

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Кафедра теоретичної кібернетики


Бакалаврська робота

за спеціальністю 122 Комп'ютерні науки

на тему:

**ДОСЛІДЖЕННЯ ПІДХОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ
РОЗПІЗНАВАННЯ ЖЕСТІВ**

Виконав студент 4-го курсу
Чепой Олександр Михайлович



(підпис)

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

Студент



(підпис)

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



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної кібернетики
« 1 » червня 2022 р.,
протокол № 11

Завідувач кафедри

доктор фіз.-мат. наук, професор
Юрій КРАК



(підпис)

Київ - 2022

РЕФЕРАТ

Обсяг роботи 43 сторінки, ілюстрацій 8, джерел посилань 18.

Ключові слова: РОЗПІЗНАВАННЯ ЖЕСТІВ, МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ, ФУНКЦІЯ АКТИВАЦІЇ, НЕЙРОННІ МЕРЕЖІ

Об'єктом дослідження є підходи машинного навчання для розпізнавання жестів.

Метою роботи є дослідження та порівняння різних підходів комп'ютерного розпізнавання жестів

Результати роботи: досліджені сучасні підходи машинного навчання у задачі розпізнавання жестів, а саме проведено детальний аналіз та порівняння технологій та шляхів розв'язання проблем, що стоять на різних етапах розпізнавання жестів.

ЗМІСТ

РЕФЕРАТ	2
ЗМІСТ	3
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ЩО ТАКЕ МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ	8
ПРИКЛАДНЕ ВИКОРИСТАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ	9
ІСТОРІЯ РОЗВИТКУ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ	10
ЩО ТАКЕ НЕЙРОННІ МЕРЕЖІ ТА ГЛИБОКЕ НАВЧАННЯ	13
4.1 ВИЗНАЧЕННЯ НЕЙРОННОЇ МЕРЕЖІ	13
4.2 НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ	15
4.3 ТИПИ НЕЙРОМЕРЕЖ І ЇХ КЛАСИФІКАЦІЯ	16
4.3.1 ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ	18
4.3.2 РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ	18
4.4 ЗАВДАННЯ, ЯКІ ВИРІШУЮТЬСЯ НЕЙРОННИМИ МЕРЕЖАМИ	19
ЩО ТАКЕ КОМП'ЮТЕРНИЙ ЗІР	20
5.1 ЯК ПРАЦЮЄ КОМП'ЮТЕРНИЙ ЗІР	21
5.2 ІСТОРІЯ РОЗВИТКУ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ	23
5.3 ДЕ ЗАСТОСОВУЄТЬСЯ ТЕХНОЛОГІЯ КОМП'ЮТЕРНОГО ЗОРУ	24
МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ ТА НЕЙРОМЕРЕЖ	24
6.1 ЗБІР ЗОБРАЖЕНЬ	25
6.2 СЕГМЕНТАЦІЯ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ	26
6.3 РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЯ ЖЕСТІВ	26
6.3.1 РОЗПІЗНАВАННЯ ЖЕСТУ НА ОСНОВІ НЕЙРОМЕРЕЖ	27
СУЧАСНИЙ СТАН РОЗВИТКУ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ ТА НЕЙРОМЕРЕЖ	29
OPENSOURCE РІШЕННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ	30
8.1 TensorFlow	30

	4
8.2 OpenCV	31
8.3 MediaPipe	32
ПРИКЛАД ТА ПРОГРАМНА РЕАЛІЗАЦІЯ	33
9.1 НЕОБХІДНІ УМОВИ	33
9.2 ПРОГРАМНИЙ КОД	34
9.3 РЕЗУЛЬТАТ	39
ПРОБЛЕМИ	42
ВИСНОВКИ	43
ДЖЕРЕЛА	44

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

CNN – convolutional neural network, згорткова нейронна мережа

RNN – recurrent neural network, рекурентна нейронна мережа

AI(ШІ) – Artificial Intelligence, штучний інтелект

HCI – Human-computer interaction

ANN – Artificial neural network

OpenCV – Open Source Computer Vision Library

GAN – Generative Adversarial Nets

ВСТУП

За останнє сторіччя інформаційні та комп'ютерні технології розвинулись дуже сильно. Людство змогло вмістити машини, що займали обсяг цілої будівлі до розмірів пристрою що можна покласти у кишеню або ж використовувати замість наручного годинника. Ще більш дивує те, що сучасні, набагато менші пристрої мають обчислювальну потужність набагато більшу ніж їх предки що займали обсяг цілих будівель.

Окрім мінімізації та збільшення сирової обчислювальної потужності, розвинулась також ціла низка периферійних пристроїв, що значно розширило можливості комп'ютерів взаємодіяти із зовнішнім світом та людині взаємодіяти із комп'ютером. В купі, це все зробило комп'ютери дуже універсальним інструментом, що можна використовувати для величезної кількості задач. В решті решт, було дуже мало шансів, що потенціал такого інструменту не будуть використовувати:

- Різні компанії почали використовувати комп'ютери для автоматизації та оптимізації своїх процесів.
- Науковці та інженери використовують комп'ютер для обчислень, моделювання та проектування.
- Персональні комп'ютери є майже в кожній хаті, оскільки це незамінний інструмент для навчання, отримання інформації та дозвілля.
- Комп'ютерні технології використовують в таких галузях як військова галузь, надання безпеки у публічних місцях та медицина.

Таке широке використання комп'ютерів призвело до появи багатьох нових алгоритмів, технологій, шляхів використання комп'ютерів, і розвиток у

багатьох напрямках все ще продовжується. Це все призвело до того що ми живемо у світі в якому Інтернет, смартфони, ПК, IoT є невіддільною частиною життя багатьох людей, а XXI століття називають століттям інформаційних технологій.

Серед технологій та алгоритмів які зараз активно розвиваються можна відзначити алгоритми та методи розпізнавання жестів. В цьому дослідженні розглянемо їх детальніше.

Мета й завдання роботи: метою роботи є дослідження та порівняння різних підходів комп'ютерного розпізнавання жестів. Для досягнення цього завдання було покладено такі цілі роботи.

- 1) Визначити підстави появи технології розпізнавання жестів та її використання.
- 2) Провести порівняльний аналітичний огляд наявних методів захоплення, відстеження та розпізнавання жестів людини.
- 3) На основі досліджених методів реалізувати свою систему розпізнавання.

1. ЩО ТАКЕ МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ

Жести – це форма безсловесної комунікації за допомогою видимих рухів тіла, що передають якусь інформацію та повідомлення замість усного мовлення або разом із ним. Жестами зазвичай є певні рухи рук, лиця або ж інших частин тіла.

Алгоритми розпізнавання жестів — це сукупність інформаційних технологій та алгоритмів, призначених для того, щоб розпізнавати, ідентифікувати та класифікувати жести людини. Як вхідні дані для цих алгоритмів може виступати зображення, відеофайли, відеопотік із цифрових камер або ж потік даних зі спеціалізованих пристроїв. Як вихід, отримуємо жести які були показані у джерелі даних та, можливо, інтерпретація цих жестів. Вже розроблено багато підходів із використанням камер та алгоритмів комп'ютерного зору для інтерпретації мови жестів. Однак ідентифікація та розпізнавання пози, ходьби та поведінки людини також є об'єктом дослідження технік розпізнавання жестів.

2. ПРИКЛАДНЕ ВИКОРИСТАННЯ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ

Цю технологію можна розглядати як потенційно новий спосіб комунікації людини та комп'ютера, що дозволяє машині розуміти мову тіла людини. Використовуючи цю технологію можна багатшу систему взаємодії із комп'ютером, що потенційно є багатшою, ніж класична взаємодія за допомогою клавіатури та миші через GUI(графічний інтерфейс користувача).

Окрім взаємодії із ПК, технологію розпізнавання жестів можна використовувати для керування IoT системами та розумними домами. Наприклад вмикати та вимикати світло за допомогою жестів, керувати температурою або вологістю повітря, зачиняти або відчиняти замки або ж керувати іншими пристроями в будинку.

Слід зазначити що такі рівні автоматизації можуть бути запровадженні не тільки в цивільних домах, а й на виробництвах, в офісах, на майданчиках тощо.

Цю технологію можна використовувати також для збагачення взаємодії із транспортними засобами. Наприклад це дозволить водіям керувати іншими системами в автомобілі, такими як опалення та охолодження, а також підключатися до систем розумного дому. Наприклад, уявіть, що ви можете перевірити свою домашню камеру безпеки, коли ви їдете додому, просто зробивши жест рукою.

З жестів можна отримати багато інформації про людину яка їх робить: емоції, стан тощо.

Технологію розпізнавання жестів можна використовувати у системах що можуть перекладати з мови жестів.

3. ІСТОРІЯ РОЗВИТКУ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ

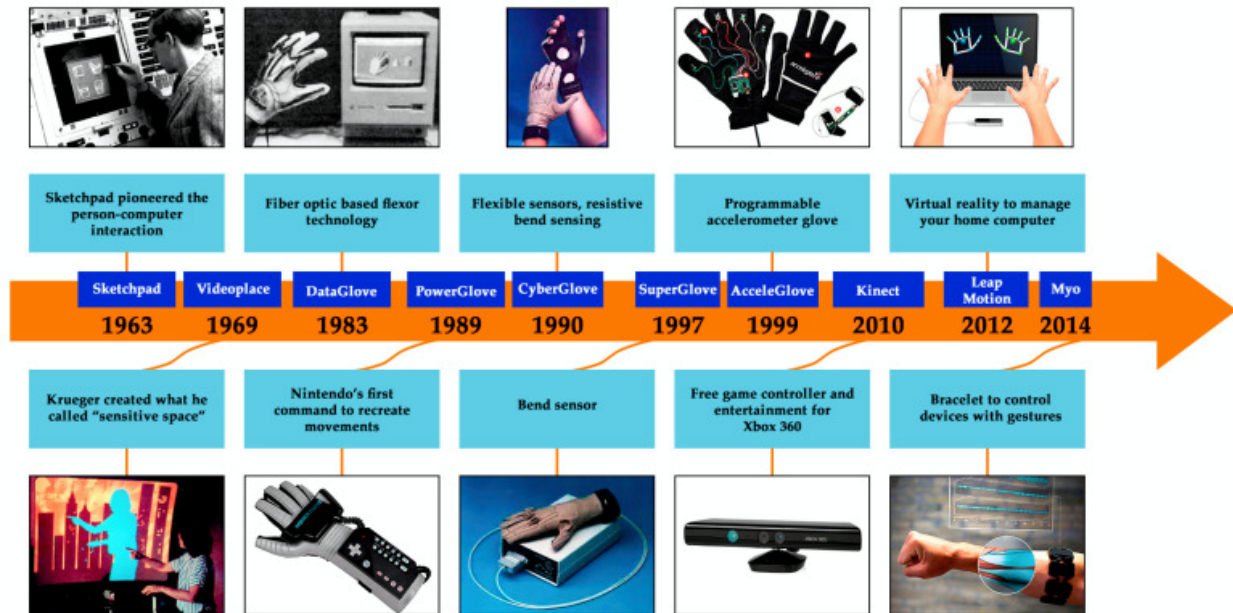


Рисунок 1 – розвиток методів розпізнавання жестів.

Як зазначалося вище, з точки зору взаємодії людини з машиною, розпізнавання жестів вважається найбільш інтуїтивним і природним, тому його розвиток постійно розвивається, залежно від удосконалення датчиків, які використовуються для фіксації жестів. У цьому сенсі розпізнавання жестів по суті еволюціонувала від інтуїтивного розпізнавання до більш формального розпізнавання, заснованого на вдосконаленні експериментів із датчиками, які використовуються для цієї мети.

Спочатку, протягом 1960-х років, дослідники почали використовувати планшети та спеціальні олівці, які фіксували письмо, сенсорні інтерфейси або вказівні пристрої. Пізніше, в 1969 році, інженер Мирон Крюгер почав працювати з прототипами віртуальної реальності та стверджував, що

майбутнє можливе без екранів, в яких люди будуть безпосередньо взаємодіяти з навколишнім середовищем. Цей підхід пізніше буде називатися природною взаємодією, а ці інтерфейси взаємодії — природними інтерфейсами користувача (NUI).

Лише в 1980-х роках почнеться фіксація руху за допомогою рукавичок з датчиками згинання та положення. Ці пристрої зазвичай носять на руках і вони обладнані датчиками для збирання даних про положення у просторі, даних про згинання фаланг пальців, суглоб та зап'ясть. Всі ці дані вже потім можуть бути передані комп'ютеру для обробки. Цей тип пристроїв можна поділити на активні та пасивні.

Активні: зазвичай рукавички з електронними датчиками, які поєднанні з комп'ютером через кабель або ж бездротове з'єднання та активно передають дані зі сенсорів.



Рисунок 2 – активна рукавичка версії NN-2020-MXA

Пасивні: рукавички що як правило не мають електронних датчиків. Натомість вони мають певні кольорові маркери через яких вже інші пристрої визначають положення руки у просторі.



Рисунок 3 – пасивні рукавички версії PASS-2017-LLA

У 1990-х роках розширилися роботи з визначення жестів у зображеннях і відео за допомогою методів комп'ютерного зору, що вдосконалюється до сьогодні, коли доступні системи відстеження людей у реальному часі. Крім комп'ютерного зору, існують також електромагнітні системи, які визначають місцеперебування об'єкта, вимірюючи електромагнітні поля, створювані передавачем, наприклад, радіочастотним.

4. ЩО ТАКЕ НЕЙРОННІ МЕРЕЖІ ТА ГЛИБОКЕ НАВЧАННЯ

Нейронні мережі дуже добре показали себе у розпізнаванні жестів і дуже багато методів працюють на основі нейромережових технологій. Перед тим як детальніше розглянемо ці методи, слід дослідити нейронні мережі загалом.

4.1 ВИЗНАЧЕННЯ НЕЙРОННОЇ МЕРЕЖІ

Штучна нейронна мережа (ШНМ) – математична модель, а також її програмне й апаратне втілення, що створена з метою імітування природних біологічних нейронних мереж для розв'язання задач, що розв'язати за допомогою звичайних алгоритмів дуже важко.

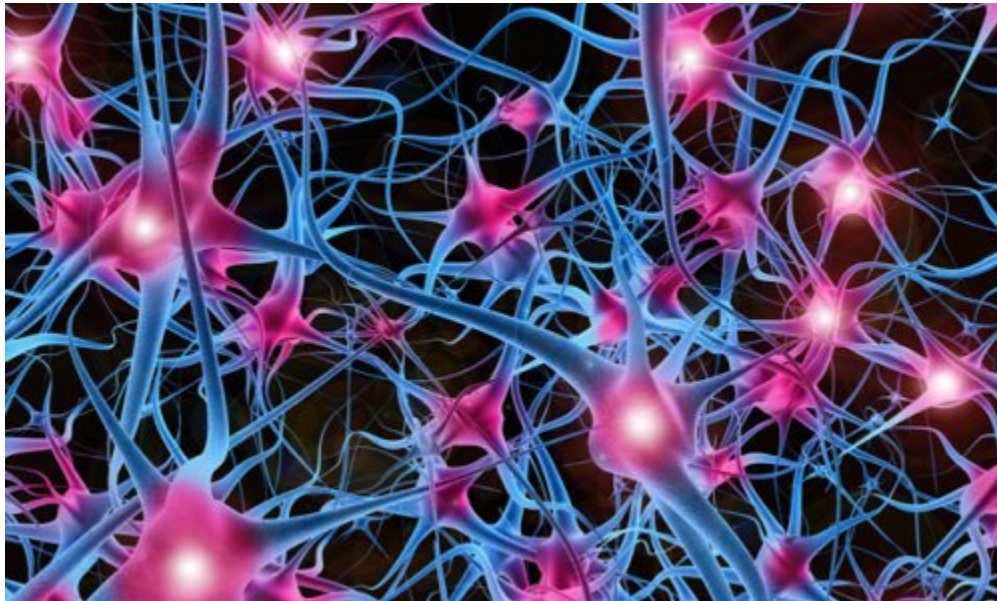


Рисунок 4 – Біологічна нейронна мережа

Основним елементом ШНМ є штучний нейрон (artificial neuron). Штучний нейрон це математична та програмна спрощена модель біологічного нейрона. Як і біологічний аналог, штучний нейрон приймає сигнали з різних входів, обробляє їх певним чином і передає результат обробки іншим нейронам.

Розглянемо принцип роботи біологічних нейронів детальніше.

Біологічні нейрони пов'язані між собою аксонами – відростками нервових клітин, що здатні передавати електричні сигнали. Місце стику нейрона та аксона називається синапсом. Синапси здатні посилювати або послаблювати електричні сигнали що надходять з аксонів. Штучні нейрони також мають схожий механізм синаптичного зв'язку. Штучні синапси мають один параметр – ваговий коефіцієнт. Саме він відповідає за посилення або послаблення сигналу при передачі сигналів між нейронами.

Штучна мережа складається з багатьох шарів нейронів та зв'язків між сусідніми шарами. Таким чином, вхідна інформація перетворюється на сигнали, передається нейронам вхідного шару, що обробляють її певним, залежним від вагових коефіцієнтів, чином та передають нейронам наступного шару. Цей процес повторюється до вихідного шару і на виході отримуємо результат роботи мережі. Процес підбору вагових коефіцієнтів називається навчанням нейронної мережі. Головна мета навчання – підбір таких вагових коефіцієнтів які призводять до отримання бажаного результату.

Структура найпростішої нейронної мережі представлена на малюнку нижче. Зеленим кольором позначені нейрони вхідного шару, блакитним – нейрони прихованого шару, жовтим – нейрон вихідного шару.

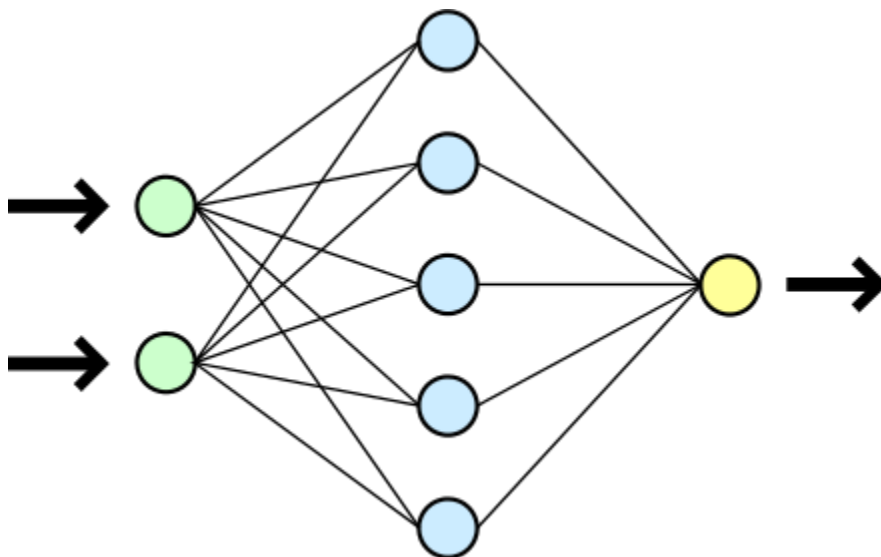


Рисунок 5 – Модель Нейронної Мережі

Перший шар називають вхідним. Нейрони вхідного шару отримують сигнали ззовні і їх обробки передають сигнали на наступний шар. Шари що знаходяться між першим і останнім називають прихованими шарами. Прихований шар отримує сигнали з попереднього шару, обробляє їх та передає далі – на наступний шар. Цей процес виконується доки сигнали не надійдуть до останнього шару, який називають вихідним. Оскільки штучна нейронна мережа імітує біологічну, то в ній також процеси вхідного рівня пов'язані з процесами прихованого рівня, а вже ті пов'язані із декількома процесами вихідного рівня. Описана ШНМ вважається однією із простіших, однак вона здатна на навчання і може знаходити не тільки прості взаємозв'язки в даних, а й взаємозв'язки між взаємозв'язками. Такі ШНМ можуть мати багато прихованих шарів. Кожен наступний шар може виділяти взаємозв'язки в попередньому і тому мережа здатна на глибоке навчання.

4.2 НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

Для навчання нейронних мереж використовується процес, який називається “Методом зворотного поширення помилки або Backpropagation”[2]. Згадаємо, що кожен нейрон нейронної мережі приймає вхідні значення, помножені на вагу, щоб зобразити силу цього зв'язку. Зворотне поширення виявляє правильні вагові коефіцієнти, які слід застосувати до вузлів нейронної мережі, порівнюючи поточні виходи мережі з бажаними або правильними виходами. Різниця між бажаним і поточним виходом обчислюється за допомогою функції втрат, котру ще називають Loss-функцією. Іншими словами, функція Loss говорить нам, наскільки точно наша нейронна мережа робить прогнози для даного входу.

Функція втрат, загалом, це різниця між правильними вихідними значеннями та обчисленими вихідними значеннями. Насправді існують три загальні функції втрат для нейронних мереж: середня квадратична помилка, помилка перехресної ентропії та двійкова помилка крос-ентропії.

Потім алгоритм коригує кожен вагу, щоб мінімізувати різницю між обчисленим і правильним значенням. Термін «зворотне поширення» походить від того факту, що алгоритм повертається назад і коригує ваги та зміщення після обчислення відповіді. Чим менше Loss-функція для мережі, тим точнішою вона стає. Таким чином, процес навчання можна кількісно оцінити як мінімізацію результату функції втрат. Кожен цикл прямого поширення та корекції зворотного поширення для зменшення втрат називається епохою. Простіше кажучи, зворотне розповсюдження полягає в пошуку найкращих вхідних ваг і зміщень, щоб отримати більш точний вихід або «мінімізувати втрати». Цей процес потребує дуже багато обчислювальних потужностей. Фактично, обчислювальної потужності було недостатньо до відносно недавнього часу, щоб зробити цей процес практичним для широкого використання.

4.3 ТИПИ НЕЙРОМЕРЕЖ І ЇХ КЛАСИФІКАЦІЯ

Зараз у Вікіпедії перелічено 26 типів штучних нейронних мереж. Деякі мають такі назви як: хаотична, сіамська, осциляторна, адаптивного резонансу тощо. Інші називанні на честь їх винахідників. З метою систематизації вже наявних і майбутніх нейромереж, робляться спроби їх класифікації.

Класифікація за типом вхідних даних:

- Виконавчі нейронні мережі: обробляють двійкові числа на вході
- Аналогові нейронні мережі: обробляють дійсні числа на вході,

- Образні нейронні мережі: обробляють знаки, ієрогліфи, символи на вході.

Класифікація за характером навчання:

- Навчання без вчителя: вихідний простір рішень формується тільки на основі вхідних впливів
- Навчання з учителем: вихідний простір рішень нейронної мережі відомо
- Навчання з підкріпленням: система призначення штрафів і заохочень ШНМ, що визначаються в результаті взаємодії мережі з середовищем.

Класифікація за характером налаштування синапсів:

- ШНМ з фіксованими зв'язками: вагові коефіцієнти одразу підібрані, виходячи з умов задачі
- Мережі з динамічними зв'язками: синаптичні зв'язки змінюються у процесі навчання.

Класифікація за часом передачі сигналу:

- Синхронні мережі: мережі в яких час передачі для кожного синаптичного зв'язку дорівнює нулю, або константі
- Асинхронні мережі: мережі в яких час передачі для кожного зв'язку між елементами свій, але теж постійний.

Класифікація за характером зв'язків:

- Мережі прямого поширення: мережі в яких всі зв'язки направлені строго від вхідних нейронів до вихідних
- Рекурентні мережі: мережі в яких сигнал з вихідних нейронів або нейронів прихованого шару частково передається назад на входи нейронів вхідного шару

Серед різних типів ШНМ слід виділити декілька.

4.3.1 ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ

Згорткові нейронні мережі (Convolutional neural networks або CNN[3]) відрізняються від інших тим, що дуже добре працюють з обробкою зображень, мовлення або аудіо. Вони складаються з трьох типів шарів: згорткові шари, агрегувальні шари та повноз'єднані шари.

Згортковий шар – це перший, вхідний, шар CNN після якого можуть йти або ж додаткові згорткові шари або ж агрегувальні шари. Повноз'єднаний шар завжди є останнім вихідним шаром ШНМ.

З кожним шаром CNN збільшується у своїй складності, ідентифікуючи більші частини зображення. Більш ранні шари зосереджені на простих елементах, таких як кольори та краї. Коли дані зображення просуваються через шари CNN, він починає розпізнавати більші елементи або форми об'єкта, поки нарешті не ідентифікує передбачуваний об'єкт.

4.3.2 РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ

Рекурентна нейронна мережа (RNN)[4] — це тип штучної нейронної мережі, яка використовує послідовні дані або дані часових рядів. Ці алгоритми глибокого навчання зазвичай використовуються для порядкових або тимчасових проблем, таких як мовний переклад, обробка природної мови (nlp), розпізнавання мовлення. Подібно до прямих і згорткових нейронних мереж (CNN), рекурентні нейронні мережі використовують навчальні дані для навчання. Вони відрізняються своєю «пам'яттю», оскільки вони беруть інформацію з попередніх входів, щоб впливати на поточний вхід і вихід. У

той час як традиційні глибокі нейронні мережі припускають, що входи та виходи незалежні один від одного, вихід рекурентних нейронних мереж залежить від попередніх елементів у послідовності.

4.4 ЗАВДАННЯ, ЯКІ ВИРІШУЮТЬСЯ НЕЙРОННИМИ МЕРЕЖАМИ

Існує ціла низка задач, для яких дуже важко формалізувати та створити алгоритм який зміг би ці задачі розв'язати. Проте багато з цих задач людський мозок може вирішувати. Саме для обчислення таких задач з якими біологічні нейронні мережі справляються найкраще використовуються ШНМ. Ось перелік найпоширеніших завдань:

- **Розпізнавання образів.** За образи можна вважати: текстові символи, зображення, зразки звуків і т. д. Зараз ця область застосування нейронних мереж є найбільш актуальною. Зокрема, саме ШНМ, що розпізнають образи використовує Google для пошуку зображень, або сучасні смартфони визначають де на зображенні з камери знаходяться обличчя.
- **Класифікація** - розподіл даних по параметрах. Наприклад Маємо набір даних про людей, такі як: вік, платоспроможність кредитна історія. Можна навчити ШНМ класифікувати людей на тих кому можна давати кредит, а кому ні.
- **Прийняття рішень і управління.** Отримуючи різні дані про стан керованої системи ШНМ може приймати ті чи інші рішення.
- **Кластеризація.** Під час навчання ШНМ розбиває отриманні данні на різні класи, при цьому кількість класів заздалегідь невідома. Під час роботи мережа вже може визначати до якого класу відносити той чи інший сигнал, або ж це може бути зовсім новий клас.

- **Прогнозування.** Попередньо вже згадувалось про здатність ШНМ знаходити патерни, залежності та взаємозв'язки між даними. Саме завдяки цьому мережа може прогнозувати майбутнє значення якоїсь послідовності, виходячи з попередніх даних

5. ЩО ТАКЕ КОМП'ЮТЕРНИЙ ЗІР

Сучасні методи розпізнавання жестів ґрунтуються здебільшого на технології комп'ютерного зору, яка своєю чергою ґрунтується на технології штучних нейронних мереж.

Комп'ютерний зір — це область досліджень штучного інтелекту, головною метою якої є надати змогу комп'ютерним системам отримувати значущу інформацію з різних візуальних даних, таких як: цифрові зображення, відео та інші. Вже виходячи з отриманої інформації система може виконувати певні дії або давати рекомендації. Говорячи неформально, комп'ютерний зір дозволяє комп'ютерам бачити, спостерігати і розуміти.

Принцип роботи заснований на імітуванні природного зору людини. Тобто отримується необроблені дані від периферійних джерел(камери, наприклад, або ж вже готове цифрове зображення) та ці дані обробляються нейронними мережами, отримуючи потрібну інформацію.

Людська зорова система тренується протягом усього життя та навчена розпізнавати дуже багато образів, розрізняти освітлення, віддаленість та має працювати з дуже широкою предметною областю. Натомість системи комп'ютерного зору сьогодні працюють в основному з якимось конкретним завданням, оскільки навчити багатозадачну штучну нейромережу, схожу за функціоналом з людською, з поточним рівнем технологій дуже важко. Але зі

своїм вузькоспеціалізованим завданням комп'ютерна мережа справляється набагато швидше та точніше ніж людина.

Комп'ютерний зір використовується в різних галузях — від енергетики та комунальних послуг до виробництва та автомобільної промисловості – і ринок продовжує зростати.

5.1 ЯК ПРАЦЮЄ КОМП'ЮТЕРНИЙ ЗІР

Запровадження технологій комп'ютерного зору потребує дуже багато даних. Комп'ютер має аналізувати дані доти, доки не навчиться розрізняти відмінності та впізнавати зображення. Наприклад, для того, щоб навчити комп'ютер розпізнавати кораблі, потрібно надати йому велику кількість зображень кораблів та об'єктів з ними пов'язаних.

Для всього цього використовуються дві основні технології: глибоке навчання нейромережі та згорткові нейронні мережі

Машинне навчання використовується, щоб навчити комп'ютер про контекст візуальної інформації. Комп'ютер сам навчає себе розрізняти зображення замість того, щоб хтось запрограмував його.

Згорткові нейронні мережі використовуються для того, щоб модель машинного навчання вміла «бачити» об'єкти, розбиваючи зображення на пікселі, яким надано теги або мітки. Модель використовує мітки для виконання згорток (математична операція над двома функціями для створення третьої функції) і робить прогнози щодо того, що воно «бачить». Нейронна мережа виконує згортки і перевіряє точність своїх прогнозів у серії ітерацій, поки передбачення не почнуть збуватися. Потім модель розпізнає або бачить зображення так, як люди.

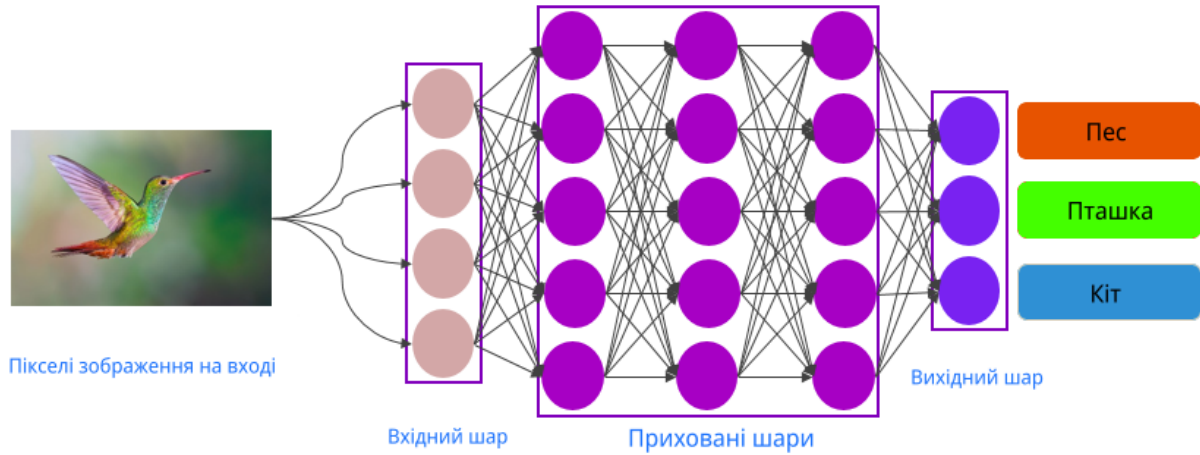


Рисунок 6 – Модель Згорткової Нейронної Мережі для розпізнавання об'єктів

Подібно до того, як людина розрізняє об'єкти, згорткова нейронна мережа спочатку визначає краї та загальну форму об'єкта, потім визначає додаткову інформацію протягом кожної ітерації свого процесу передбачення. Згорткові нейронні мережі використовуються для обробки статичних поодиноких зображень. Технологія рекурентних нейронних мереж використовується схожим чином для того, щоб працювати із відео. Рекурентні нейронні мережі дозволяють комп'ютеру знаходити зв'язки між послідовністю зображень.

5.2 ІСТОРІЯ РОЗВИТКУ ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ

Спроби вчених та інженерів розробити способи, щоб машини могли бачити й розуміти візуальні дані почалися близько 60 тому. Перший експеримент відбувся в 1959 році, коли група нейрофізіологів показали кішці різні зображення та спостерігали реакції в її мозку. Результати показали, що спочатку мозок реагує на жорсткі краї або лінії. Формально кажучи, це означало, що процес обробки зображень починається з простих форм, таких як прямі лінії та краї.

Приблизно в той же час була розроблена перша технологія комп'ютерного сканування зображень, яка дозволила комп'ютерам оцифрувати та отримувати зображення. В 1963 році комп'ютери змогли перетворити двовимірні зображення в тривимірні форми. У 1960-х роках Штучний інтелект став академічна область дослідження, і це також поклало початок пошукам технологій для розв'язання проблеми людського зору.

У 1974 з'явилася технологія оптичного розпізнавання символів (optical character recognition або OCR), що могла розрізняти текст написаний будь-яким шрифтом та на різних поверхнях.

У 1982 році, нейробіолог Девід Марр встановив, що природа зорового процесу є ієрархічною і ґрунтуючись на цьому він створив машинний алгоритм для розпізнавання країв, кутів, кривих та інших базових фігур. В той самий час Куніхіко Фукусіма створив систему, що могла розпізнавати патерни. Цю мережу назвали Неокогнітрон та працювала вона на основі згорткових шарів.

У нульових почали досліджувати методи розпізнавання об'єктів. У 2001 році з'явилася перша програма для виявлення обличчя людини. Почалася стандартизація структури датасетов для навчання. У 2010 у

вільному доступі з'явився набір різних датасетів ImageNet, що містив мільйони маркованих зображень для тисячі класів об'єктів. Це стало основою для багатьох моделей які використовуються і досі.

5.3 ДЕ ЗАСТОСОВУЄТЬСЯ ТЕХНОЛОГІЯ КОМП'ЮТЕРНОГО ЗОРУ

У сфері комп'ютерного зору проводиться багато теоретичних досліджень, але це не лише гола теорія. Реальні застосунки цієї технології демонструють, наскільки важливий комп'ютерний зір для бізнесу, розваг, транспорту, охорони здоров'я та повсякденного життя. Ключовою рушійною силою розвитку цих додатків є потік візуальної інформації, що надходить від смартфонів, систем безпеки, камер дорожнього руху та інших візуальних приладів. Ці дані можуть відігравати важливу роль у діяльності різних галузей. Ця інформація створює тестовий майданчик для навчання програм комп'ютерного зору.

6. МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ ТА НЕЙРОМЕРЕЖ

Методи розпізнавання жестів на основі комп'ютерного зору та нейромереж отримують зображення, що містять руку людини, за допомогою камер, а потім використовує технології комп'ютерного зору і машинне

навчання, щоб оцінку та розпізнавання жесту. На цей час процес розпізнавання жестів на основі комп'ютерної графіки поділяються на три етапи: збір зображень, виявлення та сегментація рук, розпізнавання та класифікація жестів.



Рисунок 5 – Етапи процесу розпізнавання жестів

6.1 ЗБІР ЗОБРАЖЕНЬ

Збір зображень ділиться за методом отримання зображень на RGB зображення та зображення глибини. Зображення RGB можна отримати за допомогою звичайної камери, а камери глибини, такі як Kinect і Leap Motion, мають можливість одночасно отримувати зображення глибини та зображення RGB. Використання зображень глибини дає змогу записувати частину просторової інформації, що полегшує розпізнавання та класифікацію жестів. Жести поділяються на статичні та динамічні, які визначають, обробляти одну фотографію чи відео. У процесі збору зображень виникають такі проблеми, як оклюзія, різна інтенсивність і напрям світла, що висуває підвищені вимоги до надійності алгоритму. З висуванням практичності розпізнавання жестів все більше і більше алгоритмів присвячені досягненню інваріантності освітлення та розв'язання проблем оклюзії.

6.2 СЕГМЕНТАЦІЯ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Сегментація жестів на основі згорткових нейронних мереж включає оптимізацію за допомогою згорткових нейронних мереж на основі FCN (Full Convolutional Neural Networks) або SegNet.

Мережа з повною згорткою (FCN) — це нейронна мережа, яка виконує лише операції згортки (і підвибірки чи підвищення дискретизації).

Еквівалентно, FCN – це CNN без повністю підключених шарів.

SegNet — це модель семантичної сегментації. Ця базова архітектура сегментації, яку можна навчати, складається з мережі кодувальника, відповідної мережі декодера, за якою слідує шар класифікації по пікселях.

FCN і SegNet замінюють останній шар CNN шаром деконволюції, і зображення відновлюється до його початкового розміру шляхом збільшеної вибірки, і кожен піксель прогнозується. У порівнянні з CNN, FCN і SegNet можуть приймати вхідні зображення будь-якого розміру, більше не вимагають, щоб усі зображення мали однаковий розмір, і уникають проблеми повторного зберігання та обчислень згортки. Однак у FCN є й очевидні недоліки. Коли коефіцієнт підвищення дискретизації високий, зображення не дуже чітке, і чутливість до деталей потрібно покращити; зв'язок між різними пікселями використовується не повністю

6.3 РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЯ ЖЕСТІВ

Розпізнавання жестів поділяється на динамічне і статичне. Статичні жести – це жести однієї картинки, а динамічні – це зміни рухів жестів у часі, тобто кілька послідовних статичних жестів. Зображення розпізнавання жестів

поділяється на карту глибини, карту RGB та карту RGB-D. Карта глибини може безпосередньо відображати відстань між камерою та об'єктом.

Зображення карти глибини подібне до сірого зображення. Різниця в тому, що карта глибини показує відстань між об'єктом і камерою з кожним пікселем. Зображення RGB-D містить триканальні RGB зображення та зображення глибини. Хоча два зображення виглядають по-різному, між пікселями існує відповідність один до одного. В останні роки розпізнавання жестів здебільшого базується на штучних нейронних мережах (ANN-CNN, RNN, GAN), алгоритмах НММ і DTW. Алгоритми DTW і НММ спочатку використовувалися для розпізнавання мовлення.

Алгоритм DTW заснований на ідеї DP (динамічного програмування) для розв'язання проблеми різної довжини вимови. Алгоритм DTW не вимагає великої кількості даних для навчання, як НММ та згортка нейронної мережі. Алгоритм простий і швидкий, ідея полягає в тому, щоб знайти оптимальний шлях і знайти відповідну послідовність з найменшими накладними витратами відповідно до оптимального шляху. НММ шукає приховані послідовності з спостережуваних послідовностей і знаходить значення, яке має бути виражене за допомогою жестів. Спочатку для класифікації зображень використовувалася згорткова нейронна мережа. У розпізнаванні жестів її можна використовувати для виділення ознак і зменшення розмірності або безпосередньо класифікації. Вона більш гнучка, але вимагає великої кількості маркованих навчальних даних. Це займає час у процесі навчання та потребує прискорення GPU.

6.3.1 РОЗПІЗНАВАННЯ ЖЕСТУ НА ОСНОВІ НЕЙРОМЕРЕЖ

Завдання розпізнавання жестів можна розглядати як типову проблему розпізнавання певних об'єктів на зображенні. В цьому дуже добре себе показали ШНМ, а саме згорткові нейронні мережі. мережами. Нейронні

мережі в цілому демонструють високу точність виявлення об'єктів, при умовах різних шумів, спотворень, різного масштабу та різних умов освітлення. CNN гарна технологія для розпізнавання статичних жестів. Статичні жести – це ті, які вимагають обробки лише одного зображення на вході класифікатора, перевагою цього підходу є менші обчислювальні витрати. Динамічні жести вимагають обробки послідовностей зображень і більш складних підходів до розпізнавання жестів. Якщо ж потрібно обробити не одне статичне зображення, а, наприклад відео, то для цього використовують рекурентні нейронні мережі. RNN здатні знаходити взаємозв'язки між послідовними кадрами відео.

Для підвищення точності роботи моделей на основі CNN та RNN, мережі можуть притерпіти незначні зміни в архітектурі. В відкритому доступі є дуже багато датасетів з великою кількістю маркованих зображень різних жестів.

Алгоритм на основі нейронної мережі є гнучким і може бути налаштований у багатьох аспектах. Його можна використовувати в поєднанні з різними алгоритмами. Однак загальна мережева структура надзвичайно складна.

7. СУЧАСНИЙ СТАН РОЗВИТКУ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ НА ОСНОВІ КОМП'ЮТЕРНОГО ЗОРУ ТА НЕЙРОМЕРЕЖ

В останні роки методи на основі нейронних мереж стали масовими. [10] Було запропоновано метод генерування 3D-моделей людської руки з RGB-зображень у режимі реального часу за допомогою вдосконаленої GAN (Generative Adversarial Nets) [11]. Жести генеруються GAN, а 2D-координати та 3D-координати отримуються за допомогою багатоцільового навчання. У 2017 році група дослідників [12] запропонувала метод тривимірної оцінки за допомогою одного RGB-зображення руки. Вони використовували PoseNet [13], який калібрує положення камери відповідно до сцени, положення руки калібрується відповідно до положення камери, а положення суглоба руки інтерпретується в поєднанні з попередніми знаннями, щоб точно отримати кістковий вузол за допомогою одне зображення RGB. Недоліком є те, що вся мережа займає багато часу, і все ще залишається багато місця для оптимізації, тому можна сподіватися досягти розпізнавання в реальному часі на ЦП.

У 2015 році було запропоновано метод на основі згорткової нейронної мережі [14] для попереднього позиціювання та повторної оптимізації. Він може точно визначити місцеперебування руки на одному зображенні глибини після тренування з кількома позначеними зображеннями глибини. Метод розташування вузла називається HandDeer. На основі HandDeer, запропонований DeerPrior++ у 2017 році.

Результати експериментів демонструють, що запропоновані алгоритми можуть точно розпізнавати жести в режимі реального часу і демонструють хорошу адаптацію до різних типів рук. Загальний рівень розпізнавання

становить понад 94%. Рівень розпізнавання все ще перевищує 93%, коли зображення жестів руки обертаються, перекладаються або масштабуються.

8. OPENSOURCE РІШЕННЯ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ

Дуже багато досліджень та готового коду по в галузі розпізнавання жестів та штучного інтелекту загалом є в відкритому доступі в інтернеті. Це дуже хороша тенденція, оскільки відкритість матеріалів та доступність інструментів сприяє розвитку галузі загалом. Також це дозволяє не дослідникам, а ентузіастам спробувати технологію власноруч. Наведемо список інструментів корисних для роботи із ШНМ, глибоким навчанням та комп'ютерним зором, а тому і корисних для роботи з розпізнаванням жестів.

8.1 TensorFlow

TensorFlow[15] — це бібліотека з відкритим сирцевим кодом, розроблена Google в основному для програм глибокого навчання. Бібліотека також підтримує традиційне машинне навчання. TensorFlow спочатку був розроблений для великих чисельних обчислень без урахування глибокого навчання. Однак бібліотека виявилася дуже корисною і для розробки глибокого навчання, і тому Google відкрив його з відкритим кодом.

TensorFlow приймає дані у вигляді багатовимірних масивів вищих розмірів, які називаються тензорами. Багатовимірні масиви дуже зручні при обробці великих обсягів даних.

TensorFlow працює на основі графіків потоків даних, які мають вузли та ребра. Оскільки механізм виконання у вигляді графіків, набагато легше

виконувати розподілений код TensorFlow на кластері комп'ютерів під час використання графічних процесорів.

Бібліотека TensorFlow надає API як для мови Python, так і C++ До створення спеціальних бібліотек, програмування механізмів для машинного навчання було набагато складнішим. Ця бібліотека надає API високого рівня, усуваючи необхідність складного кодування низькорівневих процесів нейронної мережі, налаштування нейрона або програмування нейрона. Бібліотека виконує всі ці завдання. TensorFlow також має інтеграцію з Java та R.

TensorFlow підтримує обчислювальні пристрої як CPU, так і GPU Програми глибокого навчання дуже складні, а процес навчання вимагає багато обчислень. Це займає багато часу через великий розмір даних і включає багато ітераційних процесів, математичні обчислення, множення матриць тощо. Виконання всіх цих процесів на звичайному CPU займатиме дуже багато часу.

Графічні процесори (GPU) популярні в контексті ігор, де потрібно, щоб екран і зображення мали високу роздільну здатність. Однак вони також використовуються для розробки програм глибокого навчання.

Однією з основних переваг TensorFlow є те, що він підтримує обчислення на графічних процесорах, а також центральних процесорах.

8.2 OpenCV

OpenCV (Open Source Computer Vision Library) – бібліотека комп'ютерного зору та машинного навчання з відкритим сирцевим кодом. OpenCV створено для забезпечення загальної інфраструктури для додатків комп'ютерного зору в комерційних продуктах. Бібліотека поширюється за ліцензією BSD, тому OpenCV дозволяє компаніям легко використовувати та змінювати код.

Бібліотека має понад 2500 оптимізованих алгоритмів, які включають повний набір як класичних, так і найсучасніших алгоритмів комп'ютерного зору та машинного навчання. Ці алгоритми можна використовувати для виявлення та розпізнавання облич, ідентифікації об'єктів, класифікації дій людини на відео, відстеження рухів камери, відстеження рухомих об'єктів, вилучення 3D-моделей об'єктів, створення тривимірних хмар точок зі стереокамер, з'єднання зображень разом для отримання високої роздільної здатності. зображення всієї сцени, знаходити схожі зображення з бази даних зображень, видаляти червоні очі із зображень, зроблених за допомогою спалаху, слідкувати за рухами очей, розпізнавати пейзаж і встановлювати маркери для накладання на них доповненої реальності тощо.

8.3 MediaPipe

MediaPipe — це фреймворк для створення конвеєрів машинного навчання для обробки даних часових рядів, таких як відео, аудіо тощо. Ця кросплатформна платформа працює на ПК/серверах, Android, iOS та вбудованих пристроях, таких як Raspberry Pi та Jetson Nano.

Фреймворк написаний на C++, Java та Obj-C, який складається з наступних API.

- API калькулятора (C++).
- API побудови графіка (Protobuf).
- API виконання графіків (C++, Java, Obj-C).



Рисунок 6 - Приклад роботи бібліотеки MediaPipe

9. ПРИКЛАД ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

Для демонстрації ефективності роботи сучасних методів розпізнавання жестів створимо програму що визначає та класифікує статичні жести руки з камери у режимі реального часу. Все це можна зробити використовуючи відкриті бібліотеки та фреймворки.

Програма буде створена на основі мови Python[18], використовуючи такі інструменти як: TensorFlow, OpenCV та MediaPipe.

9.1 НЕОБХІДНІ УМОВИ

Для розробки та роботи програми, на комп'ютері має бути встановлений інтерпретатор мови Python[18]. Далі слід встановити всі необхідні інструменти для роботи:

- TensorFlow:

Це можна зробити запустивши в консолі команду
“pip install tensorflow”.

- OpenCV:

Це можна зробити запустивши в консолі команду
“pip install opencv-python”.

- MediaPipe:

Це можна зробити запустивши в консолі команду
“pip install mediapipe”.

Також нам знадобиться наперед навчена модель нейронної мережі для класифікації жестів.

9.2 ПРОГРАМНИЙ КОД

Для початку нам слід імпортувати необхідні бібліотеки

```
import cv2
import numpy as np
import mediapipe as mp
import tensorflow as tf
from tensorflow.keras.models import load_model
```

Наступним кроком буде ініціалізація моделі ШНМ.

```
# initialize mediapipe model
mpHands = mp.solutions.hands
hands = mpHands.Hands(max_num_hands=1,
min_detection_confidence=0.7)
mpDraw = mp.solutions.drawing_utils
```

- Модуль `mp.solutions.hands` містить алгоритм розпізнавання рук. Тому ми створюємо об'єкт і зберігаємо його в `mpHands`.
- За допомогою методу `mpHands.Hands` ми налаштували модель. Першим аргументом є `max_num_hands`, тобто максимальна кількість рук буде виявлена моделлю в одному кадрі. `MediaPipe` може виявляти кілька рук в одному кадрі, але ми будемо виявляти лише одну руку за раз у цьому проєкті.
- `mp.solutions.drawing_utils` намалює виявлені ключові точки для нас, щоб нам не довелося малювати їх вручну.

Далі ініціалізуємо модель TensorFlow:

```
# Load the gesture recognizer model
model = load_model('mp_hand_gesture')
```

Використовуючи функцію `load_model`, ми завантажуюмо попередньо навчену модель TensorFlow.

Потім потрібон зазначити назви кожного класу жестів.

```
# Load class names
classNames = ["okay",
              "peace",
              "thumbs up",
              "thumbs down",
              "call me",
              "stop",
              "rock",
              "live long",
              "fist",
              "smile"]
```

Далі потрібно реалізувати читання відеопотоку з камери. Для цього будемо використовувати функціонал бібліотеки OpenCV.

```
while True:
    # Read each frame from the webcam
    _, frame = cap.read()

    x, y, c = frame.shape

    # Flip the frame vertically
    frame = cv2.flip(frame, 1)
```

```

cv2.imshow("Output", frame)

if cv2.waitKey(1) == ord('q'):
    break

# release the webcam and destroy all active windows
cap.release()

cv2.destroyAllWindows()

```

- Ми створюємо об'єкт VideoCapture і передаємо аргумент «0». Це ідентифікатор камери системи. У цьому випадку ми маємо 1 веб-камеру, підключену до системи.
- Функція cap.read() зчитує кожен кадр з веб-камери.
- Функція cv2.flip() перевертає кадр.
- cv2.imshow() показує кадр у новому вікні openCV.
- Функція cv2.waitKey() тримає вікно відкритим, поки не буде натиснута клавіша «q».

Наступним кроком буде визначення та сегментація руки на фреймі. Це зроблено за допомогою функціоналу бібліотеки MediaPipe.

```

framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Get hand landmark prediction
result = hands.process(framergb)

# print(result)

className = ''

# post process the result
if result.multi_hand_landmarks:

```

```

landmarks = []
for handslms in result.multi_hand_landmarks:
    for lm in handslms.landmark:
        # print(id, lm)
        lmx = int(lm.x * x)
        lmy = int(lm.y * y)

        landmarks.append([lmx, lmy])

# Drawing landmarks on frames
mpDraw.draw_landmarks(frame, handslms,
mpHands.HAND_CONNECTIONS)

```

- MediaPipe працює з зображеннями RGB, але OpenCV читає зображення у форматі BGR. Отже, за допомогою функції `cv2.cvtColor()` ми конвертуємо кадр у формат RGB.
- Функція процесу приймає кадр RGB і повертає клас результату.
- Потім ми перевіряємо, чи вдалося розпізнати руку чи ні, використовуючи метод `result.multi_hand_landmarks`.
- Після цього ми перебираємо кожну знайдену руку (в нашому випадку ми задали що ми перевіряємо тільки 1) та зберігаємо координати в списку, який називається орієнтирами або оперними точками.
- Тут висота зображення (y) і ширина зображення (x) помножуються на результат, оскільки модель повертає нормований результат. Це означає, що кожне значення в результаті знаходиться від 0 до 1.
- І, нарешті, за допомогою функції `mpDraw.draw_landmarks()` ми малюємо всі орієнтири в кадрі.

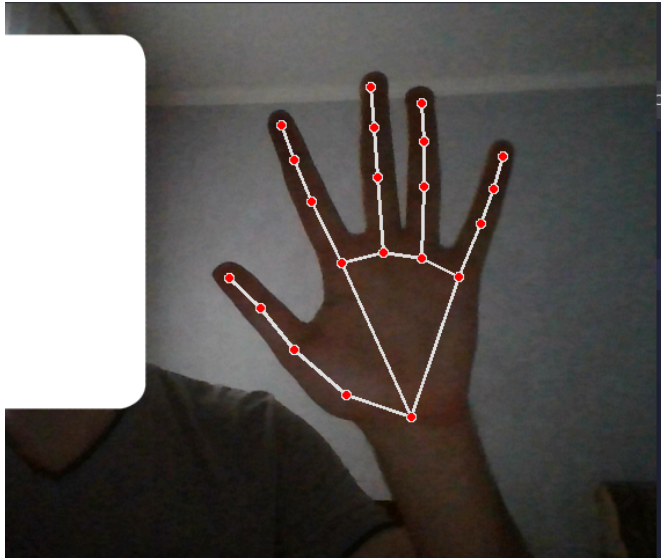


Рисунок 7 – Демонстрація роботи модуля розпізнавання та сегментації рук.

Отримані орієнтири кидаємо на вхід нейромережі для класифікації жестів.

```
# Predict gesture
prediction = model.predict([landmarks])
# print(prediction)
classID = np.argmax(prediction)
className = "none"
if prediction[0][classID] > 0.7:
    className = classNames[classID]

# show the prediction on the frame
cv2.putText(frame, className, (10, 50),
cv2.FONT_HERSHEY_SIMPLEX,
            1, (0,0,255), 2, cv2.LINE_AA)
```

- Функція `model.predict()` приймає список опорних точок і повертає масив, що містить 10 класів передбачення для кожного орієнтира.
- Вихід виглядає так -

```
[[8.8656884e-01 3.9141513e-11 2.0740410e-06 1.3242191e-13 1.1913717e-11
1.3757647e-06 8.3538955e-03 6.2032868e-09 1.0507387e-01 9.7458908e-10]]
```

- `np.argmax()` повертає індекс максимального значення в списку.
- Після отримання індексу ми можемо просто взяти ім'я класу зі списку `classNames` якщо значення більше ніж 0.7. Це зроблено для того що б жести які скоріш за все модель не знає, помилково не відображалися як якийсь відомий клас.
- Потім за допомогою функції `cv2.putText` ми показуємо виявлений жест у кадрі.

9.3 РЕЗУЛЬТАТ

Отже, підсумує загальний алгоритм процедури розпізнавання жестів. Перш за все програма отримає запис із камери та відзеркалює кадр. Далі, використовуючи бібліотеку `MediaPipe`, що працює на принципі згорткових нейронних мереж, та містить наперед навчену модель, програма знаходить опорні точки на кадрі, що позначають певний сектор руки чи окремого пальця людини. Відповідні точки можна поєднати ребрами, і отриманий граф можна вивести поверх кадру з камери. Отримаємо щось на кшталт скелета руки.

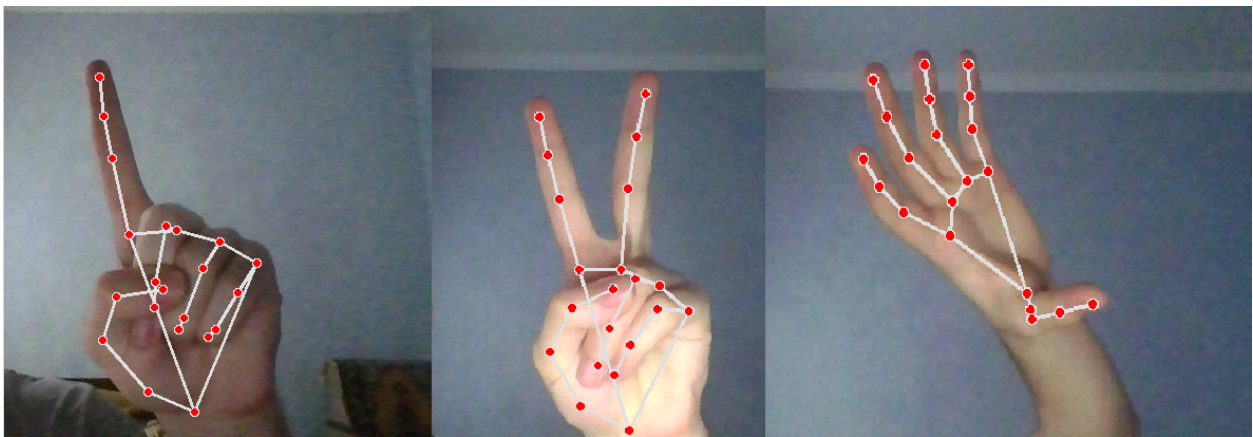


Рисунок 9 – Демонстрація опорних точок руки.

Наступним кроком, перетворюємо координати кожної опорної точки в вектор, та передаємо його на вхід наперед навчаної моделі нейронної мережі, яка опираючись на ці точки може робити передбачення, що до жесту який показує рука.

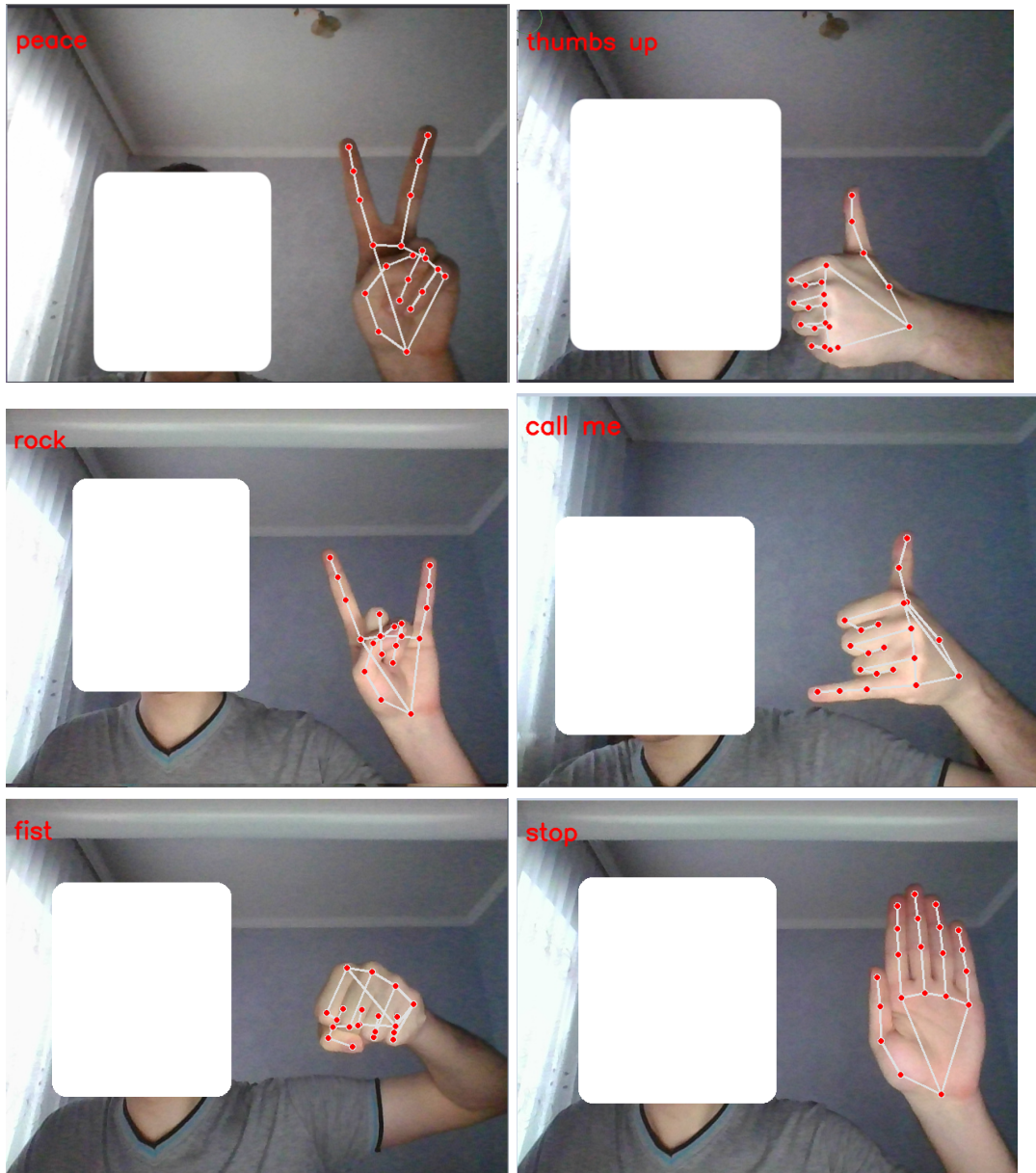


Рисунок 9 – Демонстрація роботи програми.

Бачимо, що програма працює. Кажучи про точність роботи, треба рахувати точність сегментації руки та розпізнавання жестів окремо. Модуль сегментації рук бібліотеки MediPipe дуже точно відстежує руку та не помиляється із розпізнаванням різних частин руки(на відео це помітніше ніж тільки рисунках). При умові що камера бачить руку цілком, можна сказати що точність десь 98%. Модель що розпізнає жести використовуючи опорні точки працює теж досить точно. Однак окремі жести можуть викликати труднощі(“*rock*”, “*stop*”). В найгіршому випадку – точність 60%, в середньому – 85%. Також слід зазначити що результат роботи програми стійкий до шумів, освітлення, положення руки та інших нюансів.

10.ПРОБЛЕМИ

Після більш ніж 20-річного досвіду технологія розпізнавання жестів на основі комп'ютерного зору досягла значного прогресу. Було досліджено та розроблено дуже багато методів. Однак всі ці методи стикаються з багатьма обмеженнями, і все ще існують певні вузькі місця в розвитку.

1. Серйозно вплив на результат залежно від фону. Подібно до традиційної класифікації зображень, хороша сегментація жестами все ще потрібна на складних фонах. Коли виконується розпізнавання жестів, чи можна точно відокремити жест від фону, є ключем до підвищення точності розпізнавання;
2. Проблеми з оклюзією. При динамічному розпізнаванні жестів жести можуть бути заблоковані деякими об'єктами в навколишньому середовищі, що посилює складність відстеження жестів;
3. Різні жести подібні, і один і той же жест може відрізнитися;
4. Різні кути огляду та інтенсивність світла. Інваріантність обертання та інваріантність освітлення важче виконати в процесі розпізнавання жестів.

У порівнянні з багатьма методами, метод нейронної мережі повільний, точний і залежить від даних. Він вимагає великої кількості мічених даних і потужної обчислювальної швидкості.

ВИСНОВКИ

В рамках даної роботи були досягнуті наступні результати:

- Досліджені основні підходи та методи сегментації рук, та розпізнавання жестів.
- Були проаналізовані переваги та недоліки відомих методів та архітектур нейромереж.
- Наведено сучасний стан дослідження проблеми розпізнавання жестів, відкриті задачі та проблеми.

Після майже 20 років розробки методів розпізнавання жестів було вирішено багато проблем та вузьких місць. Але все ще існують такі проблеми, як погана універсальність, чутливість до змін освітлення та оклюзії, а також низька продуктивність у реальному часі.

Сама технологія має дуже великий потенціал. Вона вже дуже багато використовується у взаємодії людини із комп'ютером і буде використовуватися ще більше із розвитком технологій IoT та автоматизації буденних операцій.

ДЖЕРЕЛА

- 1) https://en.wikipedia.org/wiki/Gesture_recognition [Електронний ресурс]
- 2) [How Do Neural Networks Learn?](#) [Електронний ресурс]
- 3) [Convolutional Neural Networks](#) [Електронний ресурс]
- 4) [Recurrent Neural Networks](#) [Електронний ресурс]
- 5) [Recent methods in vision-based hand gesture recognition](#) [Електронний ресурс]
- 6) [Perspective and Evolution of Gesture Recognition for Sign Language: A Review](#) [Електронний ресурс]
- 7) Xianghan Wang, Jie Jiang, Yingmei Wei, Lai Kang, Yingying Gao(2018)
Research on Gesture Recognition Method Based on Computer Vision
- 8) Munir Oudah, Ali Al-Naji, Javaan Chahl (2020)
- 9) Hand Gesture Recognition Based on Computer Vision: A Review of Techniques
- 10) Mueller F, Bernard F, Sotnychenko O, GANerated Hands for Real-time 3D Hand Tracking from Monocular RGB[J]. 2017.
- 11) Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Nets[C].International Conference on Neural Information Processing Systems.2014,3:2672-2680.
- 12) Zimmermann C, Brox T. Learning to Estimate 3D Hand Pose from Single RGB Images[J]. 2017:4913-4921.
- 13) Kendall A, Grimes M, Cipolla R. Convolutional networks for real-time 6-DOF camera relocalization. CoRR abs/1505.07427 (2015)[J]. 2015.
- 14) Oberweger M, Wohlhart P, Lepetit V. Hands Deep in Deep Learning for Hand Pose Estimation[J]. Computer Science, 2016.
- 15) <https://www.tensorflow.org> [Електронний ресурс]

- 16) <https://opencv.org> [Электронный ресурс]
- 17) <https://google.github.io/mediapipe> [Электронный ресурс]
- 18) <https://www.python.org> [Электронный ресурс]