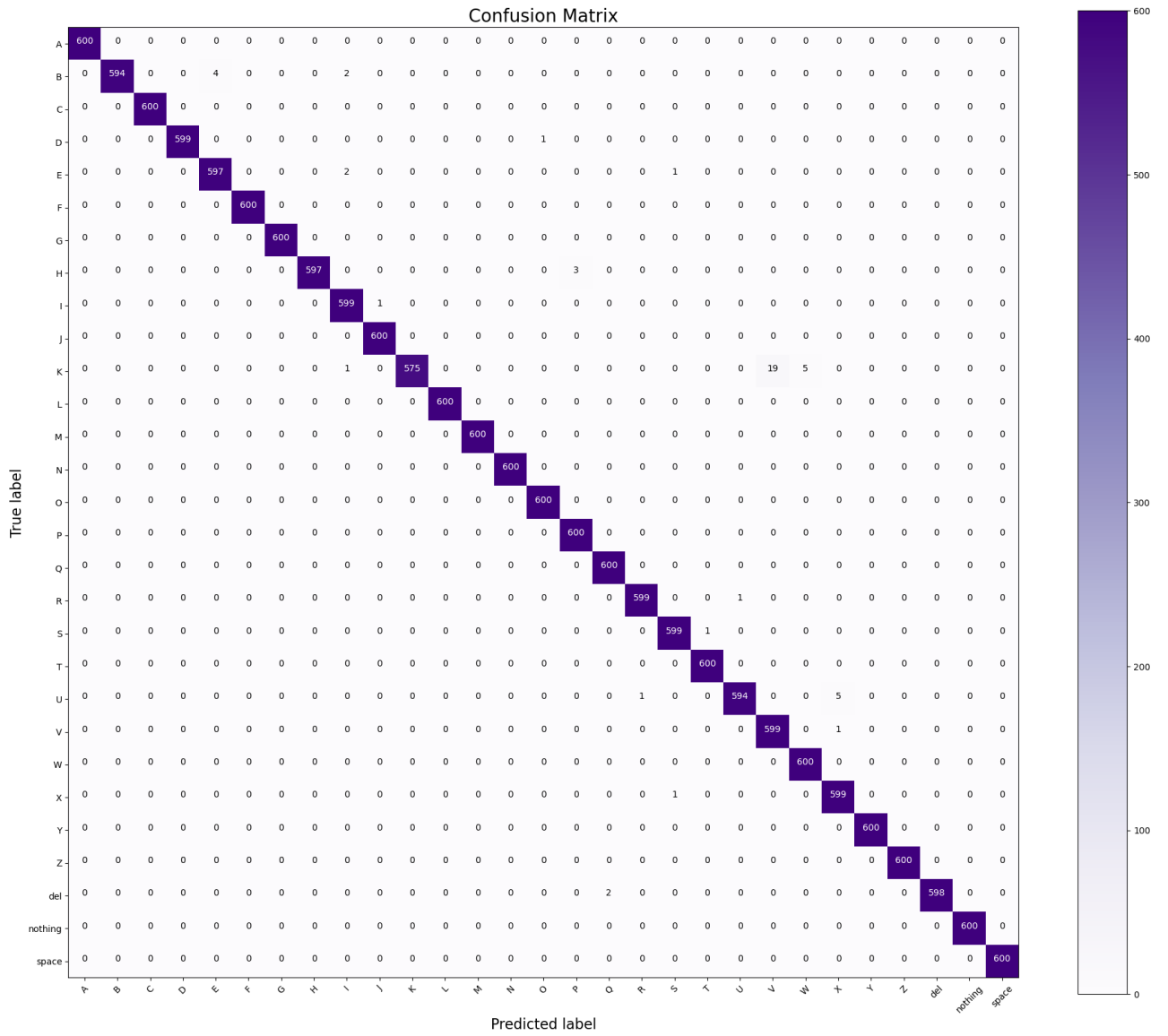


Рис. 3.5 – Конфузійна матриця

Як можна побачити на матриці найскладніші для розпізнавання символи у нашої навченої моделі – В та U.

3.2. Тестування моделі у режимі реального часу

Для впровадження тестування моделі в реальному часі використовувалися раніше згадані бібліотеки OpenCV та MediaPipe. OpenCV дозволила взаємодіяти з камерою та обробляти отримані зображення, забезпечуючи ефективну роботу моделі. З іншого боку, бібліотека MediaPipe наочно демонструвала процес відстеження жестів штучним інтелектом, роблячи його доступним для реального використання та сприяючи зрозумінню ефективності моделі в реальних умовах.

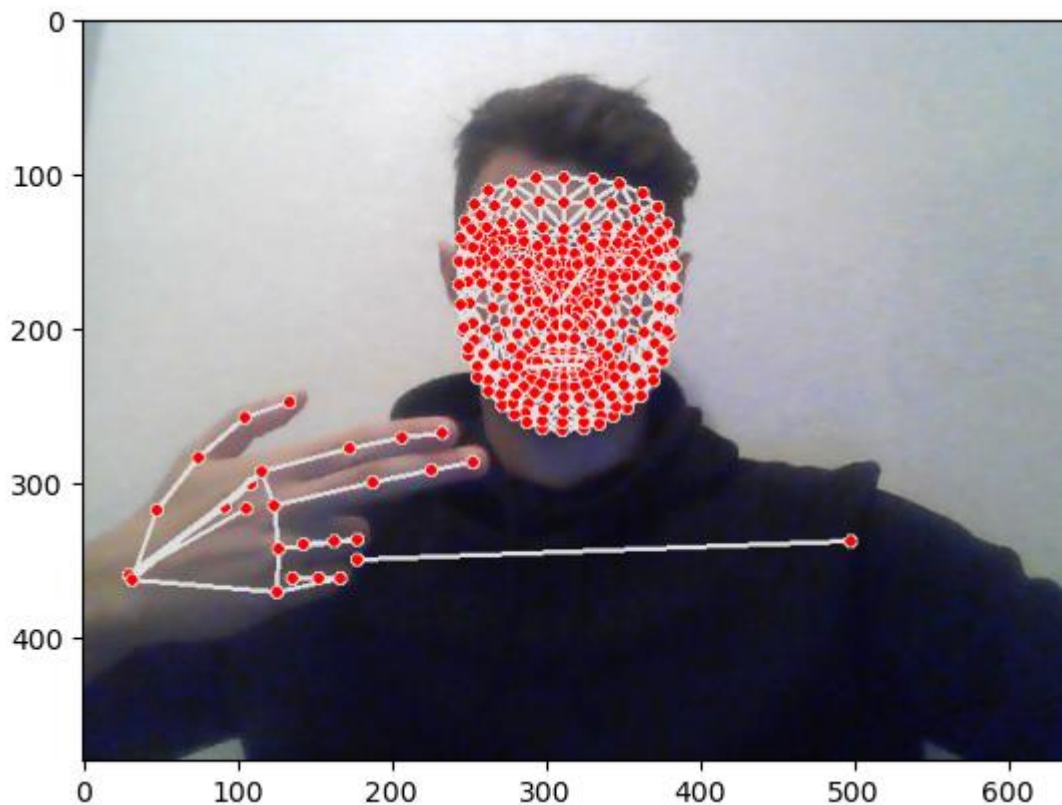


Рис. 3.6 – Відстеження ключових точок за допомогою бібліотеки mediapipe

У поєднанні з цими технологіями під час обробки кожного кадру програма подає фрагмент зображення на вхід навченої моделі. Після аналізу

отриманого фрагменту модель здійснює передбачення, а результат виводиться на екран. Цей процес відбувається в реальному часі.

Ось декілька прикладів найбільш розпізнаваних символів:

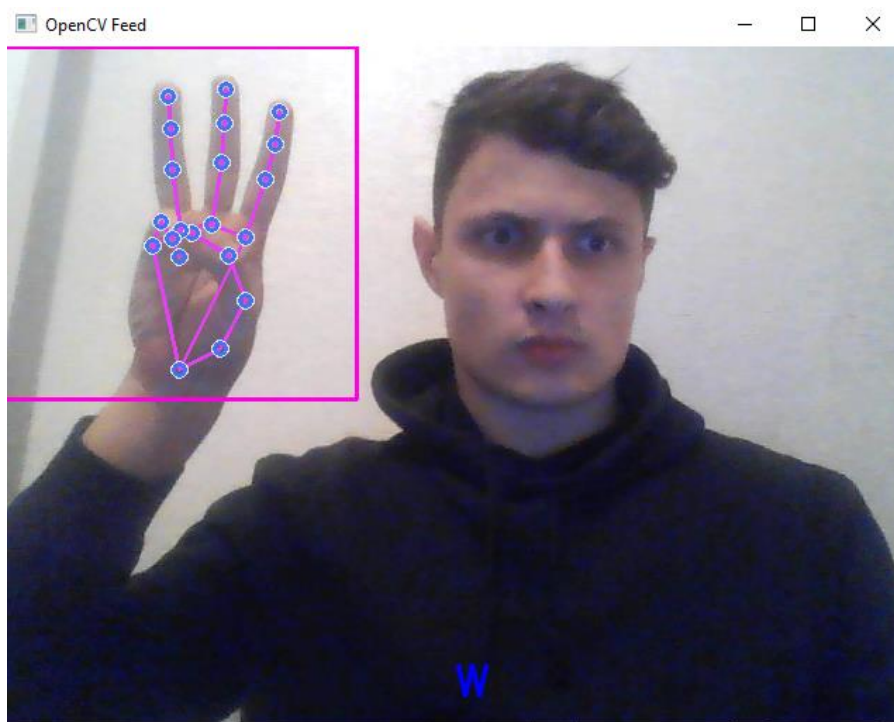


Рис. 3.7 – Приклад передбачення літери «W» в реальному часі

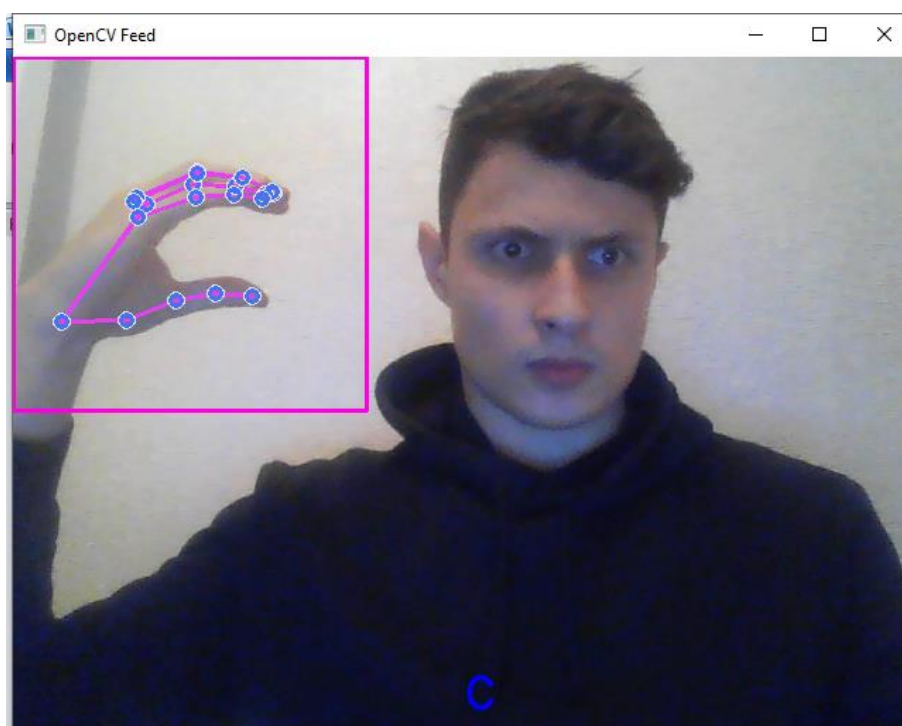


Рис. 3.8 – Приклад передбачення літери «С» в реальному часі

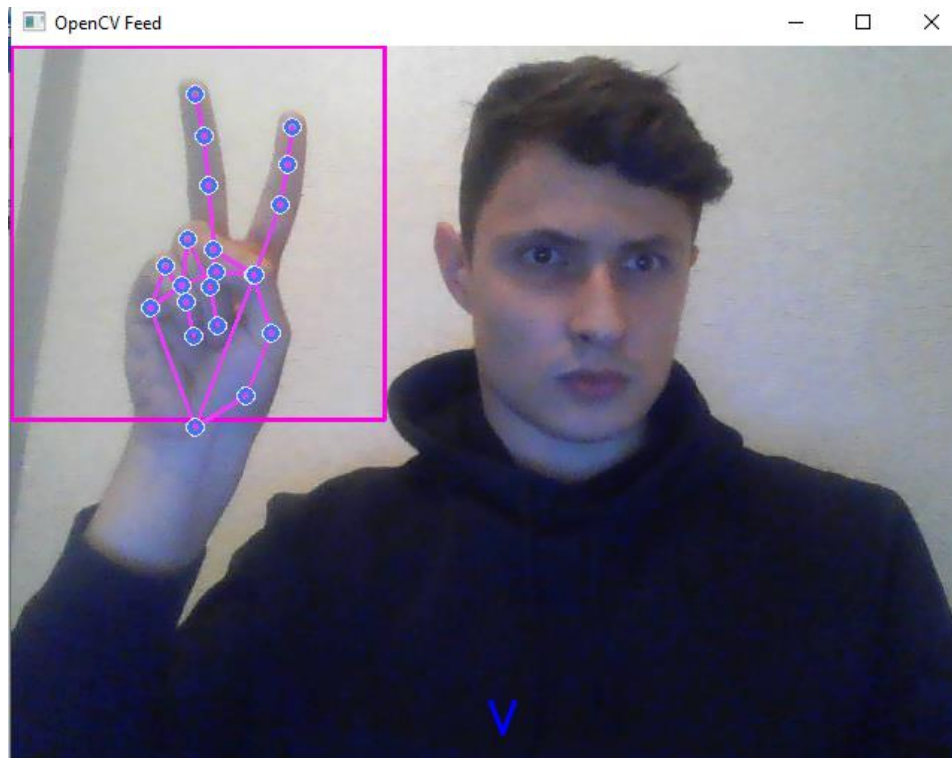


Рис. 3.9 – Приклад передбачення літери «V» в реальному часі

Під час проведення досліджень у реальному часі виявлено відмінності між тренувальною та фактичною точністю у режимі реального часу. Ці розбіжності можуть бути обумовлені кількома факторами:

- *Правильність демонстрації символів.* Некоректне чи нечітке представлення символів користувачем може призводити до помилок у їх розпізнаванні.

- *Положення руки.* Розташування руки може впливати на якість отриманого зображення символів. Зміни в куті або відстані можуть ускладнювати завдання розпізнавання.

- *Якість отриманого зображення.* Якість отриманого зображення може бути критичним фактором. Низька роздільна здатність чи нечіткість можуть ускладнювати завдання розпізнавання.

В майбутньому, на основі даних факторів можна покращити якість роботи даної моделі.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Оскільки навчання штучного інтелекту може виконуватися лише на комп'ютерах, тому варто дотримуватись нормативних документів, спрямованих на охорону праці користувачів ПК. Найбільш повним нормативним документом щодо забезпечення охорони праці користувачів ПК є «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин» (ДСанПН 3.3.2.007-98) [12].

Особлива увага звертається на те, що дотримання вимог, викладених в Правилах, значно знизить наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з відеодисплейними терміналами. У Правилах викладені гігієнічні й ергономічні вимоги до організації робочих приміщень та робочих місць, параметрів робочого середовища, дотримання яких дає змогу запобігти порушенням стану здоров'я користувачів ПК.

Користувачі ПЕОМ постійно працюють з обладнанням, яке підключено до електропостачання. При організації робочого місця потрібно ретельно перевіряти безпечність підключення до електромереж.

На користувачів під час роботи з комп'ютерною технікою можуть діяти такі види небезпек:

- ураження електричним струмом;
- енергетична небезпека (виникає через коротке замикання: опіки, електрична дуга, викид розплавленого металу);
- небезпека займання;
- термонебезпека (дія високих температур через нагрівання конструктивних елементів);
- механічна небезпека (травми через падіння, дії рухомих частин, поріз про гострі частини конструктивних елементів);

- небезпека випромінювання (дія звукового (акустичного), високочастотного, інфрачервоного, ультрафіолетового та іонізуючого випромінювання, а також видимого світла високої інтенсивності когерентної (лазерного випромінювання));
- хімічна небезпека (контакт із деякими хімікатами, які використовують для того, щоб обслуговувати обладнання, або від вдихання їхньої пари).

У процесі експлуатації електрообладнання може статися порушення цілісності ізоляції проводів, кабелів, обмоток машин та інших, які перебувають під напругою струмоведучих частин тобто, відбувається замикання струмовідних частин на землю. Це неухильно призводить до появи напруги на ненаправляючих частинах обладнання, і може призвести до ситуації, коли працівник опиниться під впливом електричного струму, що, в свою чергу, може викликати нещасний випадок.

Електробезпека обслуговуючого персоналу забезпечується системою організаційних та технічних заходів та засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля та статичної електрики. Одним із засобів захисту персоналу при спробі ізоляції є застосування захисного заземлення.

Замиканням на землю називається випадкове електричне з'єднання з землею, що знаходяться під напругою електроустановок. Замикання на землю може статися внаслідок появи контакту між струмовідними частинами та заземленим корпусом або конструктивними частинами обладнання при падінні на землю обірваного дроту, при порушенні ізоляції обладнання і т.д. У всіх цих випадках струм від частин, що знаходяться під напругою, проходить у землю через електроди, які здійснюють контакт із ґрунтом. Спеціальні металеві електроди прийнято називати заземлювачами.

Захисне заземлення – це навмисне з'єднання із землею за допомогою заземлювального пристрою металевих невідповідних частин обладнання, які можуть опинитися під напругою внаслідок порушення ізоляції електроустановки.

Відповідно до ДСанПІН 3.3.2.007-98 «Гігієнічні вимоги до персональних електронно-обчислювальних машин та організації роботи» приміщення, де розміщуються робочі місця з ПЕОМ, повинні бути обладнані захисним заземленням (зануленням) відповідно до технічних вимог щодо експлуатації електроустановок та обчислювальної техніки.

Відповідно до цього нормативного документа заземлювальні пристрої повинні бути обрані та змонтовані таким чином, щоб:

- значення опору розтіканню заземлювального пристрою відповідало вимогам захисту та роботи установки протягом періоду експлуатації;
- протікання струму замикання на землю і струмів витоку не створювало небезпеки, зокрема щодо нагрівання, термічної та динамічної стійкості;
- було забезпечено необхідну міцність або додатковий захист залежно від заданих зовнішніх факторів.

Робочі місця з ПЕОМ не слід розміщувати поблизу силових кабелів та вводів, високовольтних трансформаторів, технологічного обладнання, що створює перешкоди у роботі ПЕОМ.

Оскільки безпосередньо на ПЕОМ має подаватися стабілізоване електроживлення (з відхиленням від 220 В не більше -10% +15%), подачу електроенергії до комп'ютерних приміщень слід здійснювати від окремого незалежного джерела живлення.

Зазвичай підключення ПЕОМ здійснюється через блок живлення або пристрій живлення, що мають мережевий фільтр, конденсатори якого призначені для шунтування через провід занулення, і відповідні трисмугові вилку і розетку високочастотних перешкод мережі живлення на землю.

У цьому випадку до розетки повинні бути підключені три дроти: один фазний, другий нульовий робочий провідник і третій нульовий захисний провідник (НЗП). Нульовий захисний провідник необхідно з'єднати з нульовим дротом мережі.

В іншому випадку (якщо НЗП нікуди не підключати), на корпусі системного блоку може з'явитися напруга близько 110 змінного струму. Це станеться тому, що конденсатори фільтра працюють як ємнісний діляник напруги.

Оскільки ємності конденсаторів мають однакові значення, напруга мережі 220 В розділиться навпіл і, якщо врахувати, що середній опір тіла людини-1000 Ом, а опір статі (дерев'яного) та взуття близько-330 Ом, то сила струму, що проходить через тіло людини, становитиме 83 мА. При цьому можливе настання паралічу дихання [6].

Надалі при експлуатації ПЕОМ необхідно дотримуватись наступних рекомендацій:

- постійно контролювати надійність з'єднання контактів трипровідних розеток;
- додатково підключити системний блок до НЗП, наприклад закріпити провідник під гвинт кріплення джерела живлення;
- підключати дисплей (за наявності лише двопровідної однофазної мережі) рекомендується через узгоджувальний пристрій. При цьому мережні фільтри та всі кабелі живлення повинні знаходитися якнайдалі від оператора в компактному положенні з тильного боку робочого місця;
- не підключайте корпус комп'ютера до парового або водяного опалення. Якщо джерело живлення комп'ютера несправне, батареї можуть опинитися під напругою;
- не ставити системний блок у зоні підвищеної вологості та підвищеного вмісту пилу, на підлогу, біля ніг оператора;
- не можна торкатися одночасно екрана монітора та клавіатури (можливий підвищений електростатичний потенціал);
- щоб уникнути ураження електричним струмом, забороняється торкатися задній панелі системного блоку і перемикати роз'єми периферійних пристроїв працюючого комп'ютера;

- необхідно встановлювати ПЕОМ (ПК) тільки на жорстко закріпленій підставці, яка виключає навіть випадковий струс системного блоку;
- не рекомендується встановлення ПЕОМ та його клавіатури на поверхні, що накопичують статичну електрику (органічне скло та поліровані лакові поверхні);
- температура повітря в приміщенні допускається в межах 20-25 ° С при відносній вологості до 75%; різкі перепади температури не допускаються;
- не допускається зайва запиленість повітря в приміщенні (не більше 1 мг/м³ при максимальному розмірі частинок 3 мкм); обов'язкове вологе щоденне прибирання приміщення;
- необхідно щодня протирати вологою серветкою екран, приєкранний фільтр, клавіатуру та інші частини ПЕОМ.

Таким чином, розробка моделі машинного навчання для розпізнавання жестової мови, буде виконана з вимог техніки безпеки.

4.2. Безпека в надзвичайних ситуаціях

Оскільки розробка та навчання моделей штучного інтелекту, потребує багато комп'ютерних ресурсів, то за для ефективної розробки такого продукту, всі девайси повинні розміщатися в одному місці. Приміщення із робочими місцями користувачів комп'ютерів для забезпечення електробезпеки обладнання, а також для захисту від ураження електричним струмом самих користувачів ПК повинні мати достатні технічні засоби захисту відповідно до ДНАОП 1.1.10-1.1.01.97 “ Правила безпечної експлуатації електроустановок”, НПАОП 40.1-1.32-01 “Правила будови електроустановок. Електрообладнання спеціальних установок”.

З метою запобігання ушкодженням, що можуть статися через ураження електричним струмом, загоряння, коротке замикання тощо, розроблено

загальний стандарт безпеки IEC 950. Загальним стандартом електробезпечності для країн Європейської співдружності є Semark.

Під час проектування систем електропостачання, монтажу силового електрообладнання та електричного освітлення будівель та приміщень для ПЕОМ необхідно дотримуватись вимог вищеназваних нормативно-правових актів, а також Правил пожежної безпеки в Україні, ДСанПіН 3.3.2.007-98, розділів ДБН, що стосуються штучного освітлення і електротехнічних пристроїв, та вимог нормативно-технічної і експлуатаційної документації заводу-виробника ПЕОМ.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники тощо), електропроводи та кабелі за виконанням та ступенем захисту мають відповідати класу зони за ПУЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, перейти на негорючу ізоляцію.

Лінія електромережі для живлення ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ виконується як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів.

Використання нульового робочого провідника як нульового захисного провідника забороняється. Нульовий захисний провід прокладається від стійки групового розподільчого щита, розподільчого пункту до розеток живлення. Не допускається підключення на щиті до одного контактного затискача нульового

робочого та нульового захисного провідників. Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі повинна бути не менше площі перерізу фазового провідника.

Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ПУЕ.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ПЕОМ, периферійні пристрої ПЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

Неприпустимим є підключення ПЕОМ та периферійних пристроїв ПЕОМ до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення ПЕОМ, периферійних пристроїв слід виконувати за магістральною схемою, по 3...6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 36 В за своєю конструкцією повинні відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В і мають бути

пофарбовані в колір, який візуально значно відрізняється від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах з урахуванням вимог ПУЕ та Правил пожежної безпеки в Україні.

Електромережу штепсельних розеток для живлення ПЕОМ, периферійних пристроїв ПЕОМ при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

При розташуванні в приміщенні за його периметром до 5 ПЕОМ, використанні трипровідникового захищеного проводу або кабелю в оболонці з негорючого або важкогорючого матеріалу дозволяється прокладання їх без металевих труб та гнучких металевих рукавів.

Електромережу штепсельних розеток для живлення ПЕОМ при розташуванні їх у центрі приміщення, прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не дозволяється застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, що містять сірку. Відкрита прокладка кабелів під підлогою забороняється. Металеві труби та гнучкі металеві рукави повинні бути заземлені. Заземлення повинно відповідати вимогам НПАОП 40.1-1.21-98 [7].

Для підключення переносної електроапаратури застосовують гнучкі проводи в надійній ізоляції.

Тимчасова електропроводка від переносних приладів до джерел живлення виконується найкоротшим шляхом без заплутування проводів у конструкціях машин, приладів та меблях. Доточувати проводи можна тільки шляхом паяння з наступним старанним ізолюванням місць з'єднання.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією; залишення під напругою кабелів та проводів з неізольованими провідниками;
- застосування саморобних подовжувачів, які не відповідають вимогам ПВЕ до переносних електропроводок;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
- використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

Таким чином розробка моделі штучного навчання буде безпечна в плані запобігання надзвичайних ситуацій.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була досліджена модель штучного інтелекту для розпізнавання жестової мови. Під час дослідження було проведено аналітику предметної області, розглянуто конкретні технології для розробки моделей штучного інтелекту, а також наочно продемонстровано роботу моделі. Вихідним продуктом даного дослідження є готова модель глибокого навчання, що розпізнає символи мови жестів ASL у режимі реального часу.

В аналітичній частині було розглянуто актуальність можливості розпізнавання жестової мови, перспективи використання штучного інтелекту для вирішення даної проблеми. Також було проаналізовано можливі алгоритми машинного навчання, які можна використати для розробки. Останнім пунктом варто було зазначити важливість підбору та обробки даних. Результатом даного розділу є чітко визначена постановка задачі.

В наступному розділі увагу було сконцентровано увагу на огляді технологічних застосунків використаних у програмі, а також механізмах роботи конкретних алгоритмів глибинного навчання моделі штучного інтелекту.

В розділі про результати розробки та тестування моделі, було продемонстровано конкретні налаштування та статистичні дані, що стосуються безпосередньої розробки та навчання штучного інтелекту. На етапі тестування модель перевірялася на здатність відслідковувати жести в режимі реального часу.

Дане дослідження має потенціал для удосконалення. В подальшому із необхідною кількістю ресурсів продукт можна розширити у повнофункціональну програму із розпізнавання жестової мови із повноцінним інтерфейсом, що допоможе людям із вадами мовлення спілкуватися з іншими людьми без жодних труднощів, лише використовуючи мобільний додаток у своєму смартфоні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жестова мова – [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%96%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0
2. Факти про мову жестів – [Електронний ресурс] – Режим доступу до ресурсу: <https://proit.org.ua/naipopuliarnishi-bibliotieki-shi-ta-mashinnogho-navchannia-python/>
3. What is American sign language? – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.nidcd.nih.gov/health/american-sign-language>
4. Найпопулярніші бібліотеки ШІ та машинного навчання Python – [Електронний ресурс] – Режим доступу до ресурсу: <https://proit.org.ua/naipopuliarnishi-bibliotieki-shi-ta-mashinnogho-navchannia-python/>
5. Що таке глибоке навчання? – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.unite.ai/uk/>
6. Нейронні мережі та глибоке навчання – [Електронний ресурс] – Режим доступу до ресурсу: <https://evergreens.com.ua/ua/articles/neural-network.html>
7. Types of Neural Networks and Definition of Neural Network – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mygreatlearning.com/blog/types-of-neural-networks/>
8. Gesture Recognition in Deep Learning: A Leap into the Future – [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@evertongomede/gesture-recognition-in-deep-learning-a-leap-into-the-future-46f2d40b659b>
9. Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network – [Електронний ресурс] –

Режим доступу до ресурсу: <https://www.nature.com/articles/s41598-023-43852-x>

- 10.Tensorflow – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/>
- 11.Python – [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>
- 12.ДСанПІН 3.3.2.007-98 [Електронний ресурс]. – 1998. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>.
- 13.Мюллер А., Гвідо С. Введення в машинне навчання за допомогою Python. 480 с.
- 14.Жидецький В.Ц. Охорона праці користувачів комп'ютерів. – Львів: Афіша, 2000.– 176с.
- 15.Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (Освітньо-професійна програма - «Програмне забезпечення систем», Освітньо-наукова програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.: М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк - Тернопіль: ТНТУ, 2020- 51с.
- 16.І.Я. Мудрик, Способи комп'ютеризованого збору та аналізу параметрів аномальних неврологічних рухів. Матеріали Міжнародної науково-технічної конференції „Фундаментальні та прикладні проблеми сучасних технологій“ ТНТУ ім.І.Пулюя. 2018/5/22, 221-222.
- 17.«БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» / В.С. Стручок – Тернопіль: ФОП Паляниця В. А., –156 с.
- 18.Навчальний посібник «ТЕХНОЕКОЛОГІЯ ТА ЦИВІЛЬНА БЕЗПЕКА. ЧАСТИНА «ЦИВІЛЬНА БЕЗПЕКА»» / автор-укладач В.С. Стручок– Тернопіль: ФОП Паляниця В. А., – 156 с.

ДОДАТКИ

ДОДАТОК А

Публікація в науковому виданні

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

ТЕРНОПЛЬ
2023

УДК 004.056

О. Редчук

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

**ДОСЛІДЖЕННЯ ЗДАТНОСТІ ШТУЧНОГО ІНТЕЛЕКТУ ДО
РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ**

O.Redchuk

**RESEARCH ON THE ABILITY OF ARTIFICIAL INTELLIGENCE TO
RECOGNIZE SIGN LANGUAGE**

Штучний інтелект (ШІ) є ключовим каталізатором інновацій у світі сучасних технологій, розширюючи межі можливостей для взаємодії між людьми та комп'ютерами. Одним з найважливіших та високотехнологічних областей в цьому контексті є дослідження здатності штучного інтелекту до розпізнавання жестової мови. Це напрямок вивчення на перетині мовознавства, комп'ютерної науки та психології зосереджений на створенні інтерфейсів, що можуть ефективно спілкуватися з людьми за допомогою жестів і рухів тіла.

Сучасні технології вже дозволяють штучному інтелекту аналізувати та розпізнавати різноманітні жести, відкриваючи перед нами нові перспективи для покращення способу взаємодії з комп'ютерами. Дослідження в цьому напрямку не тільки розширюють можливості комунікації, але також відкривають шляхи для подальшого розвитку у сфері адаптивних технологій, що враховують різні стилі та культурні особливості жестової мови.

Узагальнюючи сказане, здатність ШІ адаптуватися до різноманітних жестів відкриває шлях до створення інноваційних інтерфейсів. Ці інтерфейси не лише піднімають взаємодію між людьми та технологіями на новий рівень, але також стимулюють розвиток нових концепцій та підходів до створення ефективних інтерактивних систем. Такі досягнення в сфері розпізнавання жестів обіцяють надавати значний внесок у покращення якості життя та розвиток інновацій в ряді сфер, включаючи освіту, медицину та розваги.

Література

1. Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network. URL: <https://www.nature.com/articles/s41598-023-43852-x>

2. Що таке глибоке навчання? URL: <https://www.unite.ai/uk>

СЕКЦІЯ 5. НОВІТНІ ФІЗИКО-ТЕХНІЧНІ ТА ОСВІТНІ ТЕХНОЛОГІЇ

- I. Белікова, V. Медвідь, O. Шовкун**
 ЕКСПЕРИМЕНТАЛЬНА УСТАНОВКА ДЛЯ ДОСЛІДЖЕННЯ ТЕМПЕРАТУРНОЇ ЗАЛЕЖНОСТІ ЕЛЕКТРИЧНИХ ТА СВІТЛОВИХ ПАРАМЕТРІВ СВІТЛОДІОДІВ ТА СВІТЛОДІОДНИХ ДЖЕРЕЛ СВІТЛА
I. Beliakova, V. Medvid, O. Shovkun
 EXPERIMENTAL SETUP FOR STUDYING THE TEMPERATURE DEPENDENCE OF ELECTRICAL AND LIGHT PARAMETERS OF LEDS AND LED LIGHT SOURCES 228
- Ю.Гладько, Н.Гашин, Н.Крива**
 ІНЖЕНЕРНА МЕТОДИКА РОЗРАХУНКУ ІНДУКЦІЙНОГО НАГРІВАЧА
Yu. Gladko, N.Gashchyn, N.Kryva
 ENGINEERING METHODOLOGY OF CALCULATION OF AN INDUCTION HEATER 229
- Володимир Кочан, Вячеслав Лозін, Роман Осідак**
 АД-НОС ВИЗНАЧЕННЯ АМПЛІТУДНО-ЧАСТОТНИХ ХАРАКТЕРИСТИК ЕЛЕКТРОННИХ ВУЗЛІВ
Volodymyr Kochan, Vyacheslav Lozin, Roman Osidak
 AD-HOC DETERMINATION OF FREQUENCY RESPONSE OF ELECTRONIC UNITS 231
- Ігор Коноваленко, Михайло Бучинський, Володимир Зубко, Павло Марущак**
 ВПЛИВ ПАРАМЕТРІВ ОСВІТЛЕННЯ НА РЕЗУЛЬТАТИВНІСТЬ АВТОМАТИЗОВАНОГО ДЕФЕКТОМЕТРИЧНОГО КОНТРОЛЮ
Ihor Konovalenko, Mykhailo Buchynskiy, Volodymyr Zubko, Pavlo Maruschak
 THE INFLUENCE OF LIGHTING PARAMETERS ON PERFORMANCE OF AUTOMATED DEFECTOMETRICAL CONTROL 232
- Р.В. Материн, І.О. Баран**
 КОНЦЕПЦІЯ НАСКРІЗНОГО ПРОЕКТУВАННЯ НАВІГАЦІЙНОГО МОДУЛЯ
R.V. Materyn, I.O. Baran
 NAVIGATION MODULE END-TO-END DESIGN CONCEPT 233
- В. Ю. Остаполец, М.В. Деркач**
 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ З ВИКОРИСТАННЯМ ГЕОІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
V.Y. Ostapolets, M.V. Derkach
 DEVELOPMENT OF A MOBILE APPLICATION USING GEOINFORMATION TECHNOLOGIES 234
- А.І. Очерклевич, В.М. Сенюк, Л.Н. Бодрова**
 КОНСТРУКТИВНІ РІШЕННЯ КАРКАСІВ ПРОМІСЛОВИХ БУДІВЕЛЬ
A.I. Ocherklevych, V.M. Senyuk, L.N. Bodrova
 STRUCTURAL ENGINEERING SOLUTIONS FOR FRAMES OF INDUSTRIAL BUILDINGS 235
- О. Редчук**
 ДОСЛІДЖЕННЯ ЗДАТНОСТІ ШТУЧНОГО ІНТЕЛЕКТУ ДО РОЗПІЗНАВАННЯ ЖЕСТОВОЇ МОВИ
O. Redchuk
 RESEARCH ON THE ABILITY OF ARTIFICIAL INTELLIGENCE TO RECOGNIZE SIGN LANGUAGE 236

ДОДАТОК Б

Лістинг коду:

Підключення бібліотек:

```
from numpy.random import seed
#!pip install tensorflow==1.14
import cv2
from sklearn.model_selection import train_test_split
import os
import numpy as np
import skimage
from skimage.transform import resize
import csv
import itertools
from tqdm import tqdm
from glob import glob
from matplotlib import pyplot as plt
from numpy import floor
import random
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
#from keras.layers import Conv2D, Dense, Dropout, Flatten
#from keras.models import Sequential
#from keras.preprocessing.image import ImageDataGenerator

%pip install mediapipe --user
import mediapipe as mp
```

Підготовка даних:

```
def load_images(directory):
    images = []
    labels = []
    for idx, label in enumerate(uniq_labels):
        print(label, " is ready to load")
        for file in os.listdir(directory + "/" + label):
            filepath = directory + "/" + label + "/" + file
            image = cv2.resize(cv2.imread(filepath), (64, 64))
            images.append(image)
            labels.append(idx)
    images = np.array(images)
    labels = np.array(labels)
    return(images, labels)
```

```
import keras

train_dir = "./input/asl_alphabet/asl_alphabet_train/asl_alphabet_train"

uniq_labels = sorted(os.listdir(train_dir))

images, labels = load_images(directory = train_dir)
print("Data has been loaded")
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size = 0.2, stratify = labels)

n = len(uniq_labels)
train_n = len(X_train)
test_n = len(X_test)

print("Total number of symbols: ", n)
print("Number of training images: ", train_n)
print("Number of testing images: ", test_n)
```

Налаштування моделі:

```
y_train = keras.utils.to_categorical(y_train)
y_test = keras.utils.to_categorical(y_test)
```

```
X_train = X_train.astype('float32')/255.0
X_test = X_test.astype('float32')/255.0
```

```

from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Conv2D, Dense, Dropout, Flatten
from keras.layers import Flatten, Dense
from keras.models import Sequential

model = Sequential()
model.add(Conv2D(filters = 64, kernel_size = 5, padding = 'same', activation = 'relu', input_shape = (64, 64, 3)))
model.add(Conv2D(filters = 64, kernel_size = 5, padding = 'same', activation = 'relu'))
model.add(MaxPooling2D(pool_size = (4, 4)))
model.add(Dropout(0.5))
model.add(Conv2D(filters = 128 , kernel_size = 5, padding = 'same', activation = 'relu'))
model.add(Conv2D(filters = 128 , kernel_size = 5, padding = 'same', activation = 'relu'))
model.add(MaxPooling2D(pool_size = (4, 4)))
model.add(Dropout(0.5))
model.add(Conv2D(filters = 256 , kernel_size = 5, padding = 'same', activation = 'relu'))
model.add(Dropout(0.5))
model.add(Flatten())
model.add(Dense(29, activation='softmax'))

model.summary()

```

Навчання моделі:

```

hist = model.fit(X_train, y_train, epochs = 5, batch_size = 64)

```

Зберігання моделі для економії часу:

```

model.save('mymodel.h5')

```

```

saved_model = keras.models.load_model('./mymodel.h5')

```

Перевірка точності моделі:

```

score = saved_model.evaluate(x = X_test, y = y_test, verbose = 0)
print('Accuracy for test images:', round(score[1]*100, 3), '%')

```

Побудова конфузійної матриці:

```
def plot_confusion_matrix(y, y_pred):
    y = np.argmax(y, axis = 1)
    y_pred = np.argmax(y_pred, axis = 1)
    cm = confusion_matrix(y, y_pred)
    plt.figure(figsize = (24, 20))
    ax = plt.subplot()
    plt.imshow(cm, interpolation = 'nearest', cmap = plt.cm.Purples)
    plt.colorbar()
    plt.title("Confusion Matrix")
    tick_marks = np.arange(len(uniq_labels))
    plt.xticks(tick_marks, uniq_labels, rotation=45)
    plt.yticks(tick_marks, uniq_labels)
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    ax.title.set_fontsize(20)
    ax.xaxis.label.set_fontsize(16)
    ax.yaxis.label.set_fontsize(16)
    limit = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], 'd'), horizontalalignment = "center", color = "white" if cm[i, j] > limit else "black")
    plt.show()
```

```
from sklearn.metrics import confusion_matrix
import itertools

y_test_pred = saved_model.predict(X_test, batch_size = 64, verbose = 0)
plot_confusion_matrix(y_test, y_test_pred)
```

Тестування в режимі реального часу:

```
mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities
```

```
def mediapipe_detection(image, model):
    '''Detect the landmarks of the image using the mediapipe model'''
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False # Image is no longer writeable
    results = model.process(image) # Make prediction
    image.flags.writeable = True # Image is now writeable
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR CONVERSION RGB 2 BGR
    return image, results
```

```
def draw_landmarks(image, results):
    '''Function to draw landmarks from the detected image'''
    mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_TESSELATION) # Draw face connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS) # Draw pose connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS) # Draw right hand connections
```

```

def draw_styled_landmarks(image, results):
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250), thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
                              )

```

```

labels = {0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'I', 9: 'J', 10: 'K', 11: 'L', 12: 'M', 13: 'N', 14: 'O',
          15: 'P', 16: 'Q', 17: 'R', 18: 'S', 19: 'T', 20: 'U', 21: 'V', 22: 'W', 23: 'X', 24: 'Y', 25: 'Z', 26: 'del', 27: 'nothing', 28: 's

cap = cv2.VideoCapture(0)
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
    while True:
        isTrue, frame = cap.read()

        if not isTrue:
            break

        image, results = mediapipe_detection(frame, holistic)
        print(results)

        # Draw landmarks with styled connections
        draw_styled_landmarks(image, results)

        img = frame[:250, :250]
        img = cv2.resize(img, (64, 64,))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = np.array(img).reshape((1, 64, 64, 3))

```

```

res = saved_model.predict(img)[0]
print(res)
n = np.argmax(res)
print(res[n], labels[n])
print(frame.shape)
h, w, c = frame.shape
cv2.rectangle(image, (0, 0), (250, 250), (225, 0, 255), 2)
cv2.putText(image, labels[n], (w//2, h-20),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)

draw_styled_landmarks(image, results)
cv2.imshow('OpenCV Feed', image)

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

cap.release()
cv2.destroyAllWindows()

```

ДОДАТОК В

