

Ім'я користувача:
Volodymyr Donchenko

ID перевірки:
1016074145

Дата перевірки:
20.01.2024 19:17:47 EET

Тип перевірки:
Doc vs Internet

Дата звіту:
20.01.2024 19:19:20 EET

ID користувача:
100012947

Назва документа: Виправ_Магістерська_робота_Чебаненко_I_O_KI_111111

Кількість сторінок: 76 Кількість слів: 14684 Кількість символів: 114241 Розмір файлу: 1.76 MB ID файлу: 1015781723

8.08% Схожість

Найбільша схожість: 2.44% з Інтернет-джерелом (<http://dspace.wunu.edu.ua/bitstream/316497/17483/1/OlikhVY.PDF>)

8.08% Джерела з Інтернету

938

Сторінка 78

Пошук збігів з Бібліотекою не проводився

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

27

Міністерство освіти і науки України
Державний заклад
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних
технологій

Кафедра інформаційних технологій та
систем

Чебаненко Ілля Олегович

Розробка системи комп'ютерної допомоги для людей з вадами зору

кваліфікаційна робота
здобувача вищої освіти другого (магістерського)
рівня освітньої програми «Комп'ютерні мережі»
за спеціальністю 123 Комп'ютерна інженерія

Особистий підпис	Ілля ЧЕБАНЕНКО
Науковий керівник	Галина КОЗУБ, кандидат технічних наук, доцент кафедри інформаційних технологій та систем
Завідувач кафедри	Микола СЕМЕНОВ, кандидат педагогічних наук, доцент кафедри інформаційних технологій та систем

Полтава – 2024

АНОТАЦІЯ

Чебаненко І.О.

Тема: «Розробка системи комп'ютерної допомоги для людей з вадами зору».

Спеціальність: 123 „Комп'ютерна інженерія”

Установа: ДЗ ЛНУ імені Т. Шевченка, 2024р.

Магістерська робота містить: 75 с., 16 рис., додат., 39 джерел.

Мета роботи: аналіз існуючих навігаційних систем, пристроїв та методів розпізнавання для людей з вадами зору, проблем класифікації графічних образів, огляд засобів та технологій навігаційного забезпечення, розробка системи комп'ютерної допомоги для людей з вадами зору.

Результати роботи. Розроблено програмний модуль у PyCharm за допомогою мови програмування Python, який реалізує запропоновані класи, призначені для отримання результату. У модулі використані необхідні бібліотеки: pyttsx3, datetime, SpeechRecognition, Wikipedia, webbrowser, os.path, smtplib.

Висновок. Розроблене ПО дає можливість людям з вадами зору отримувати за допомогою голосу необхідну інформація, через веб сайти Google, YouTube, виконувати пошук в Вікіпедії, надсилати електронні листи, відтворювати музику і.т.д. Це допоможе повноцінно існувати й орієнтуватись кожній людині з вадами зору.

Ключові слова. PYTHON, ВАДИ ЗОРУ, ТЕХНОЛОГІЇ НАВІГАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ, НАВИЧКИ ОРІЄНТАЦІЇ ТА РУХЛИВОСТІ (O&M), ГРАФІЧНІ ОБРАЗИ, PYCHARM. IDE.

ANNOTATION

Chebanenko I.O.

Theme: „Development of a computer assistance system for visually impaired people”.

Specialty: 123 "Computer engineering"

Institution: Luhansk Taras Shevchenko National University, 2024p.

Master's thesis contains: 75 p., 16 figures, appendixes, 39 sources.

Purpose of the work - Development of a voice assistant, which is an application program that is able to understand human speech, speak natural language and perform the necessary electronic and technical tasks for the end user.

Results of work. An application program, the voice assistant "Lucy", has been created. It is an electronic assistant for visually impaired people who are deprived of the ability to see from birth or due to illness. Class descriptions and recommendations for their use are offered. Codes in Pycharm, IDE. that implement the proposed classes have been developed. The necessary libraries were imported/installed: pyttsx3, datetime, SpeechRecognition, Wikipedia, webbrowser, os.path, smtplib.

Conclusion. The developed software allows visually impaired people to receive the necessary information by voice, through Google, YouTube, search Wikipedia, send emails, play music, etc. This will help every visually impaired person to live and navigate to the fullest.

Keywords. PYTHON, ASSISTANCE TO VISUALLY IMPAIRED PEOPLE, NAVIGATION TECHNOLOGIES, ORIENTATION AND MOBILITY SKILLS (O&M), GRAPHIC IMAGES, PYCHARM. IDE.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. КЛАСИФІКАЦІЯ СИСТЕМ, ПРИСТРОЇВ ТА МЕТОДІВ РОЗПІЗНАВАННЯ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ.....	11
1.1. Допоміжні засоби для подорожей.....	11
1.2. Сенсорні входи.....	13
1.3. Модальності зворотного зв'язку.....	15
1.4. Висновки до розділу 1.....	16
РОЗДІЛ 2. ОГЛЯД ЗАСОБІВ ТА ТЕХНОЛОГІЙ НАВІГАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЛЮДЕЙ З ПОРУШЕННЯМИ ЗОРУ.....	17
2.1. Системи візуальних образів.....	17
2.1.1. Стереокамера.....	18
2.1.2. Мережа IP-камер.....	18
2.1.3. VS-LAM.....	19
2.1.4. Камери RGB-D.....	20
2.1.5. Майкрософт Кінект.....	21
2.1.6. LiDaR.....	22
2.2. Невізуальні системи даних.....	22
2.2.1. BLE-маяки.....	23
2.2.2. Інтернет речей на основі (IoT).....	23
2.2.3. Ультразвукові датчики.....	24
2.2.4. ІЧ-датчики.....	25
2.3. Картографічні системи.....	25
2.4. Системи із 3D-звуком.....	26
2.5. Рішення на базі смартфонів.....	27
2.6. Висновки до розділу 2.....	30
РОЗДІЛ 3. ПРОБЛЕМИ КЛАСИФІКАЦІЇ ГРАФІЧНИХ ОБРАЗІВ.....	33
3.1. Методи розпізнавання графічних образів.....	33
3.2. Алгоритмічні побудови.....	40

	5
3.3. Модель розпізнавання образів.....	50
3.4. Висновки до розділу 3.....	56
РОЗДІЛ 4. РОЗРОБКА ЗАСТОСУВАННЯ ДЛЯ ДОПОМОГИ ЛЮДЯМ З	
ВАДАМИ ЗОРУ.....	57
4.1. Опис програмної реалізації.....	57
4.2. Висновки до розділу 4.....	65
ВИСНОВКИ.....	66
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	68
ДОДАТКИ.....	73

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ETA	-	електронні засоби пересування;
EOA	-	електронні засоби орієнтації;
PLD	-	пристрої позиціонування;
RFID	-	радіочастотна ідентифікація;
NFC	-	зв'язок на ближній відстані;
LASS	-	система LiDaR Assist Spatial Sensing;
IOT	-	інтернет речей на основі;
IHC	-	штучною нейронної мережі;
ІЧ	-	інфрачервоний датчик;
BLD	-	bluetooth з низьким енергоспоживанням;
ГІС	-	географічної інформаційної системи;
ІТ	-	інформаційні технології;
ПК	-	персональний комп'ютер;
ПЗ	-	програмне забезпечення.

ВСТУП

У 2022 році в усьому світі 1,1 мільярда людей жили з втратою зору:

- 43 мільйони людей сліпі;
- 295 мільйонів осіб мають порушення зору від помірного до важкого ступеня;
- 258 мільйонів людей мають легкі порушення зору;
- 510 мільйонів людей мають проблеми із зором ;

Зір є однією з ключових фізіологічних функцій сенсорної системи, і природа наділяє людину здатністю отримувати 80-90% інформації про навколишній світ через очі. Ця інформація важлива для повноцінного функціонування та орієнтації людини, а також для естетичного сприйняття світу.

В Україні наразі існує приблизно 300 тисяч осіб, які мають обмежені можливості бачення з народження або через хворобу. Згідно з даними Всесвітньої організації охорони здоров'я, щорічно кількість сліпих осіб у світі зростає на 1 мільйон, при цьому кожні 5 секунд втрачає зір одна доросла людина, а кожну хвилину – одна дитина. Річну статистику інвалідів в Україні через проблеми зі зором складає близько 12 тисяч осіб.

До 2017 року своїх голосових помічників представили найбільші світові ІТ-компанії: на англomовному ринку працює Siri від Apple, Assistant від Google, Alexa від Amazon, Meta від Facebook, Cortana від Microsoft. На азіатському ринку – Duer від Baidu та Bixby від Samsung.

Це хмарні персональні помічники, які використовують питання-відповідну систему, програмний клієнт якої входить до складу iOS, iPadOS, watchOS, macOS і tvOS компанії Apple. Дані програми використовують обробку природного мовлення, щоб відповідати на запитання та давати відповіді.

Тож технології, які можуть допомогти людям з обмеженими можливостями ефективніше проявляти себе на робочому місці, поліпшити

якість життя, без сумнівів, необхідні, а розробка системи комп'ютерної допомоги для людей з вадами зору, яку виконано в рамках теми магістерської роботи, актуальна.

Мета роботи - аналіз існуючих навігаційних систем, пристроїв та методів розпізнавання для людей з вадами зору, проблем класифікації графічних образів, огляд засобів та технологій навігаційного забезпечення, розробка системи комп'ютерної допомоги для людей з вадами зору.

Для досягнення мети необхідно вирішити наступні завдання:

- аналіз існуючих сучасних засобів та класифікація систем, пристроїв та методів розпізнавання при подорожах та навігації для слабозорих людей;
- особливості алгоритмічних побудов за класифікацією графічних образів;
- аналіз існуючих підходів до розробки системи розпізнавання графічних образів;
- огляд засобів та технологій навігаційного забезпечення для людей з порушеннями зору;
- розробка застосування «Люсі» для допомоги людям з вадами зору засобами Python.

Об'єкт дослідження – методи розпізнавання мовлення.

Предмет дослідження – технології створення системи комп'ютерної допомоги для людей з вадами зору.

Методи дослідження. Теоретичні методи: аналіз науково-технічної та навчально-методичної літератури з проблем дослідження; розробка, проектування і тестування для полегшення процедури розпізнавання мовлення та графічних образів.

Наукові положення, очікувані наукові результати.

1. Сформовано аналіз підходів до розробки комп'ютерних додатків для допомоги людям з вадами зору, а також виявлено недоліки таких підходів;

2. Створено методику проєктування комп'ютерного додатку на основі використання випробуваних і перевірених досвідом досягнень в областях розробки і функціонування програмних засобів.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій магістерської роботи:

- обґрунтовано коректністю поставлених проблем і прийнятих припущень при описі процесів, що відбуваються при оптимізації,
- достатнім обсягом вибірки і верифікованими на явних результатах даних отриманих шляхом тестування.

Наукова новизна: розроблено систему «Люсі» комп'ютерної допомоги для людей з вадами зору засобами Python.

Практичне значення полягає у можливості застосування отриманих результатів роботи для спілкування з помічником голосом і введенням запитів з клавіатури в режимі реального часу.

Особистий внесок магістра складається у:

- аналізі сучасних засобів та класифікація систем, пристроїв та методів розпізнавання при подорожах та навігації для слабозорих людей;
- виборі методів досліджень і технологій реалізації;
- створенні методики розробки програмного забезпечення;
- створенні системи комп'ютерної допомоги для людей з вадами зору;

Апробація результатів магістерської роботи. Основні положення і результати були представлені на II Міжнародній науково-практичній конференції «SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS», Відень, AUT, 2024р.

Публікації. За результатами досліджень, викладених в дипломній роботі, опубліковано 1 тези, «Система комп'ютерної допомоги для людей з вадами зору» DOI. <https://doi.org/10.36074/grail-of-science.19.01.2024.031> [39].

Структура і обсяг роботи

Робота складається з вступу, чотирьох розділів, висновків списку використаних джерел, додатків. Обсяг роботи становить 75 сторінок, обсяг використаної літератури – 39 джерел.

Перший розділ містить аналіз існуючих сучасних комп'ютерних технологій, які допомагають слабоворим людям поліпшити якість життя. Надано огляд найсучасніших програм створених для людей з вадами зору і незрячих користувачів комп'ютерів та мобільних телефонів.

У другому розділі описано засоби та технології навігаційного забезпечення для людей з порушеннями зору, проводиться дослідження проблеми класифікації графічних образів, наведено алгоритмічні побудови та рецепторну структуру сприйняття інформації.

У третьому розділі предстало проблеми класифікації графічних образів та методи розпізнавання графічних образів, описано алгометричні моделі розпізнавання образів.

Четвертий розділ представляє програмну реалізацію загальних модулів комп'ютерного застосування розробленої системи допомоги для людей з вадами зору.

Додатки містять головні елементи коду.

РОЗДІЛ 1

КЛАСИФІКАЦІЯ СИСТЕМ, ПРИСТРОЇВ ТА МЕТОДІВ РОЗПІЗНАВАННЯ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

Люди з порушеннями зору часто використовують навички орієнтації та рухливості (O&M). Зазвичай це індивідуальне навчання з фахівцем з мобільності, і людина з ослабленим зором вивчає методи та навички, які допоможуть йому безпечно та незалежно подорожувати у своєму середовищі.

У контексті навчання орієнтації та мобільності орієнтація відноситься до знання людиною того, де вона знаходиться в навколишньому середовищі і як вона впевнено та безпечно дістанеться місця призначення у незнайомій обстановці.

Допоміжні засоби отримали загальну назву Assistive Technologies. Асистивні технології пов'язані з технологіями, обладнанням, пристроями, апаратами, послугами, системами, процесами та модифікаціями навколишнього середовища, які дозволяють їм долати різні фізичні, соціальні, інфраструктурні бар'єри та бар'єри доступності на шляху до незалежності та жити активним, продуктивним та незалежним життям на рівних. Дослідження допоміжних навігаційних засобів для сліпих або людей з вадами зору дуже пов'язано з тим, що його сфера охоплення тягнеться від фізіологічних факторів, пов'язаних із втратою зору, до людських факторів.

1.1. Допоміжні засоби для подорожей

Обмеження звичайних допоміжних рішень призвели до розвитку підтримуваних технологією рішень, які можна використовувати для направлення користувача шляхом надання необхідних навігаційних даних шляхом виявлення перешкод. Величезний потенціал, що з'явився в обчислювальних та комунікаційних платформах, можна розглядати як перевагу його використання при розробці рішення для навігації та доступності для людей із вадами зору.

Класифікація навігаційних систем, пристроїв та методів розпізнавання включає три категорії: електронні засоби орієнтації (EOA), пристрої визначення місця розташування (PLD) та електронні засоби пересування (ETA).

Електронні засоби орієнтування (EOA) призначені для допомоги сліпим і слабозорим людям у пошуку навігаційного шляху. Камера та різні датчики зазвичай поєднуються для визначення перешкод та шляхів. Системи EOA зазвичай потребують додаткової інформації про довкілля. Обмеження EOA полягають у складності увімкнення складного обчислювального пристрою з легким пристроєм управління в реальному часі.

Пристрої позиціонування (PLD) використовуються для визначення точного розташування пристроїв, що використовують технології глобальної системи позиціонування (GPS) та географічної інформаційної системи (ГІС). Комбінація навігаційних систем на основі GPS та ГІС може використовуватися для направлення користувача від поточного розташування до пунктів призначення. Однак, ця комбінація не повністю працюватиме для людей з порушеннями зору, тому що система не може допомогти користувачеві уникнути перешкод перед ним. Отже, PLD зазвичай необхідно комбінувати з іншими датчиками виявлення перешкод. Ще одне обмеження PLD полягає в тому, що вони повинні приймати сигнали від супутників GPS, тому їх можна використовувати лише на відкритому повітрі, а не в приміщенні.

Електронні засоби пересування (ETA) – це пристрої загального призначення, які допомагають людям з порушеннями зору уникати перешкод. ETA може покращити дальність виявлення перешкод та орієнтирів, а також може дати кращу орієнтацію. В результаті він може полегшити безпечні, прості та зручні завдання навігації. ETA складається з сенсорного блоку (блоків) введення для отримання вхідних даних із навколишнього

середовища та одного або кількох способів зворотного зв'язку для надання користувачеві інформації, яка може допомогти у навігації.

1.2. Сенсорні входи

Загальні сенсорні входи, що використовуються в ЕТА - це звичайна камера (або камера мобільного телефону), камера глибини, радіочастотна ідентифікація (RFID), маяк Bluetooth, ультразвуковий датчик, інфрачервоний датчик і т.д.

Камера сучасного смартфона може видавати непогані з якості знімки, до того ж вони невеликого розміру. Але основне обмеження звичайних камер смартфонів полягає в тому, що вони не надають інформації про глибину, тому такі системи не можуть визначити відстань від користувача до перешкод. Зазвичай зображення з камер обробляються для виявлення перешкод попереду під час навігації.

Камера глибини надає інформацію про відстань. Серед систем розпізнавання камери глибини Microsoft Kinect зазвичай використовується як основне обладнання для розпізнавання в системах глибокого аналізу зору. Microsoft Kinect використовує нову камеру Time-of-Flight (ToF) для обчислення глибини. У порівнянні з чисто двовимірними зображеннями, зображення глибини можуть надати більше інформації про перешкоди. Одним із основних обмежень камер Kinect є те, що їх не можна використовувати в умовах інтенсивного освітлення. Крім камер Kinect для аналізу глибини використовуються інші камери, засновані на глибині, - це Light Detection and Ranging (LiDaR). Недоліком як систем на основі Kinect, так і LiDaR є їхній надмірний розмір, який не зручний у використанні, а також незручний для навігації людиною. Але останні смартфони також мають вбудований датчик глибини на додаток до звичайної камери. Особливості, такі як менший розмір, портативність,

Радіочастотна ідентифікація (RFID) відноситься до технології, в якій цифрові дані будуть закодовані в мітках або смарт-етикетках та перехоплені зчитувачем за допомогою радіохвиль. Технологія страждає від нестабільної точності сигналів, переривання сигналів, колізії зчитувачів та/або тегів, низької швидкості зчитування і т.д. Більше того, користувач повинен знати про місцезнаходження зчитувача RFID у контексті навігації. [11].

Маяк - це тип RFID, що використовується для ідентифікації об'єктів за допомогою радіочастоти. Це більше схоже на активний RFID, тому що він не залежить від зчитувача RFID для вироблення енергії. Маяки Bluetooth – це невеликі апаратні пристрої, які постійно передають сигнали Bluetooth із низьким енергоспоживанням (BLE). Маяк BLE передає пакети даних через проміжки часу. Ці пакети даних є мобільним додатком або встановленими сервісами на смартфонах поблизу. Bluetooth Low Energy передає менше даних на менший діапазон, тому споживає менше енергії, ніж інші навігаційні пристрої. Маяки Bluetooth потребують високої вартості установки та обслуговування, оскільки приймачі або випромінювачі необхідно встановлювати по всій стелі.

Зв'язок ближнього радіусу дії (NFC) [9]. також ґрунтується на протоколах RFID. Основна відмінність NFC з RFID у тому, що NFC-пристрій може бути у ролі зчитувача, а й у ролі мітки. Також можлива передача інформації між двома пристроями NFC.

Ультразвукові навігаційні системи використовують ультразвук для вимірювання відстані до перешкод та повідомляють інформацію користувачеві за допомогою голосу чи вібрації. Однак одним з обмежень цих систем є те, що вони не можуть визначати точне розташування перешкод через широкий кут променя ультразвуку. Крім того, ці системи не можуть розпізнавати тип перешкоди (наприклад, автомобіль або велосипед) [22].

Інфрачервоний (ІЧ) датчик використовується для визначення певних характеристик довкілля шляхом випромінювання або виявлення ІЧ-сигналу.

Інфрачервоні датчики також можуть вимірювати тепло, що випромінюється об'єктом, і виявляти його рух. Недоліком пристроїв на основі ІЧ-випромінювання є те, що природне та штучне світло може заважати йому. Системи на основі ІЧ є дорогими в установці через велику кількість тегів, які необхідно встановити та підтримувати.

1.3. Модальності зворотного зв'язку

ЕТА з датчиками зору/не зору може виявляти перешкоди під час навігації. Ця інформація разом із підказками напряду має бути надіслана користувачам та допомагає їм у навігації. Найпоширенішими способами зворотного зв'язку є аудіо, тактильні або вібрації. Деякі системи використовують їхню комбінацію і таким чином надають користувачеві мультимодальну можливість отримання навігаційних підказок.

Звуковий зворотний зв'язок зазвичай надається у навігаційній системі з використанням навушників або динаміків. Недоліком звукового зворотного зв'язку є занепокоєння, яке завдається користувачеві, коли інформація переповнюється, або це може також дратувати, коли користувач пропускає звуки навколишнього середовища через слухові сигнали. Навушники з кістковою провідністю використовуються в багатьох системах навігації, щоб мінімізувати цю проблему. Гарнітура забезпечує передачу звуку у внутрішнє вухо, що дозволяє користувачеві сприймати звуковий сигнал, не блокуючи слуховий прохід.

Тактильний зворотний зв'язок також використовується в деяких системах навігаційного зворотного зв'язку, де зворотний зв'язок передається через ногу, кисть, руку, пальці або будь-які частини тіла користувача, де може відчуватись тиск. Це дозволяє користувачеві отримувати повідомлення про напрямки, а також уникати перешкод, відчуючи відчуття у різних точках тиску тіла. На відміну від методів звукового зворотного зв'язку, тактильний зворотний зв'язок може використовуватися, щоб не відволікати користувачів

через звуки навколишнього середовища, інші мобільні пристрої, щоб дати користувачам зворотний зв'язок при навігації.

1.4. Висновки до розділу 1

Сформовано аналіз сучасних існуючих пристроїв, методів та навігаційних систем, при подорожах для людей з вадами зору.

Свого часу невисокий розвиток пристроїв та методів розпізнавання при подорожах та навігації призвели до сучасного розвитку засобів технологічної підтримки рішень, які можна використовувати шляхом надання необхідної навігаційної допомоги слабозорій людині. Як свідчить розгляд досліджень класифікації навігаційних систем, пристроїв та методів існують три категорії: EOA, PLD та ETA. Встановлено що кожний пристрій позиціонування, або комбінація навігаційних систем на основі GPS та ГІС може використовуватися для направлення користувача від поточного розташування до пунктів призначення. Однак, ця комбінація не повністю працюватиме для людей з порушеннями зору, тому що система не може допомогти користувачеві уникнути перешкод перед ним. Отже, PLD зазвичай необхідно комбінувати з іншими датчиками виявлення перешкод. Ще одне обмеження PLD полягає в тому, що вони повинні приймати сигнали від супутників GPS, тому їх можна використовувати лише на відкритому повітрі, а не в приміщенні.

Багато з існуючих зараз систем обтяжливі або важковагові. Портативність є одним із важливих факторів, що визначають практичність та прийнятність таких пристроїв користувачами. Незважаючи на те, що багато систем інтегровані з мобільним телефоном, їх використання не може вважатися зручним для користувача, як очікувалося. Рішення має бути розроблено таким чином, щоб користувач міг носити систему із собою та використовувати її з більшою легкістю.

РОЗДІЛ 2

ОГЛЯД ЗАСОБІВ ТА ТЕХНОЛОГІЙ НАВІГАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЛЮДЕЙ З ПОРУШЕННЯМИ ЗОРУ

Огляд допоміжних навігаційних систем для людей з порушеннями зору включає наукові публікації приблизно за п'ятирічний період (2017–2021 рр.). Дослідження проводилося завдяки цифровим бібліотекам, а також відповідним базам даних з використанням комбінацій ключових слів «навігаційні системи для людей з вадами зору» та «навігаційні системи для сліпих». Основні технології, що використовуються в кожній з цих систем, відрізняються за конкретним середовищем. Кожен із них отримав свої переваги, а також обмеження. У цьому розділі класифікуються різні електронні засоби пересування, що розглядаються в цьому огляді, на основі технології, що лежить в їх основі. Класифікація проводиться за п'ятьма категоріями: (1) системи візуальних зображень, (2) системи невізуальних даних, (3) системи на основі карт, (4) системи з 3D-звуком та (5) рішення на базі смартфонів.

2.1. Системи візуальних образів

Навігація на основі зору або оптична навігація використовує алгоритми комп'ютерного зору та оптичні датчики, включаючи різні типи камер, для отримання візуальних характеристик з навколишнього середовища. Система намагається виявляти перешкоди за допомогою візуальних функцій, а потім допомагає користувачеві безпечно переміщатися, даючи вказівки, як їх уникнути. У літературі робилися спроби включити у конструкцію навігаційної системи різні пристрої/технології, що ґрунтуються на візуальних образах. Сюди входять системи, в яких використовується стереокамера, мережа IP-камер, камера RGB-D і т.д. Це можна класифікувати як пристрої/технології захоплення візуальних зображень.

2.1.1. Стереокамера

Бурбакіс Н.Г. і Кавракі Д. представили методологію навігації, використовувану інтелектуальним помічником на ім'я Tyflos. Система Tyflos оснащена двома камерами огляду та захоплює зображення з навколишнього тривимірного середовища або за командою користувача, або у безперервному режимі (відео). Потім він перетворює ці зображення на відповідні словесні описи, які використовуються для встановлення вербального спілкування з користувачем.

Навігаційна система, поєднує біноклярну камеру, інерційний вимірювальний блок (IMU) та навушники в велосипедному шоломі. Коли об'єкт виявляється в певному місці, він перетворюється на джерело звуку і передається користувачеві через навушники. Конкретна техніка називається бінауральним рендерингом і відноситься до техніки створення звуків, які можна локалізувати за напрямом та відстанню за допомогою навушників. Система призначена для роботи лише на відкритому повітрі.

2.1.2. Мережа IP-камер

У системі запропонованій [12], камери були встановлені на стелі кожної кімнати. Фотографії, зроблені з довкілля, було проаналізовано системою віддаленої обробки з допомогою алгоритму комп'ютерного зору. Використовуючи простий інтерактивний мобільний додаток, встановлений на смартфон, користувач зможе дістатися пункту призначення. На рис.2.1 схематично показано, як працює система. Основна проблема із системою пов'язана з витратами на встановлення різних IP-камер у приміщенні.



Рис.2.1 Архітектура системи з урахуванням IP-камери [13]

2.1.3. VS-LAM

Візуальна одночасна локалізація та картографування (VS-LAM) — це технологія, яку можна використовувати для визначення розташування та позиціонування з використанням візуальних вхідних даних із камери. Оскільки для цієї технології потрібен лише один датчик камери і вона може працювати без інших датчиків, вона набула великої популярності в області проектування навігаційних систем. Відповідним чином [13] запропонували рішення для внутрішньої навігації, що використовує алгоритм VSLAM для вирішення проблем, пов'язаних з локалізацією всередині приміщень та будівництвом віртуальних сліпих доріг. Система використовувала динамічну стратегію вибору **поділей**, щоб допомогти користувачам орієнтуватися, уникаючи перешкод шляху. VSLAM також використовувався в [14]. Тут система пов'язана з хмарним сервером, а основними компонентами є шолом із стереокамерами, смартфон на базі Android, веб-додаток, а також платформа хмарних обчислень. Загальний вигляд системи показано (рис 2.2) . Оцінки показали, що є можливості для підвищення точності виявлення і розпізнавання об'єктів.



Рис.2.2 Хмарна та навігаційна система на основі машинного зору [14]

2.1.4. Камери RGB-D

Інтелектуальний засіб ситуаційної обізнаності та навігації (ISANA) [15] був електронним прототипом SmartCane, що використовує Google Tango [16] планшет як мобільна обчислювальна платформа. Використовуючи вбудовану камеру RGB-D, в ISANA застосовувався ефективний метод виявлення та запобігання перешкодам на основі алгоритму фільтра Калмана (TSM-KF). Вони також розробили мультимодальний людино-машинний інтерфейс (HMI) із взаємодією мови та звуку та надійною тактильною взаємодією через електронну SmartCane.

Відповідним чином [17] запропонували кіберфізичну систему, яка використовує датчик RGB-D для виявлення об'єктів, а також для розуміння сцен. У системі доступні як слуховий, так і вібротактильний режими зворотного зв'язку. Системі потрібний доступ до Інтернету для виконання обчислень, але вона може працювати як у приміщенні, так і на відкритому повітрі.

Система, запропонована [18], також використовували камеру RGB-D для полегшення навігації всередині приміщень для людей з порушеннями

зору. Система також складається зі стандартного ноутбука, на якому встановлено навігаційне програмне забезпечення, інтерфейсу користувача смартфона і жилета з тактильним зворотним зв'язком. На основі голосової інструкції користувача система може визначити початкову та кінцеву точки. Крім зберігання раніше згенерованих карток для навігації, система також може створювати картки під час подорожі користувача.

Система, запропонована в [19] використовували комбінацію носимих і соціальних датчиків для забезпечення навігаційних напрямів сліпих користувачів. Використовуючи датчики, такі як RGB-D і IMU, система забезпечує виявлення об'єктів в межах прямої видимості за допомогою режимів звукового і вібраційного зворотного зв'язку. А на глобальному рівні користувачі приймають рішення, використовуючи соціальні сенсори (такі як повідомлення у Facebook, твіти у Twitter і т.д.). Соціальні датчики можуть аналізувати його вміст, розміщений іншими, і завдяки цьому сліпі користувачі можуть отримувати попередження про різні інциденти або аварії, що сталися в певному місці. Використовуючи цю інформацію, сліпий користувач може вирішити, чи слід слідувати йому маршруту, запланованому навігаційною системою.

2.1.5. Майкрософт Кінект

Microsoft Kinect поставляється з камерами RGB-D, що викликає у дослідників великий інтерес до його використання в розробці навігаційних систем для людей з вадами зору. Тож варто окремо згадати про події, що відбулися у цій галузі. Kinect - це лінійка пристроїв введення, чутливих до руху, вироблених Microsoft, які підходять для виявлення об'єктів і таким чином можуть використовуватися для підтримки навігації. Він також підтримує великий набір функцій та може працювати в умовах низького освітлення. Система, запропонована, використовували алгоритм, який бере вхідні дані від Microsoft Xbox Kinect 360. Це допомагає зробити 3D-карту внутрішніх приміщень та визначити глибину перешкоди/людини. Подібно до

цього, [20] також представила систему обходу перешкод для сліпих за допомогою камери глибини Kinect. Зображення глибини з камери Kinect оброблялися методом середнього на основі вікон, а потім використовувалися для виявлення різних перешкод. Коли система розпізнає перешкоду, вона надсилає голосовий зворотний зв'язок користувачеві через навушники.

2.1.6. LiDaR

Оскільки використання LiDaR стало популярним у розробці автономних транспортних засобів та робототехніки, дослідники також спробували поекспериментувати з його можливостями у допоміжному навігаційному дизайні для людей із вадами зору. Система LiDaR Assist Spatial Sensing (LASS), запропонована [21] використовує датчик LiDaR для визначення перешкод, а потім перетворює його на стереофонічний звук різної висоти. Просторова інформація від датчика, така як орієнтація та відстань до перешкод, перекладається як відносний крок. Система, запропонована в також повідомляється про використання датчика LiDaR, інтегрованого з білою тростиною. У сканування тростини для виявлення перешкод є один недолік, крім ваги та розміру, що робить систему трохи незручною.

2.2. Невізуальні системи даних

У цьому розділі обговорюються навігаційні системи, які не використовують алгоритми зору або оптичні датчики як основний вибір. Пропонуються різні системи з використанням різних датчиків, таких як ультразвукові, маяки, ІЧ-датчики і т.д. Незважаючи на те, що існують системи, які використовують як візуальні, так і невізуальні дані, системи, що обговорюються тут, в основному залежать від невізуальних функцій для надання користувачам орієнтирів при навігації.

2.2.1. BLE-маяки

Системи, що використовують маяки Bluetooth, неодноразово згадувалися в літературі. На думку Наір та ін. [22] запропоновано гібридне рішення для позиціонування та навігації, яке поєднує в собі обидва маяки Bluetooth з низьким енергоспоживанням (BLE).

У системі GuideBeacon [23] смартфон використовувався для взаємодії з маяками на основі Bluetooth, розміщеними в різних місцях всередині приміщень, які можуть допомогти користувачам у визначенні напрямків навігації. Деякі поліпшення, необхідні в запропонованій системі, пов'язані з інтерфейсом користувача і навігаційними модулями.

2.2.2. Інтернет речей на основі (IoT)

Інтернет речей (IoT) — це взаємодія між різними системами, які можуть передавати дані через мережу, не вимагаючи будь-якої форми взаємодії людини чи машини. Про навігаційні системи, що використовують концепцію IoT, повідомлялося після того, як вони стали популярними для різних додатків.

«Indriya» — інтелектуальна система навігації для людей з вадами зору» [24] являє собою портативний пристрій, що використовується в поєднанні з розумною тростиною. Система може виявляти перешкоди попереду на відстані до трьох метрів та розрізняти людей та об'єкти з точністю до 80%. Він може забезпечувати як вібраційне, так і голосове сповіщення перед можливим зіткненням. Indriya використовує менше датчиків для реалізації на основі IoT з підтримкою платформи Android. Система має тенденцію помилок при виявленні щаблів, виявленні нахилу і т.д.

«Сліпий гід: рішення для допомоги сліпим у будь-який час і в будь-якому місці» [25] також заснований на Інтернеті речей, який може працювати з існуючими рішеннями, такими як біла тростина, щоб допомогти слабозорим орієнтуватися як у приміщенні, так і на вулиці. Якщо бездротовим датчиком виявлено перешкоду, сигнал буде надіслано на центральний пристрій, яким є

плата Raspberry Pi. Після ідентифікації об'єкта користувач буде проінформований про його ім'я, а також про відстань від користувача за допомогою голосового зворотного зв'язку. Важливим обмеженням прототипу є те, що для розпізнавання об'єктів потрібний доступ до Інтернету, що робить систему залежною від місця доступу до мережі передачі даних.

2.2.3. Ультразвукові датчики

Навігаційні системи, засновані на ультразвукових датчиках, можна як найпоширеніший вибір у дизайні після візуальних (камерних) рішень. Пропоновані системи, що використовують цю технологію, працюють спільно з деякими електронними платами, такими як Raspberry Pi або Arduino. [25].

«Ультразвукова сліпа палиця для повністю сліпих людей, що дозволяє уникати будь-яких перешкод» [Цитата26] може класифікувати перешкоди та навколишні ситуації за допомогою ультразвукових датчиків. Система може надавати пріоритетну інформацію користувачеві за допомогою вібро- та звукового оповіщення. Система була розроблена для виявлення мокрої підлоги, перешкод на рівні підлоги та рівня колін. Одним із обмежень NavGuide є те, що він не може виявити яму або спуск. Крім того, NavGuide може визначити мокру підлогу тільки після того, як користувач настане на неї.

Дж. Боренштейн и И. Ульрих « Применение технологий мобильных роботов для помощи людям с нарушениями зрения », GuideCane використовували ультразвукові датчики для виявлення перешкод під час навігації, а підключений вбудований комп'ютер міг визначати напрямок руху, керований системою, а також користувача, який її використовує. Обмеження GuideCane полягають у тому, що він не може виявляти перешкоди, що нависають, такі як стільниці, а також не може виявляти важливі елементи, такі як межі тротуарів.

2.2.4. ІЧ-датчики

ІЧ-датчики відрізняються низьким енергоспоживанням та низькою вартістю порівняно з ультразвуковими датчиками, і з цих причин компанія також експериментувала з конструкцією навігаційної системи [27]. Крім того, IR використовується у поєднанні з іншими технологіями, такими як Google Tango та Unity. [28].

Марцек П. та Кос А., « Точная навигационная система с низким энергопотреблением для слепых с инфракрасными датчиками », запропоновали в [29], рішення з використанням інфрачервоних датчиків, які допомагають виявляти різні об'єкти, такі як будівлі, стіни і т.д. Пристрій повинен бути розміщений на руках користувача та може передавати навігаційні сигнали за допомогою вібрації. Ці сигнали надають інформацію про кроки руху та найближчі загрози.

2.3. Картографічні системи

Користувачі з порушенням зору використовують різні тактильні інструменти, такі як карти з опуклими точками, невеликі прототипи або магнітні дошки після навчання експлуатації та технічного обслуговування. Були запропоновані різні мультимодальні карти для полегшення навігації сліпих і людей з вадами зору. Ці інструменти є ефективним способом просторового навчання за відсутності зору, але мають обмеження. Одним із таких є неможливість оновлення вмісту карток. Для подолання цих обмежень було розроблено доступні інтерактивні карти. [30]. Використовуючи підхід до спільного проектування, автори Ж. Альбуи-Перруа і др «На шляху до мультисенсорної карти доповненої реальності для сліпих і людей з вадами зору: підхід до спільного проектування », [31] спроектували та розробили карту доповненої реальності, яку можна використовувати на заняттях з експлуатації та технічного обслуговування. Цей прототип поєднує у собі проекцію, аудіовихід та використання тактильних жетонів. Отже, це дозволяє людям із порушеннями зору як досліджувати карту, так і створювати її.

Карти SmartTactMaps, запропоновані [32], були засновані на смартфонах цей підхід для допомоги сліпим людям у вивченні тактильних карт. Тривимірна карта навколишнього середовища, створена з використанням датчика RGB-D, була опублікована в [33]. Система може отримувати семантичну інформацію із зображень RGB, щоб допомогти людям з порушеннями зору орієнтуватися вдома.

Надрукована на 3D-принтері аудіовізуальна тактильна карта під назвою LucentMaps [34] була також запропонована для людей з порушеннями зору. Автори ж заявляли про спрощення комбінації мобільних пристроїв із фізичними тактильними картами. Система ВізМап [35] використовує комп'ютерний зір та краудсорсинг для збирання різної інформації у приміщенні. Він використовує зрячих добровольців на місці, щоб знімати різні відео та використовувати їх для створення тривимірної просторової моделі. Ці відеокадри семантично маркуються і вбудовуються в реконструйовану 3D-модель, яка може формувати просторове подання навколишнього середовища.

2.4. Системи із 3D-звуком

Про систему Sound of Vision повідомляється в [36] являє собою сенсорний заміщувальний пристрій, що носить, який допомагає людям з порушеннями зору орієнтуватися, створюючи і передаючи слухові і тактильні уявлення про навколишнє середовище. Користувач отримуватиме як звуковий, так і тактильний зворотний зв'язок. Система потребує поліпшення показників зручності використання та точності.

Електронний дорожній посібник на основі Stereo Vision (SVETA) є електронним дорожнім посібником, що складається з головного убору зі стереокамерами і навушниками. Пропонується процедура соніфікації для зіставлення зображення диспаратності із музичним звуком у стереофонічному режимі. Кожному такому музичному звуку відповідає деяка

інформація про особливості перешкоди, з якою стикається користувач. Система відмінно працює у приміщенні. Необхідність використання системи обертається недоліком для користувачів-початківців. Цільові користувачі повинні пройти навчання різним значенням стереомузальних звуків, перш ніж використовувати його [37].

2.5. Рішення на базі смартфонів

Навігаційні рішення на основі смартфонів забезпечують мобільність та зручність для користувачів. У цьому розділі приводяться та описуються різні рішення, запропоновані на платформі смартфона для людей з вадами зору.

NavCog3, запропонований [36], була внутрішньою навігаційною системою, яка надавала покрокові інструкції та негайний зворотний зв'язок при виявленні неправильної орієнтації. Він також надає інформацію про визначні пам'ятки та деякі визначні пам'ятки у прилеглих місцях. Система забезпечує звуковий зворотний зв'язок із користувачами. На Рис.2.3 показано загальний вигляд системи NavCog3 . А. Ганц , Дж. М. Шафер та ін. [Цитата37] запропонували програму PERCEPT-II, у якій цільовий користувач отримує навігаційні інструкції до обраного пункту призначення при дотику до певних орієнтирів у додатку, встановленому на мобільному пристрої. Місця призначення були позначені мітками Near Field Communication (NFC). рис 2.4 ілюструє архітектуру PERCEPT-II. У системі є обмеження,



Рис 2.3 Огляд системи Navcog3 [37]



Рис 2. 4 Архітектура PERCEPT-II.

Лин Б.-С., та ін. "Проста система навігації на основі смартфона для людей з порушеннями зору", [9] запропонували додаток для смартфона, який можна інтегрувати із системою розпізнавання зображень для створення допоміжної навігаційної системи. Залежно від доступності мережі, система може вибрати два режими роботи: онлайн-режим і автономний режим. Коли

система ініціюється, смартфон захоплює зображення, а потім надсилає його на сервер для обробки. Сервер використовує деякі алгоритми глибокого навчання для розпізнавання різних перешкод. До основних обмежень системи належать високе енергоспоживання та необхідність високошвидкісного підключення до мережі.

Система TARSIOUS [38], спрямована на розширення можливостей користувачів у візуальному розумінні зовнішнього середовища. Компоненти системи включають додаток TARSIOUS для мобільних пристроїв, веб-сервер, центр віддаленої допомоги, а також мітки Bluetooth LE/iBeacon, розміщені вздовж вулиць у точках інтересу. До основних проблем системи TARIUS відноситься розміщення Bluetooth-маяків вулицями, що призводить до високої вартості, а також може призвести до перешкод сигналу.

«ENVISION: Допоміжна навігація для людей з вадами зору смартфонів», [10] використовується спеціальний метод для надійного і точного виявлення статичних та динамічних перешкод у потоковому відео в реальному часі, записаному смартфоном із середньою апаратною потужністю. Систему можливо покращити, якщо модулі розпізнавання та класифікації перешкод допоможуть цільовим користувачам краще зрозуміти довкілля.

Проект «Активний зір з людиною в контурі для людей з вадами зору» (ActiVis), розроблений пропонує мультимодальний інтерфейс користувача, який використовує звукові та вібраційні сигнали для передачі навігаційної інформації цільовому користувачеві. Поточна реалізація системи ActiVis є додатком для Android, заснований на пристрої Tango [16] і кістковопровідний гарнітур. Систему можливо покращити, якщо вона зможе адаптуватися до параметрів зворотного зв'язку в режимі реального часу, що може підвищити продуктивність системи.

R. Manduchi і J. Coughlan, « (Комп'ютерне) бачення без зору », система невізуальної підтримки для навігації», складається з тактильного пояса, а

також персонального цифрового помічника (КПК), який запускає програму Wayfinder. Додаток керує інформацією про місцезнаходження та маршрут. Після визначення напрямку шляху інформація надсилається на тактильний дисплей. Вібратори в поясі можуть надавати користувачеві інформацію про напрямки навігації.

2.6. Висновки до розділу 2

У цьому огляді були розглянуті різні навігаційні системи, які пропонуються людям з порушеннями зору, і класифіковані на основі базової технології.

Очевидно, що навіть з розвитком та різноманітністю технологічних рішень допоміжних навігаційних пристроїв вони ще не набули такого широкого поширення, а також невисоке визнання користувачів. Відповідний вибір методів виявлення об'єктів у реальному часі: в останні роки рішення для виявлення об'єктів на основі глибокого навчання удосконалюються, і кількість часу, необхідна для виконання необхідної задачі, скорочується. У цьому сценарії важливо враховувати використання ефективного та відповідного методу виявлення об'єктів на відповідній платформі, яка може підтримувати операції у реальному часі.

Наявність декількох варіантів механізмів зворотного зв'язку: якщо система забезпечує лише один режим зворотного зв'язку, він може бути некорисним у різних випадках. Деякі люди покладаються на слуховий режим, а інші на тактильний або вібраційний режим. Але будуть ситуації, коли кожен із цих режимів стає актуальним залежно від оточення. Таким чином, якщо є можливість використовувати кілька способів зворотного зв'язку, користувач зможе вибрати один із них залежно від ситуації або середовища. Це зробить систему ефективнішою у різних середовищах. Якщо мультимодальна система зворотного зв'язку може бути реалізована на основі адаптації користувацьких переваг відповідно до ситуацій та умов навколишнього середовища, вона

буде корисна для користувачів, навіть незважаючи на складнощі в реалізації. Скорочення тривалого часу навчання: кількість часу, необхідне ознайомлення з системою, є важливим орієнтованим користувача чинником розробки навігаційної системи для людей із вадами зору. Обидва вони необхідні людям з вадами зору для освоєння більшості існуючих сьогодні допоміжних навігаційних систем. Один з аспектів, який необхідно враховувати в цьому випадку, полягає в тому, що користувачі не повинні мати особливих труднощів при вивченні або використанні нової системи. Зручність у транспортуванні та використанні: Обсяг інформації, яку сприймає користувач: Основна мета навігації — безпечно дістатися пункту призначення. Але користувача більше буде цікавити те, що все відбувається просто і ефективно. Вони можуть знадобитися для отримання інформації про зміни в навколишньому середовищі під час навігації, таких як дорожні затори, перешкоди або будь-які запобіжні ситуації тощо. Користувачі повинні бути проінформовані у потрібній кількості та у потрібний час про навколишнє середовище під час навігації. Щоб бути ефективним, навігаційному рішенню рекомендується зосередитися на передачі конкретної інформації про довкілля.

Належне управління та безпека особистих даних: управління приватними та особистими даними слід враховувати при розробці навігаційного пристрою для сліпих та людей з вадами зору. Незрячий користувач повинен мати можливість налаштовувати навігаційний пристрій відповідно до того, яку інформацію система використовує для виконання своїх процесів та які дані передаються у мережі. Це налаштування можна виконати відповідно до уподобань користувача та контексту, в якому використовується система.

РОЗДІЛ 3

ПРОБЛЕМИ КЛАСИФІКАЦІЇ ГРАФІЧНИХ ОБРАЗІВ

3.1. Методи розпізнавання графічних образів

Існує три основні етапи у методах розпізнавання графічних образів: перший - попередня фільтрація і підготовка зображення, другий - логічна обробка результатів фільтрації, і третій - застосування алгоритмів для прийняття рішень на основі логічної обробки. Між цими етапами існують умовні межі, і для вирішення конкретних завдань може бути достатньо використовувати лише два або навіть один із цих етапів.

Фільтрація включає методи, які дозволяють виділити області на зображеннях, не проводячи їх докладний аналіз. Більшість цих методів застосовують однакове перетворення до всіх точок зображення. На етапі фільтрації сам аналіз зображення відсутній, але точки, які пройшли через фільтрацію, можна розглядати як області з особливими характеристиками.

Найбільш простий метод перетворення - це застосування бінаризації до зображення шляхом встановлення порогового значення. Для RGB-зображень або зображень у відтінках сірого порог може бути встановлений на основі значень кольору.

Існують ситуації, коли для ідеальних завдань цього перетворення вистачає.

Припустимо, що ми маємо завдання автоматично виділити об'єкти на білому аркуші паперу. Процес самої бінаризації, включаючи вибір порогового значення, великою мірою визначається у даному випадку. У зазначеному прикладі зображення було піддане бінаризації, використовуючи середнє значення кольору. Зазвичай для бінаризації використовується алгоритм, який адаптивно вибирає поріг.

Використання бінаризації може призвести до цікавих результатів при роботі з гістограмами, особливо у випадку, коли ми працюємо з зображенням

в просторі кольорів HSV, а не RGB. Наприклад, можна використовувати бінаризацію для сегментації кольорів. За цим принципом можна побудувати як детектор мітки, так і детектор шкіри людини.

Класичні методи фільтрації, такі як Фур'є, ФНЧ та ФВЧ, які використовуються в радіолокації та обробці сигналів, успішно застосовуються в різноманітних задачах розпізнавання образів. В радіолокації традиційний метод, який мало застосовується в чистому вигляді для зображень, - це перетворення Фур'є, зокрема - багатопослідовне перетворення Фур'є (БПФ). Єдиним винятком є компресія зображень, де застосовується одновимірне перетворення Фур'є. У великій більшості випадків для аналізу зображень використовується двовимірне перетворення, бо одновимірний підхід зазвичай недостатній і вимагає більше обчислювальних ресурсів.

Лише небагато людей розглядають його детально, оскільки зазвичай набагато швидше і простіше використовувати готові фільтри для обробки областей, налаштовані на високі (ФВЧ) або низькі (ФНЧ) частоти. Хоча такий підхід не надає можливості провести аналіз спектра, у більшості випадків конкретні завдання відеоспостереження вимагають не аналізу, а отримання результату. Серед простих прикладів фільтрів, що виділяють низькі (фільтр Калмана, див. рис. 3.1) та високі (Фільтр Габора, див. рис. 3.2) частоти, для кожної точки зображення вибирається вікно і здійснюється згортка з фільтром такого ж розміру, призводячи до нового значення точки. Під час реалізації ФНЧ і ФВЧ отримуються зображення, представлені наступним чином: Однак що станеться, якщо для згортки з сигналом використовувати довільну характеристичну функцію? Такий підхід відомий як "Вейвлет-перетворення". Хоча це визначення вейвлетів не є строго правильним, традиційно в багатьох випадках вейвлет-аналізом називається процес виявлення довільного патерну на зображенні шляхом згортки з моделлю

цього патерну. У вейвлет-аналізі використовуються різноманітні класичні функції.

Це включає в себе вейвлети, такі як Хаар, Морлі, мексиканський капелюх і інші.

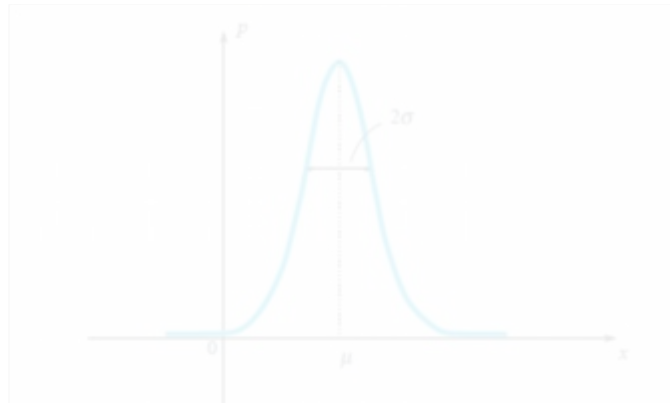


Рисунок 3.1 – Фільтр Калмана

Використання розширеної трактування вейвлетів проілюстроване задачею виявлення відблиску в зображенні оці, де використовується сам відблиск як вейвлет. Традиційні вейвлети зазвичай використовуються для стиснення або класифікації зображень.



Рисунок 3.2 – Фільтр Гарбора

Цікавим типом фільтрів є фільтрація функцій, що представляє собою математичні фільтри для виявлення простих математичних функцій на зображенні, таких як пряма, парабола чи коло. У цьому випадку створюється

зображення, на якому для кожної точки вихідного зображення відображається низка функцій, які можуть її утворювати.

Найбільш стандартним перетворенням є перетворення Хафа для прямих (рис. 3.3). У цьому методі для кожної точки (x; y) відзначається безліч точок (a; b), які представляють собою рівняння прямих у вигляді $y = ax + b$ і відповідають вихідній точці.

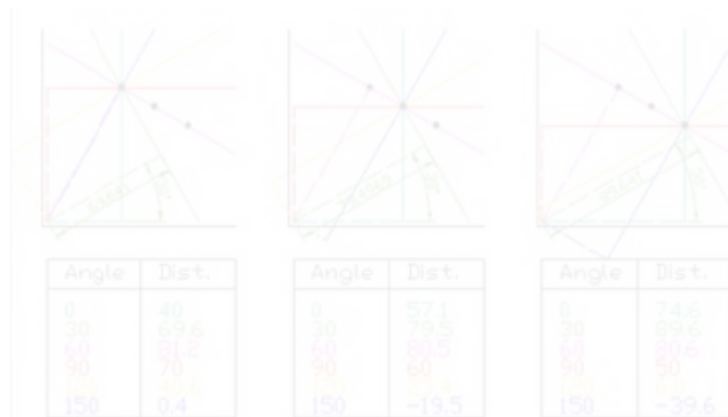


Рисунок 3.3 – Перетворення Хафа для прямих

Перетворення Хафа дозволяє виявляти різноманітні параметризовані функції, такі як кола, наприклад. Існує модифіковане варіант цього перетворення, який розширює можливості пошуку різноманітних геометричних фігур. Це перетворення користується популярністю серед математиків. Однак у контексті обробки зображень воно, на жаль, не завжди є ефективним. Його робота може бути дуже повільною, а також високою чутливістю до якості бінаризації. Навіть у ідеальних умовах часто доводиться використовувати альтернативні методи.

Перетворення Радона є аналогом перетворення Хафа для прямих. Його обчислення використовує преобразування Фур'є, що призводить до збільшення ефективності в умовах, коли кількість точок значно велика. Крім того, його можна успішно застосовувати до не бінаризованих зображень.

Серед видів фільтрів особливим є фільтрація кордонів і контурів. Контурні лінії стають дуже цінними, коли ми переходимо від обробки зображення до розпізнавання об'єктів на цьому зображенні. У випадку, коли об'єкт має складну структуру, але добре визначається, вирізання його контурів часто є єдиним способом ефективної обробки.

Є різноманітні алгоритми, спрямовані на вирішення завдання фільтрації контурів:

Найбільш поширеним є метод, відомий як Кенні, його ефективність підтверджена, і його реалізація включена в OpenCV. (Справді, в OpenCV також є фільтр Собеля, але він менше ефективний у виявленні контурів.)

Результати фільтрації піддаються логічній обробці. Фільтрація породжує набір даних, які придатні для подальшої обробки.

Перехід від застосування фільтрації до використання логіки включає в себе методи математичної морфології. Зазвичай, це представляє собою базові операції нарощування і ерозії бінарних зображень. Ці методи призначені для видалення шумів з бінарних зображень шляхом збільшення чи зменшення наявних елементів. Методи математичної морфології також служать основою для розробки алгоритмів оконтурювання, хоча зазвичай вони використовуються у поєднанні з гібридними або зв'язаними алгоритмами.

Кордони, які були отримані, легко перетворюються в контури. У випадку алгоритму Кенні цей процес відбувається автоматично, але для інших алгоритмів необхідна додаткова операція бінаризації.

Унікальність об'єкта визначається його контуром, який часто служить ключем для ідентифікації. Існує потужний математичний інструментарій, який дозволяє проводити цей аналіз, відомий як контурний аналіз.

Унікальні особливі точки представляють собою особливі характеристики об'єкта, які дозволяють порівнювати його як із самим собою, так і з об'єктами схожих класів. Деякі методи виявляють ці особливі точки в сусідніх кадрах, інші в змінних умовах освітлення або при великому

проміжку часу, а деякі спроможні визначати особливі точки, які залишаються стійкими навіть при поворотах об'єкта. Ми розпочнемо з методів, які ідентифікують менш стійкі особливі точки, але здатні до швидкого обчислення, і подалі будемо рухатися від менш складних методів до більш високорівневих:

Перший тип - це особливі точки, які залишаються стійкими протягом значного часового інтервалу, наприклад, секунд. Такі точки використовуються для відстеження об'єкта між послідовними кадрами відео або для визначення властивостей зображення з сусідніх камер. Цей тип особливих точок може включати локальні максимуми на зображенні, кути на зображенні (див. рис. 3.4), точки, де досягається максимум дисперсії, а також певні градієнти тощо.



Рисунок 3.4– Розпізнавання кутів на зображеннях

Другий тип особливих точок відрізняється тим, що вони залишаються стабільними навіть при зміні освітлення та невеликих рухах об'єкта. Головним чином такі точки використовуються для навчання та подальшої класифікації різних типів об'єктів. Наприклад, класифікатор пішохода або класифікатор особи використовує цей тип особливих точок у своїй системі.

Третій вид - це стабільні точки, і зараз ми розглянемо два методи, які забезпечують повну стабільність, а також їх модифікації. Ці методи - SURF і SIFT, дозволяють виявляти особливі точки навіть при повороті зображення. Хоча обчислення таких точок вимагає більше часу порівняно з іншими

методами, вони достатньо ефективні в обмеженому часі. На жаль, слід відзначити, що ці методи є предметом патентних прав.

Навчання. У третьому розділі ми розглянемо методи, які взаємодіють не безпосередньо зі зображенням, але дозволяють здійснювати прийняття рішень.

Для кожного зображення існує набір характеристик, які були виділені різними методами, такими як Харріса, HOG, SURF або вейвлет-перетворення.

Алгоритм навчання повинен створити модель, яка дозволить йому аналізувати нове зображення та приймати рішення щодо того, який об'єкт присутній на зображенні.

Як це виконується? Кожне тестове зображення представлено як точка в просторі ознак, де координати визначають вагу кожної конкретної ознаки на зображенні. Наприклад, можливими ознаками можуть бути: "Наявність очей", "Наявність носа", "Наявність двох рук", "Наявність вух" і так далі. Всі ці ознаки визначаються наявністю відповідних детекторів, які навчені розпізнавати частини тіла, подібні до людських.

У цьому просторі ознак точка, що відповідає людині, може мати координати [1; 1; 1; 1; ...]. Для мавпи ці координати можуть бути, наприклад, [1; 0; 1; 0; ...], а для коня - [1; 0; 0; 0; ...]. Класифікатор навчається на прикладах і розпізнає класи на основі цих ознак.

Проте, на деяких фотографіях може бути не виділено руки, на інших може бути відсутність очей, а на третій з'явиться людський ніс у мавпи через помилку класифікатора. Класифікатор, який навчається на прикладах людей, автоматично розподіляє простір ознак так, щоб сказати: якщо значення першої ознаки в діапазоні $0.5 < x < 1$, другої - $0.7 < y < 1$, і так далі, то це людина.

Фактично, мета класифікатора полягає в тому, щоб визначити в просторі ознак області, які є характерними для об'єктів певного класу.

Наведено приклад послідовного наближення до відповіді для одного з класифікаторів (AdaBoost) у двовимірному просторі.

Класифікація образів - це завдання, яке включає в себе визначення приналежності вхідного образу (наприклад, мовного сигналу або рукописного символу), представленого вектором ознак, до одного або декількох попередньо визначених класів.

Наприклад, визначення літер, розпізнавання мови, класифікація сигналів електрокардіограми та класифікація клітин крові.

Кластеризація, також відома як «категоризація» при розв'язанні завдань, відомих як класифікація образів «без вчителя», не передбачає наявності навчального набору даних із мітками класів. Алгоритм кластеризації, ґрунтований на подібності між образами, групує близькі образи в один кластер. Відомі випадки використання кластеризації включають стиск даних та дослідження властивостей даних.

Прогнозування - це завдання, що передбачає значення певного параметра на певний майбутній момент часу. Прогнози мають велике значення для ухвалення рішень у сфері бізнесу, науки та техніки. Прогнозування цін на фондовій біржі і прогнозу погоди є типовими прикладами задач прогнозування.

Задачі оптимізації можуть виникати в численних областях, таких як математика, статистика, техніка, наука, медицина і економіка.

Метою оптимізаційного алгоритму є знаходження рішення, яке відповідає обмеженням системи і досягає екстремуму (максимуму чи мінімуму) цільової функції.

3.2 Алгоритмічні побудови

Універсальність обчислювальних машин означає, що на них можна програмно втілювати будь-які алгоритми обробки інформації, включаючи

обчислювальні, управлінські, пошукові, а також алгоритми для доведення теорем, створення тривимірних графічних або аудіо композицій.

Проте важливо розуміти, що навіть обчислювальні машини та роботи не можуть вирішити будь-яке завдання у принципі. Достовірно доведено, що існують певні типи завдань, для яких не існує єдиного та ефективного алгоритму, що вирішує всі задачі даного типу. У цьому контексті деякі задачі не можна вирішити за допомогою обчислювальних машин. Цей факт сприяє кращому розумінню того, що можуть здійснювати машини та в яких сферах їхні можливості обмежені.

Серед характеристик штучних нейронних мереж основною є їх здатність до навчання, і вони здатні вивчати різноманітні методи. Більшість методів навчання мають загальні передумови і подібні характеристики. Процес навчання нейронних мереж нагадує інтелектуальний розвиток людської особистості. Проте слід відзначити, що можливості навчання штучних нейронних мереж обмежені.

Тим не менше, вже досягнуті вражаючі результати, такі як "мережа" Сейновскі, вказують на ряд практичних застосувань. Ця мережа вивчається так, щоб забезпечити відповідні виходи для різноманітних входів.

Кожний вхідний (або вихідний) безліч розглядається як вектор у цьому контексті. Процес навчання включає послідовне представлення вхідних векторів, супроводжуване одночасним налаштуванням ваг за певною процедурою. Під час навчання ваги мережі змінюються так, щоб кожен вхідний вектор відповідав вихідному вектору.

Алгоритми навчання можна розділити на алгоритми з учителем та без учителя.

У навчанні з учителем відбувається взаємодія з вчителем, який подає вхідні дані мережі, порівнює отримані виходи з необхідними значеннями та коригує ваги мережі, зменшуючи розходження. В цьому контексті, навчання з учителем вимагає наявності для кожного вхідного вектора відповідного

цільового вектора, що представляє бажаний вихід. Ці пари відомі як навчальні. Зазвичай мережа навчається на певній кількості навчальних пар. В кожному випадку подається вихідний вектор, обчислюється вихід мережі, порівнюється з відповідним цільовим вектором, і різниця (помилка) застосовується до мережі за допомогою зворотного зв'язку. Ваги змінюються відповідно до алгоритму, який прагне мінімізувати помилку. Вектори навчальної множини подаються послідовно, обчислюються помилки, і ваги підлаштовуються для кожного вектора до тих пір, поки помилка по всьому навчальному масиву не досягне прийнятного низького рівня.

Концепція навчання з вчителем була критикована за свою біологічну неправдоподібність. Важко уявити, як механізм навчання в мозку міг би порівнювати бажані і фактичні значення виходів, використовуючи корекцію через зворотний зв'язок. У порівнянні з цим, навчання без вчителя представляє собою більш правдоподібну модель для біологічних систем.

Стадії процесу навчання нового класу задач у Штучній Нейронній Мережі (ІНС) включають наступне:

1. Визначається постановка задачі та ідентифікуються основні параметри, які визначають область застосування.
2. Обирається парадигма нейронної мережі, яка включає в себе структуру вхідних даних, порогову функцію, конфігурацію мережі та методи навчання, найбільш відповідна для вирішення конкретного класу завдань. Зазвичай, сучасні нейропакети, нейроплати та еволюційні підходи дозволяють реалізувати не лише одну, а кілька базових парадигм.
3. Підготовлюється, можливо, більш обширний комплект навчальних прикладів, структурованих у формі наборів вхідних даних, пов'язаних з відомими вихідними значеннями. Вхідні дані для навчання можуть бути неповними та частково суперечливими.
4. Вхідні дані послідовно подаються на вхід Штучній Нейронній Мережі (ІНС), і отримане вихідне значення порівнюється з еталоном. Після

цього відбувається налаштування вагових коефіцієнтів міжнейронних з'єднань з метою мінімізації різниці між фактичним і бажаним виходом мережі.

5. Процес навчання повторюється до того часу, коли загальна помилка для всіх вхідних значень не досягне прийнятного рівня або Штучна Нейронна Мережа (ІНС) не встановиться в стаціонарний стан. Описаний метод навчання, що застосовується до нейроподібних мереж, відомий як "зворотне поширення помилки" (error backpropagation) і входить до числа класичних алгоритмів нейроматематики.

Після налаштування та навчання Штучна Нейронна Мережа (ІНС) може бути використана з реальними вхідними даними, не лише надаючи користувачеві правильне рішення, але й оцінюючи його рівень достовірності.

Існує різноманіття алгоритмів для розпізнавання образів. Алгоритм машинного навчання "впізнавання" образів, який ґрунтується на методі січних гіперплоскостей (див. рис. 3.5), використовує апроксимацію розділених гіперповерхонь "шматками" гіперплоскостей та включає в себе наступні основні етапи: 1. Навчання. Формування розділеної поверхні:

Створення січних площин, видалення непотрібних площин, видалення зайвих фрагментів площин.

2. Розпізнавання нових об'єктів.

При використанні методу паралельних варіантів, кілька машин одночасно і незалежно навчаються на одному і тому ж матеріалі. При розпізнаванні нових об'єктів кожна машина відносить ці об'єкти до певного образу, можливо, навіть різних образів. Остаточне рішення приймається шляхом "голосування" машин - об'єкт відноситься до того образу, до якого його віднесло більше число машин.

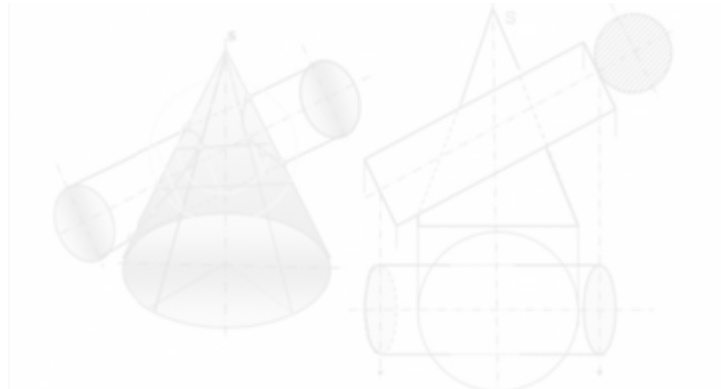


Рис 3.5 – Метод січних гіперплощин

Щоб покращити метод проведення січних площин і забезпечити більшу надійність розпізнавання, можна внести певні поліпшення. Як припущення, можна розглядати той факт, що якщо проводити січні площини близько до площини, яка проходить через середину прямої, що з'єднує об'єкт і опонент, перпендикулярно до цієї прямої, то отримана поверхня буде ближче до істинного розділу між образами. У використаному алгоритмі, заснованому на методі потенціалів, можна прив'язати певну функцію до кожного збудженого елемента поля рецепторів. Ця функція дорівнює одиниці на цьому елементі та зменшується в усіх напрямках від нього, що визначено функцією ϕ . Важливою відмінністю є те, що в цьому випадку R представляє відстань між двома сусідніми елементами поля рецепторів.

$$\phi(R) = \frac{1}{1 + \alpha R^2}$$

Для розрахунку використовується просте правило: кожен активований елемент поля рецепторів має власний потенціал, що дорівнює одиниці. Цей потенціал, в свою чергу, збільшує на $\frac{1}{2}$ потенціали всіх сусідніх (включаючи порушені) елементів по горизонталі, вертикалі і діагоналях. Проте такий метод кодування може бути удосконалений. Якщо пов'язати з кожним активованим елементом поля рецепторів певну функцію, яка дорівнює

одиниці на цьому елементі і зменшується в усіх напрямках від нього, схожу на потенціал ϕ , з відмінністю, що в цьому випадку R визначає відстань між двома сусідніми елементами поля рецепторів (рис. 3.6). Цю функцію можна апроксимувати ступінчастою функцією, яка залишається постійною в межах кожного рецептора і раптово змінюється на кордонах рецепторів.



Рис 3.6 – Метод потенцілів

Алгоритм впізнавання, що базується на методі потенцілів, можна реалізувати в дві фази:

1. Під час навчання система запам'ятовує коди всіх точок, які з'являються, та вказівки щодо того, якому об'єкту відповідає кожна точка.
2. Під час процесу впізнавання проводиться ідентифікація, і отримана інформація вказує, до якого образу належить закодована матриця.

Навіть при наявності певних обмежень у вихідній формі моделі перцептрона.

Розенблатт створив модель перцептрона, яка послужила основою для багатьох сучасних і більш складних алгоритмів навчання з вчителем. Один із прикладів такого перцептрону із рекурентною мережею показано на рис. 3.8. Він використовує алгоритм навчання з вчителем, де навчальна вибірка складається з безлічі вхідних векторів, кожен з яких має свій власний вектор мети. Компоненти вхідного вектора представлені безперервними значеннями, тоді як компоненти вектора мети є двійковими (0 або 1). Після завершення

навчання мережа отримує на вході набір безперервних входів і генерує необхідний вихід у вигляді вектора з бінарними компонентами.



Рис 3.7 - Багатошарова нейронна мережа

Процес навчання виглядає наступним чином:

1. Всі ваги мережі випадковим чином налаштовуються на невеликі значення.

2. Мережі подається вхідний навчальний вектор X , і для обчислення сигналу NET від кожного нейрона використовується стандартний вираз.

$$NET_j = \sum_i x_i w_{ij}$$

3. Розраховується значення порогової функції активації для сигналу NET від кожного нейрона відповідним чином.

4. Розраховується помилка для кожного нейрона шляхом віднімання вихідного сигналу від необхідного виходу:

$$error_j = target_j - OUT_j$$

5. Кожен ваговий коефіцієнт змінюється наступним чином:

$$W_{ij}(t+1) = w_{ij}(t) + \alpha x_j error_j$$

6. Виконуються етапи від другого до п'ятого кроку до того часу, поки помилка не досягне достатньо низького рівня.

Багатошарова нейронна мережа (БНМ) прямого поширення, відома також як багатошаровий персептрон, складається з формальних нейронів і має наступні параметри та характеристики: M - кількість шарів в мережі, N_μ - кількість нейронів у μ -му шарі, при цьому взаємозв'язки між нейронами в одному шарі відсутні. Виходи нейронів μ -го шару, де μ приймає значення від 1 до $M-1$, передаються на входи нейронів лише наступного $\mu+1$ -го шару. Зовнішній векторний сигнал x подається на входи нейронів тільки першого шару, а виходи нейронів останнього M -го шару формують вектор виходів мережі $y(M)$. Структура мережі показана на рис. 3.7.

Персептрон Розенблатта обмежується використанням бінарних виходів. Уїдроу та Хофф розширили алгоритм навчання персептрона, дозволяючи безперервні виходи і використовуючи сигмоїдальну функцію. Вони представили математичний доказ, що за певних умов мережа буде збігатися до будь-якої функції, яку вона може уявити. Їхня перша модель, відома як "АДАЛПН," включає один вихідний нейрон, а більш пізня модель, Мадаліною, розширює цю концепцію на випадок з багатьма вихідними нейронами.

Вирази, що розкривають процес навчання "АДАЛПН", мають велику схожість із тими, які застосовуються у персептрона. Важливі відмінності виявляються на четвертому етапі, де використовуються безперервні сигнали NET замість бінарних OUT.

У цьому випадку модифікований четвертий крок реалізується наступним чином: для кожного нейрона обчислюється помилка шляхом віднімання отриманого виходу від необхідного виходу.

Дослідження Кохонена у галузі самоорганізованих структур класифікують образи для задач розпізнавання, представлені у векторній формі, де кожен компонент вектора відповідає елементу образу.

Алгоритми Кохонена ґрунтуються на методі навчання без учителя. Після завершення навчання введення вхідного вектора з певного класу приводить до генерації збудженості в кожному вихідному нейроні; нейрон з найвищим рівнем збудженості представляє класифікацію. Оскільки навчання відбувається без заздалегідь визначеного цільового вектора, немає можливості передбачити, який нейрон буде відповідати вхідним векторам даного класу.

Однак це планування можна легко виконати, проведучи тестування мережі після завершення навчання.

Алгоритм розглядає набір з n ваг нейрона як вектор у n -вимірному просторі. Перед початком навчання кожен компонент цього вектора ваг ініціалізується як випадкова величина. Потім кожен вектор нормалізується до вектора одиничної довжини в просторі ваг. Це досягається шляхом ділення кожної випадкової ваги на квадратний корінь з суми квадратів компонент цього вагового вектора.

Всі вхідні вектори навчального набору також пройшли процедуру нормалізації, і мережа навчається за наступним алгоритмом:

1. Вектор X вводиться як вхід для мережі.

2. Визначаються відстані (в n -вимірному просторі) між X і ваговими

векторами кожного нейрона. В евклідовому просторі це відстань обчислюється за такою формулою

$$D_j = \sqrt{\sum_i (x_i - w_{ji})^2}$$

3. Нейрон, який має ваговий вектор, найближчий до X , оголошується переможцем. Цей ваговий вектор, званий w_c , стає основним в групі вагових векторів, які лежать в межах відстані D від

4. Група векторів ваг налаштовується згідно з наступним виразом:

$$W_j(\alpha + 1) = W_j(\alpha) + \alpha[X - W_j]$$

5. Проводяться етапи від 1 до 4 для кожного вхідного вектора.

У процесі навчання нейронної мережі поступово зменшуються значення D і α . На початку навчання коефіцієнт α встановлюється приблизно рівним 1 і поступово зменшується до 0. У той час як на початку навчання D може дорівнювати максимальній відстані між ваговими векторами, в кінці навчання воно стає настільки маленьким, що лише один нейрон навчається.

За наявною позицією, додаткове навчання може покращити точність класифікації. Відповідно до поради Кохонена, для досягнення високої статистичної точності, кількість навчальних циклів повинна бути щонайменше в 500 разів більше, ніж кількість вихідних нейронів.

Алгоритм навчання відстроює вагові вектори в області порушеного нейрона так, щоб вони стали більш подібними до вхідного вектора. Оскільки всі вектори нормалізуються до одиничної довжини, їх можна розглядати як точки на поверхні одиничної гіперсфери. У процесі навчання група сусідніх вагових точок рухається ближче до точки вхідного вектора.

Припускається, що вхідні вектори фактично групуються в класи відповідно до їх розташування в векторному просторі. Кожен конкретний клас асоціюється з відповідним нейроном, який переміщає свій ваговий вектор у напрямку центру класу та сприяє його порушенню при поданні будь-якого вектора даного класу на вхід. Після завершення процесу навчання класифікація здійснюється шляхом подання тестового вектора на вхід мережі, обчислення збудження для кожного нейрона та вибору нейрона з найвищим збудженням як індикатора правильної класифікації.

3.3. Модель розпізнавання образів

Френк Розенблат, нейрофізіолог, вніс значний внесок у розвиток нейрокібернетики, представивши модель розпізнавання образів, яку він назвав "Персептрон" (походить від латинського *percepto* - розумію, пізнаю). У

процесі розробки цієї моделі він враховував деякі уявлення про структуру мозку та зорового апарату.

Захоплений ідеєю відтворення функцій людського мозку, він застосовував просту модель біологічного нейрона та взаємодії між ними, намагаючись створити аналогічну систему.



Рисунок 3.8 – Персептронний нейрон

В ролі сприймального пристрою персептрона виступає фотоелектрична модель сітківки, що складається з численних фотосопротивлень (див. рис. 3.9). Кожен елемент у полі рецепторів може перебувати у двох станах - збудженим або не збудженим, в залежності від того, чи попадає контур фігури на відповідне фотосопротивлення. Кожен елемент має вихідний сигнал x_i (де $i = 1, 2, \dots, n$, n - кількість елементів), який приймає значення одиниці, якщо елемент збуджений, і нуля - у протилежному випадку. Наступною ланкою персептрона є асоціативні елементи, або А-елементи. У кожного А-елемента є кілька входів і один вихід. Під час підготовки персептрона до експерименту виходи рецепторів підключаються до входів А-елемента, і при цьому кожне підключення може мати або позитивний, або негативний знак.



Рисунок 3.9 – Персептронний нейрон з багатьма виходами

Вибір рецепторів, які підключаються до конкретного А-елемента, разом із визначенням знака підключення, здійснюється випадковим чином. Протягом експерименту зв'язок між рецепторами та А-елементами залишається незмінним. А-елементи виконують алгебраїчне підсумовування сигналів, які надходять на їх входи, і порівнюють отриману суму з універсальною величиною θ , яка залишається постійною для всіх А-елементів.

Якщо сума перевищує значення θ , А-елемент активується і виводить сигнал, рівний одиниці. У випадку, коли сума менша за θ , А-елемент залишається неактивованим, і його вихідний сигнал залишається нульовим. Таким чином, вихідний сигнал j-го А-елемента, де величина i_j приймає значення +1, якщо i-й рецептор підключений до входу j-го А-елемента зі знаком плюс; вона приймає значення -1, якщо рецептор підключений зі знаком мінус, і залишається нулем, якщо i-ий рецептор не підключений до j-го А-елемента ($j = 1, 2, \dots, m$, де m - кількість А-елементів).

Сигнали, що виходять з А-елементів, збільшуються за допомогою спеціальних пристроїв (підсилювачів) на змінні коефіцієнти λ_j . Кожен із цих коефіцієнтів може бути позитивним, негативним або дорівнювати нулю, і він може змінюватися незалежно від інших коефіцієнтів. Сигнали з підсилювачів сумуються, а отриманий загальний сигнал подається

на вхід реагуючого елемента або R-елемента. Якщо σ є позитивним або дорівнює нулю, R-елемент виводить одиницю; у випадку, якщо σ є негативним, він виводить нуль.

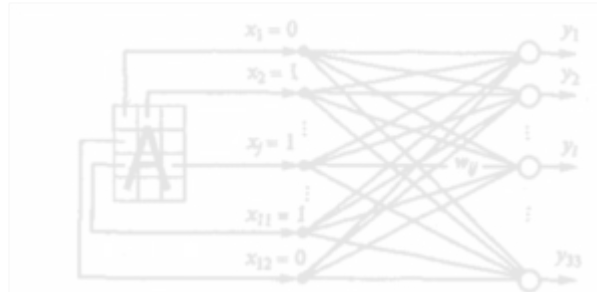


Рисунок 3.10 – Персептрон з декількома виходами

Припустимо, що на поле рецепторів подаються фігури, які відносяться до двох різних образів. Якщо вдасться налаштувати персептрон так, щоб він надійно видаючи сигнал 1 при появі на його вході фігур одного образу, то це свідчитиме про здатність персептрона вивчати розрізнення між двома образами.

Структура персептрона, яку було описано, дозволяє розділяти представлені об'єкти лише на два. Для розпізнавання більшого числа образів, таких як А, В і С, можна використовувати персептрон, який будується за схемою (рис. 3.10). Вихідний сигнал кожного А-елемента подається не лише на один, а на кілька (відповідно до кількості розрізняються образів) підсилювачів. Після множення на λ вихідні сигнали потрапляють на суматори Σ , кількість яких також відповідає числу розрізняються образів. Замість R-елемента використовується пристрій, який порівнює вихідні сигнали суматорів між собою. Представлений об'єкт відноситься до того образу, акумулятор якого має найвищий сигнал.

Для розпізнавання кількох образів можна використовувати персептрон із зміненою структурою. У такому персептроні А-елементи розподіляються на різні групи, кожна з яких пов'язана із своїм суматором і R-елементом.

Загальну суму вихідних сигналів R-елементів можна розглядати як двійковий код, що представляє номер образу, що дозволяє перцептрону класифікувати об'єкти на кілька класів. Наприклад, для класифікації на вісім класів досить трьох груп. У цьому випадку можливі такі вісім комбінацій вихідних сигналів трьох R-елементів: 000, 001, 010, 011, 100, 101, 110, 111. Поява кожної з цих комбінацій може бути розглянута як віднесення перцептроном пред'явленого йому зображення до одного з восьми образів.

Кожна група A-елементів, що пов'язана із своїм R-елементом, в структурі і функціональності є аналогічною перцептрону, який може розділяти об'єкти на два класи.

Навчання перцептрона включає послідовні такти. Кожного такту перцептрону подається об'єкт одного з образів. Залежно від реакції перцептрона на цю фігуру застосовуються певні правила для зміни коефіцієнтів λ_i . Виявляється можливим, за певну кількість тактів, привести перцептрон до стану, коли він достатньо впевнено розпізнає пред'явлені йому фігури. Існують два типи алгоритмів навчання перцептрона. У першому типі не враховується правильність відповідей перцептрона під час навчання, і зміна λ_i відбувається незалежно від того, чи "засвоїв" перцептрон фігуру, пред'явлену йому на даний такт, чи ні. У другому типі алгоритмів коефіцієнти λ_i змінюються з урахуванням правильності відповідей перцептрона.

Алгоритм першого типу реалізується таким чином. Початково встановлюється вимога до перцептрона, що після навчання він повинен видачі на виході сигнал 1 при представленні об'єктів образу A і сигнал 0 - при представленні об'єкту образу B. Далі перцептрону представляють об'єкти кожного з образів. На кожному такті перцептрон реагує на представлений об'єкт порушенням деяких A-елементів. Процес навчання полягає в збільшенні коефіцієнтів λ_i порушених A-елементів на певну величину (наприклад, на одиницю), якщо на цьому такті був представлений об'єкт образу A, і в зменшенні на цю ж величину, якщо був представлений об'єкт

образу В. Це призводить до підвищення правильності відповідей персептрона, оскільки збільшення λ_j порушених А-елементів призводить до збільшення сигналу на вході R-елемента, а їх зменшення - до зменшення сигналу. Відповідно до умови, прийнятої для персептрона, він буде видає правильні відповіді, якщо позитивні сигнали на вході R-елемента відповідатимуть образу А, а негативні - образу В.

Під час розробки персептрона Ф. Розенблат намагався емулювати деякі характеристики живого мозку. Робота персептрона або будь-якої програми, яка імітує процес розпізнавання, включає два режими: режим навчання і режим розпізнавання. Основна відмінність алгоритму персептрона від раніше розглянутих алгоритмів полягає в тому, що в процесі навчання не вимагається запам'ятовування пред'явлених об'єктів, а під час розпізнавання не потребує перебору всіх "відомих" фігур. У цьому контексті робота персептрона виявляє певну аналогію з роботою мозку, який формує уявлення про образ, не запам'ятовуючи окремі його об'єкти, і здатен впізнавати нові об'єкти без порівняння їх з кожним раніше зустріченим.

Далі, структура персептрона подібна до деяких аспектів вищої нервової системи. Зокрема, рецептори персептрона можна порівняти з рецепторами зорового апарату, а А-елементи виявляють певну схожість з нейронами. Відомо, що нейрони здатні порушуватися, якщо інтенсивність сигналу, який вони отримують від пов'язаних з ними рецепторів (або інших нейронів), перевищує певний порог. Схожість персептрона з головним мозком проявляється у випадковому характері зв'язків "рецептор - А-елемент", що визнається аналогічними властивостями структури мозку. Можливо, що взаємозв'язки між нейронами мозку у більшості випадків також мають випадковий характер, тобто вони випадково змінюються у різних тварин одного біологічного виду. Якщо припустити протилежне – тобто допустити, що всі зв'язки між нейронами мозку точно фіксовані і однакові у всіх тварин даного виду, і що зміна цих зв'язків може спричинити різке

порушення роботи мозку, то можна припустити також, що інформацію про всі ці зв'язки повинно передаватися у спадок. Оскільки кількість нейронів у мозку обчислюється мільярдами, таке припущення призводить до фантастично великого обсягу генетичної інформації. Тим не менше, при розгляді персептрона стає очевидним, наскільки біологічно природнім є поняття компактності великих наборів даних, оскільки навчання розпізнаванню таких множин виявляється можливим при випадкових зв'язках між рецепторами і нейронами. Здатність штучних нейронних мереж до самонавчання є їхньою ключовою властивістю. Аналогічно до біологічних систем, які вони відтворюють, ці нейронні мережі вдосконалюють свою модель під час спроб досягти оптимальної поведінки.

Використовуючи критерій лінійної роздільності, можна визначити, чи може одношарова нейронна мережа втілити необхідну функцію. Навіть якщо відповідь позитивна, це може мати обмежену користь, якщо відсутні засоби для знаходження необхідних значень ваг і порогів. Для того, щоб мережа стала практично цінною, потрібен систематичний метод (алгоритм) для обчислення цих параметрів. Розенблатт розробив такий алгоритм у своєму процесі навчання персептрона, включаючи доведення того, що персептрон може навчатися будь-якій функції, яку він може реалізувати.

Навчання може відбуватися з вчителем або без його участі. В разі навчання з вчителем потрібна зовнішня особа, яка оцінює поведінку системи і керує її подальшими модифікаціями. У випадку навчання без вчителя мережа самостійно виконує необхідні зміни через процес самоорганізації.

Алгоритм навчання персептрона може бути впроваджений на цифровому комп'ютері або іншому електронному пристрої, і мережа, в певному відношенні, саморегулюється. У зв'язку з цим процедуру підгонки ваг часто називають "навчанням", і говорять, що мережа "вивчається".

3.4 Висновки до розділу 3

Були розглянуті методи класифікації графічних образів, а також алгоритмічні конструкції, які можна програмно реалізувати.

На підставі аналізу та опрацювання спеціалізованої літератури та ресурсів були виявлені принципи та знання з областей теорії нейробіології, нейроінформатики, штучного інтелекту, штучних нейронних мереж, персептрона та структури машинного розпізнавання графічних образів.

Основна здатність штучних нейронних мереж до навчання визначається їх важливою характеристикою. Подібно до біологічних систем, які їх імітують, ці нейронні мережі самостійно коригують свою модель, намагаючись досягти більш ефективної поведінки. В цьому контексті була розглянута модель розпізнавання образів під назвою "Персептрон" (від латинського *percepto* - розумію, пізнаю), при розробці якої враховувалися певні уявлення про структуру мозку та зорового апарату. Процес навчання персептрона включає послідовні етапи. На кожному етапі персептрон взаємодіє з об'єктом, що належить до одного з образів, і відповідно до реакції персептрона на пред'явлений об'єкт проводяться зміни в коефіцієнтах λ_j . Алгоритм навчання персептрона, який включає доказ того, що він здатен навчатися реалізовувати різноманітні завдання, представляє собою важливий аспект цього процесу.

Алгоритм навчання персептрона може бути реалізований на цифровому комп'ютері або іншому електронному пристрої. В даному випадку, розглядається штучна нейрона-матрична мережа (ІНМС), яка працює без вчителя, опрацьовуючи та модифікуючи еталонну базу образів за допомогою тестових зображень, які подаються на вхід. Ефективність методу вагових параметрів синаптичних масивів для навчання ІНС має практичну цінність і може бути застосована в різних галузях науки і практики.

РОЗДІЛ 4

РОЗРОБКА ЗАСТОСУВАННЯ ДЛЯ ДОПОМОГИ ЛЮДЯМ З ВАДАМИ ЗОРУ

4.1. Опис програмної реалізації

Python – це високорівнева об'єктно-орієнтована мова програмування, яка використовує інтерпретацію та має динамічну семантику. Завдяки використанню структур даних високого рівня, спільно з динамічною семантикою та динамічним зв'язуванням, ця мова стає привабливою для ефективної розробки програм та поєднання наявних компонентів. Крім того, Python підтримує модулі та пакети модулів, сприяючи модульності та можливості повторного використання коду.

Charm надає користувачеві інтегроване середовище розробки (Integrated Development Environment або IDE) з рядом корисних функцій, таких як підсвічування синтаксису та виявлення помилок, зручна навігація серед проєктів і коду програми, відображення структури файлів у проєкті, швидкий перехід між файлами, класами, методами та іншими.

PyCharm – це не просто текстовий редактор для програмування **наPython**, це справжній інтелектуальний помічник, який робить процес програмування зручним і продуктивним.

Як створити проєкт у pycharm. Для цього потрібно лише кілька кроків:

- запустити PyCharm і вибрати опцію “Create New Project” на стартовому екрані;
- вибрати тип проєкту, що відповідає мові програмування Python;
- вказуємо папку, де буде розташовуватися проєкт на комп'ютері;
- вказуємо інтерпретатор мови програмування, Python.

Після натискання кнопки “Create” проєкт запропонован. Вибераємо жіноче ім'я для свого голосового помічника: «Люсі». Налаштування середовища для коду. Спочатку імпортуємо/встановимо необхідні бібліотеки:

```
pyttsx3
дата та час
розпізнавання мови
Вікіпедія
веб-браузер
шлях до ОС
смптпіб
```

Визначення функції мови. Перше та головне для голосового помічника зі штучним інтелектом – це говорити. Щоб наш бот говорив, ми напишемо функцію `talk()`, яка приймає звук як вхідні дані і вимовляє його як вихідні дані.

```
def talk(audio): прохід #поки, умови напишемо пізніше.
```

Тепер нам потрібний звук, щоб забезпечити правильний зв'язок між користувачем та помічником. Отже, ми збираємося встановити модуль з ім'ям `pyttsx3`.

Це бібліотека Python, яка допоможе перетворити текстовий формат на мовний формат.

Крім того, він працює в автономному режимі та сумісний як з Python 2, так і з Python 3.

```
Монтаж:
pip встановити pyttsx3
```

Після успішного встановлення `pyttsx3` імпортуємо цей модуль у свою програму.

```
імпортувати pyttsx3
двигун = pyttsx3.init('sapi5')
voices = engine.getProperty('voices') #отримує докладну
інформацію про поточний голос
engine.setProperty('voice', voice[1].id) # 0-чоловічий
голос, 1-жіночий голос
```

Microsoft Speech API (SAPI5) – це технологія розпізнавання та синтезу голосу, що надається Microsoft.

Зазвичай це допомагає у синтезі та розпізнаванні голосу.

Ідентифікатор голосу `VoiceId` допомагає нам вибирати різні ГОЛОСИ.

```
voice[0].id = Чоловічий голос
voice[1].id = Жіночий голос
```

Написання своєї функції `talk()`:

```
безперечно кажу (audio):
engine.say (audio)
engine.runAndWait() #Без цієї команди не буде чути.
```

Створюємо основну функцію. Створимо функцію `main()` і всередині цієї функції `main()` визначимо налаштовану мовну функцію.

```
якщо __name__=="__main__":
    talk ('Здрастуйте, , я «Люсі», ваш помічник зі штучного
інтелекту. Скажіть, будь ласка, чим я можу вам допомогти')
```

Все, що напишемо всередині функції `talk()`, буде повністю перетворено на мову. Завдяки цьому голосовий помічник має власний голос, і він готовий говорити.

Кодування функції `ishme()`. Напишемо функцію `ishme()`, завдяки якій голосовий помічник бажатиме або вітатиме нас залежно від часу на комп'ютері.

Щоб надати помічнику поточний час, потрібно імпортувати модуль `datetime`. Імпортуємо цей модуль у свою програму, використовуючи таку команду:

```
імпорт дати та часу

Тепер почнемо кодувати функцію ishme():
defishme():
    година = int(datetime.datetime.now(). hour)
```

Тут зберегли ціле значення поточної години або часу у змінній з ім'ям години. Тепер використовуватимемо це значення години всередині циклу if-else.

```
defishMe():
    година = int(datetime.datetime.now().hour)
    if hour >= 0 і час < 12:
        говорити("Доброго ранку!")
    elif годину >= 12 і годину < 18:
        говорити("Доброго дня!")

    else:
        talk("Добрий вечір!")
```

talk ('Доброго дня, я «Люсі», ваш помічник зі штучного інтелекту. Скажіть, будь ласка, чим я можу вам допомогти')

Визначення функції take Command(). Наступна важливість для голосового помічника: він повинен мати можливість приймати команди за допомогою мікрофона системи. Почнемо кодувати функцію take Command().

За допомогою функції takeCommand(), наш голосовий помічник AI також зможе повертати рядковий висновок, приймаючи введення від користувача через мікрофон.

Але перш ніж визначити функцію take Command (), потрібно встановити модуль під назвою SpeechRecognition, і команда буде виглядати так: `pip встановити мову Recognition`

Після успішного встановлення цього модуля імпортуємо цей модуль у програму, написавши оператор імпорту. Імпортуємо розпізнавання мови як SR. Кодуємо функцію takeCommand() :

```
def takeCommand():
    #Вона приймає вхідні дані мікрофона від користувача і
    повертає рядковий висновок r = sr.Recounser () з sr.Microphone
```

60

```
() як джерело: print ("Прослуховування...") r.pause_threshold =
1 audio = r .listen(джерело) )
```

Додаємо до програми блок «спробуй і виключай», щоб ефективно обробляти помилки. Використовуємо:

```
print ("Розпізнавання...")
query      =      r.recounse_google(audio,      Language='en-in')
#Використання Google для розпізнавання голосу.
print(f"Користувач сказав: {query}\n") #Запит користувача
буде надрукований.
```

крім Exception as :

```
# print(e) використовуємо лише тоді, коли хочемо надрукувати
помилку!
```

```
print("Скажи це ще раз, будь ласка...") #Скажи, що знову
буде надруковано у разі неправильного голосового
відповіді "None" #Немає рядок буде повернуто
return query
```

Тепер визначаємо завдання та отримуємо бажані результати.

Завдання 1. Щоб знайти щось у Вікіпедії, виконувати пошук, необхідно встановити та імпортувати модуль Вікіпедії у програму.

Команда для встановлення модуля Вікіпедії виглядає так:

```
pip install wikipedia
```

Після успішного встановлення модуля Вікіпедії імпортуємо модуль Вікіпедії в програму, написавши оператор імпорту.

```
if __name__ == "__main__":
    ishMe()
    while True:
        query      =      takeCommand().lower()      #Перетворення      запиту
користувача в нижній регістр
        # Логіка виконання завдань на основі запиту
        if 'wikipedia' у запиті: #якщо у запиті знайдена wikipedia,
цей блок буде виконаний
        talk('Searching Wikipedia...')
```

```
query = query.replace("wikipedia", "" )
результати = wikipedia.summary (запит, пропозиції = 5)
говорити ("Згідно з Вікіпедією")
print (результати)
говорити (результати)
```

У наведеному вище коді ми використовуємо оператор `if`, щоб перевірити, знаходиться Вікіпедія у пошуковому запиті користувача чи ні. Якщо за пошуковим запитом користувача знайдено Вікіпедію, то п'ять пропозицій (які можна змінити, змінивши кількість пропозицій з 5 на будь-яке бажане число) з короткого змісту сторінки Вікіпедії буде перетворено на мову за допомогою команди `talk.` функція.

Завдання 2. Щоб відкрити YouTube у веб-браузері:

Щоб відкрити будь-який веб-сайт, спочатку потрібно імпортувати модуль під назвою `webbrowser`.

Це вбудований модуль і нам не потрібно встановлювати його за допомогою оператора `pip`, ми можемо безпосередньо імпортувати його в нашу програму, написавши оператор імпорту.

Код:

```
elif 'відкрити YouTube' у запиті:
webbrowser.open("youtube.com")
```

Тут ми використовуємо оператор `elif`, щоб перевірити, чи увімкнено YouTube у запит користувача чи ні. Припустимо, користувач дає команду "Будь ласка, відкрийте YouTube".

Отже, у запиті користувача буде відкрито YouTube, і умова `elif` буде істинною, а значить, код виконується.

Завдання 3. Щоб відкрити пошукову систему Google у веб-браузері:

```
elif 'відкрити Google' у запиті:
webbrowser.open("google.com")
```

Ми відкриваємо Google у веб-браузері, застосовуючи ту ж логіку, яку ми використовували під час відкриття YouTube.

Завдання 4. Для відтворення музики:

Для відтворення музики нам потрібно імпортувати модуль під назвою `os`. Імпортуємо цей модуль безпосередньо за допомогою оператора імпорту.

```
elif 'грати музику' у запиті:
    music_dir = 'music_dir_of_the_user'
    song = os.listdir(music_dir)
    print (songs)
    os.startfile(os.path.join(music_dir, song[0]))
```

У наведеному вище коді спочатку відкриваємо музичний каталог користувача, а потім виводимо список всіх пісень, присутніх в каталозі, за допомогою модуля `os`.

За допомогою `os.startfile` ми можемо відтворити будь-яку пісню на наш вибір. Ми можемо також відтворити випадкову пісню за допомогою випадкового модуля. Щоразу, коли ми даємо команду на відтворення музики, голосовий помічник AI відтворюватиме будь-яку випадкову пісню з каталогу пісень.

Завдання 5: Дізнатися поточний час:

```
elif 'час' у запиті:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    talk(f"Жанно, скажи час {strTime}")
```

У наведеному вище коді ми використовуємо функцію `datetime ()` і зберігаємо поточний час системи змінної з ім'ям `strTime`.

Після збереження часу в `strTime` ми передаємо цю змінну як аргумент у функцію `talk`, і, отже, рядок часу буде перетворено на мову.

Завдання 6. Щоб відкрити Stackoverflow:

```
elif 'переповнення відкритого стека' у запиті:
    webbrowser.open('stackoverflow.com')
```

Ми використовували ту саму термінологію, що й для пошукової системи Google та YouTube.

Завдання 7. Щоб надіслати електронний лист, нам потрібно імпортувати модуль `smtplib`.

Простий протокол передачі пошти (SMTP) – це протокол, який дозволяє нам надсилати електронну пошту та маршрутизувати електронну пошту між різними поштовими серверами.

Метод під назвою `sendmail` є у модулі SMTP. Цей метод дозволяє нам надсилати електронні листи.

Приймаються 3 параметри:

Відправник: адреса електронної пошти відправника.

Отримувач: електронна адреса одержувача.

Повідомлення: рядкове повідомлення, яке потрібно надіслати одному або декільком одержувачам.

Тепер ми створимо функцію `sendEmail()`, яка допоможе нам надсилати електронні листи одному чи кільком одержувачам.

```
def sendEmail(to, content):
    Server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('youremail@gmail.com', 'ваш-пароль')
    server.sendmail('youremail@gmail.com', to, content)
    server.close()
```

Увімкнемо функцію менш безпечних програм у своєму обліковому записі Gmail. В іншому випадку функція `sendEmail` не працюватиме належним чином.

Виклик функції `sendEmail()` всередині функції `main()`:

```
elif 'електронний лист на ім'я одержувача' у запиті:
    пробуємо:
    промовляю ("Що мені сказати?")
    content = takeCommand()
    to = "ідентифікатор електронної пошти одержувача"
    sendEmail(to, content)
    talk("Електронний лист був відправлений!")
```



```
винятки у вигляді e:  
print(e)  
talk ("Вибаачте. Я не можу відправити цей лист")
```

Для обробки будь-яких можливих помилок, які можуть виникнути при надсиланні електронних листів використовуємо блок `try`.

4.2. Висновки до розділу 4

При створенні проєкту розглянуто основні функції та можливості PyCharm, що роблять його потужним інструментом для програмування на Python. Він надає зручні засоби для керування проєктом, налагодження коду, рефакторингу, інтеграції з системами контролю версій та інше.

Технічно розроблене ПЗ голосового помічника «Люсі» працює на основі вбудованого штучного інтелекту. Головна мета голосового помічника — зменшити людські зусилля та виконувати завдання з тією самою ефективністю, як і людина (або навіть краще).

ВИСНОВКИ

У рамках магістерської кваліфікаційної роботи було виконано загальний аналіз сучасного стану розглянутої проблеми, розглянуто можливості сучасних комп'ютерних засобів для слабозорих людей, приділено увагу особливостям класифікації навігаційних систем, пристроїв, методів розпізнавання та рецепторну структуру сприйняття інформації за допомогою впровадження додаткових скриптів Python.

Для досягнення мети магістерського дослідження та виконання поставлених завдань використано сучасні технології при розробці голосового помічника «Люсі»:

- pytsx3: бібліотека перетворення тексту в мову, яка включає різні механізми TTS, такі як sapi5, eSpeak.
- SpeechRecognition: бібліотека для розпізнавання мовлення з підтримкою кількох движків і API, серед яких CMU Sphinx, Google Speech Recognition, Google Cloud Speech API, Microsoft Bing Voice Recognition, Wit.ai і т.д.
- webbrowser: модуль, що надає високорівневий інтерфейс, що дозволяє відображати користувачам веб-документи.
- os.path: модуль шляху, що підходить для взаємодії та надання портативного способу використання функцій, що залежать від операційної системи.
- smtplib: модуль, що визначає об'єкт сеансу клієнта SMTP, який можна використовувати для надсилання пошти на будь-який комп'ютер з демоном прослуховувача SMTP або ESMTP.

В результаті розроблено ПЗ голосовий помічник «Люсі» для спілкування слабозорих людей, який голосом і введенням запитів з клавіатури може виконувати прикладні завдання: надсилати та отримувати електронні листи; відтворювати offline музику; виконувати пошук у Вікіпедії;

відкривати такі веб-сайти, як Google, YouTube, Stackoverflow, freecodecamp тощо, у веб-браузері.

Представлено методику розробки проєкту системи комп'ютерної допомоги для людей з вадами зору голосового помічника «Люсі» у середовищі PyCharm засобами Python. Протестовано та досліджено функціонал за яким буде працювати розроблена система. Код системи побудовано таким чином, щоб в подальшому без труднощів можна вносити необхідні зміни. Наприклад такі, як модифікації структури моделі та корекції системи.

Роботу може бути рекомендовано в якості методичного матеріалу при навчанні здобувачів вищої освіти методам програмування. Дана розробка та її впровадження актуально, оскільки стосується інформаційних технологій прогресивних напрямів, а також якості життя величезної кількості людей з вадами зору в Україні.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України «Про вищу освіту» зі змінами 2021 рік №1556-VII від 07.01.2014, редакція від 07.16.2021.
2. Державний стандарт України ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. К.: Держстандарт України, 2017. 31 с.
3. Основні вимоги до дисертацій та авторефератів дисертацій / Бюлетень ВАК України. 2002. Спецвипуск. С. 12 - 17.
4. Могильний Г. А., Сквірський В. Д., Козуб Г. О. Методичні рекомендації до виконання кваліфікаційних робіт за напрямом 0403 „Системні науки та кібернетика” зі спеціальності „Інформатика” (для всіх форм навчання). Старобільськ : ДЗ „ЛНУ імені Тараса Шевченка”, 2018. 50 с
5. ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання», Чинний від 04 березня 2016 р. № 65. К: Держстандарт України, 2016. 20 с.
6. Про затвердження Вимог до оформлення дисертації: Закон України за станом на 12.01.2017 № 40 / URL: <http://zakon2.rada.gov.ua/laws/show/z0155-17>. дата звернення: 12.08.2021.
7. Козуб Г. О., Козуб Ю. Г. Методичні рекомендації до виконання кваліфікаційної роботи за спеціальністю 122 „Комп’ютерні науки” першого рівня вищої освіти. – Старобільськ : ДЗ „ЛНУ імені Тараса Шевченка”, 2021. 99 с.
8. Бертон М.Дж., Рамке Дж., Маркес А.П., Борн Р.Р., Джонс І. і ін. Комісія Lancet Global Health за глобальним здоров’ям: бачення після 2020 року. Lancet Glob Health. 2021 рік.
9. Лин Б.-С., Ли К.-К. и Чيانг П.-Ю. « Простая система навигации на основе смартфона для людей с нарушениями зрения », Sensors , Vol. 17, нет. 6, стр. 1371 , 2017 . URL: <https://doi.org/10.3390/s17061371> .

10. Хенкар С., Алсулайман Х., Ісмаїл С., Файрак А., Джаррая С.К. і Бен-Абдалла Х., «ENVISION: Допоміжна навігація для слабоворих користувачів смартфонів», *Procedia. обчисл. наук.*, Том. 100, стор. 128-135, 2016. URL : <https://doi.org/10.1016/j.procs>.
11. Schwarze T., Lauer M., Schwaab M., Romanovas M., Böhm S., і Jürgensohn T., «Допомога для пересування на основі камери для людей з порушеннями зору», *KI-Künstliche Intelligenz*, Vol. 30, ні. 1, стор. 29–36, 2016 р. URL: <https://doi.org/10.1007/s13218-015-0407-7>.
12. Чаккур К., Бадр Г., « Нова внутрішня навігаційна система для слабоворих і сліпих », Міжнародна конференція з прикладних досліджень у галузі комп'ютерних наук та інженерії (ICAR), г., IEEE, 2015, стор. 1-5.
13. Bai J., Lian S., Liu Z., Wang K., and Liu D., «Носимо навігаційний пристрій для сліпих людей, орієнтований на віртуальну сліпу дорогу», *IEEE Trans. споживати. Електрон.*, Том. 64, ні. 1, стор. 136 – 143, 2018. URL: <https://doi.org/10.1109/TCE.2018.2812498>.
14. Бай Дж., Лю Д., Су Г. та Фу З., «Хмарна і заснована на зорі навігаційна система, що використовується для сліпих», у матеріалах Міжнародної конференції 17 верес. 2017 р. з технологій штучного інтелекту, автоматизації та управління.
15. Li B., Munoz JP, Rong X., Chen Q., Xiao J., Tian Y., Arditi A. та Yousuf M., «Мобільна допоміжна навігація в приміщенні на основі зору для сліпих», *IEEE Trans. Моб. обчисл.*, Том. 18, ні. 3, стор. 702 – 714, 2019. URL: Дої: <https://doi.org/10.1109/TMC.2018.2842751>.
16. Мардер-Еппштейн Е., «Проект танго», *ACM SIGGRAPH 2016 Real-Time Live!*, SIG-GRAPH '16, Нью-Йорк, штат Нью-Йорк, США, Асоціація обчислювальної техніки, 2016, стор 25.
17. Xiao J., Joseph SL, Zhang X., Li B., Li X., and Zhang J., «Допоміжна система навігації для людей з вадами зору», *IEEE Trans. Людина-мах. Сист.*,

Том. 45, ні. 5, стор. 635-640 , 2015 . URL: <https://doi.org/10.1109/THMS.2014.2382570>.

18. Lee YH., Medioni G., «Носима навігаційна система на основі Rgb-d камери для людей з вадами зору», Comput. Вис. Зображення. Розумію. , Том. 149, стор. 3–20 , 2016 р. URL: <https://doi.org/10.1016/j.cviu.2016.03.019>.

19. Джозеф С.Л., Сяо Дж., Чжан С., Чавда Б., Наранг К., Раджпут Н.-т., Мехта С. та Венката Субраманіам Л., «Усвідомлювати світ: до використання соціальних мереж для допомоги сліпим у навігації», IEEE Trans. Людина-мах. Сист. , Том. 45, ні. 3, стор 399 – 405 , 2015 . URL: <https://doi.org/10.1109/THMS.2014.2382582>.

20. Алі А., Алі М.А., « Сліпа навігаційна система для людей з вадами зору з використанням віконних засобів на камері Microsoft Kinect », Міжнародна конференція з досягнень у галузі біомедичної інженерії, ICABME, 2017.

21. Тон К., Омар А., Седенко В., Тран В. Х., Афтаб А., Перла Ф., Бернштейн М. Дж., Янг Ю. «Лідар допомагає в просторовому сприйнятті для слабозорих і аналіз продуктивності», IEEE Trans. Нейронна система. Реабіліт. англ., Том. 26, 9, стор. 1727-1734 , 2018 . URL: <https://doi.org/10.1109/TNSRE.2018.2859800>.

22. Наір В., Цангурі К., Сяо Б., Олмшенк Г., Чжу З. і Сейпл В. Гібридна система позиціонування в приміщенні для сліпих і слабозорих з використанням Blue-tooth і Google Tango // J.Technol. Інвалід. , Том. 6, стор. 62 – 82 , 2018.

23. Чераги С. А., Намбудірі В., Уокер Л., «Guidebeacon: пошук шляху в приміщенні на основі маяків для сліпих, слабозорих і дезорієнтованих», Міжнародна конференція IEEE з всеосяжних обчислень і комунікацій (PerCom), 2017 р.

24. Каллара С.Б., Радж М., Раджу Р., Дж. Метью Н.-Х., Падмапрабха В. Р. та Див'я Д. С., «Індіря — інтелектуальна система навігації для людей з

вадами зору», Міжнародна конференція з інноваційних обчислень та інформатики (ICICI) 2017 р., IEEE, 2017 р.

25. Віра Д., Марсилло Д., Перейра А., «Сліпий гід: рішення для допомоги сліпим у будь-який час і в будь-якому місці», Всесвітня конференція з інформаційних систем та технологій, Springer, 2017, сс. 353–363.

26. Сен А., Сен К. та Дас Дж., «Ультразвукова сліпа палиця для повністю сліпих людей, що дозволяє уникати будь-яких перешкод», у 2018 р. IEEE SENSORS, IEEE, 2018, стор. 1–4.

27. Нада А. А., Фахр М. А., Седдік А. Ф., «Паличка на основі допоміжного інфрачервоного датчика для сліпих», науково-інформаційна конференція (SAI) 2015 р., IEEE, 2015 р., стор. 1149.

28. Джафрі Р., Кампос Р.Л., Алі С.А. та Арабнія Х.Р., «Виявлення перешкод на основі даних візуальних та інфрачервоних датчиків для слабозорих з використанням комплекту для розробки планшета танго проекту Google та механізму Unity», IEEE. Доступ. , Том. 6, стор. 443 – 454, 2017. URL: <https://doi.org/10.1109/ACCESS.2017>.

29. Марцек П., Кос А., «Точна навігаційна система з низьким енергоспоживанням для сліпих з інфрачервоними датчиками », в 2019 р. MIXDES-26-а Міжнародна конференція «Змішаний дизайн інтегральних схем та систем».

30. Дюкас Дж., Брок А. М., Жуфре К., «Доступні інтерактивні карти для людей з порушеннями зору», Springer, 2018, с. 537–584.

31. Альбуї-Перруа Ж., Лавіоль Ж., Брайант К., Брок А. М. «На шляху до мультисенсорної карти доповненої реальності для сліпих і людей з вадами зору: підхід до спільного проектування», в матеріалах конференції СНІ 2018 р.

32. Гетцельманн Т., Вінклер К., «Розумні тактильні карти: заснований на смартфонах підхід до підтримки сліпих людей у вивченні тактильних

карт», у матеріалах 8-ї Міжнародної конференції ACM з всеосяжних технологій, пов'язаних з допоміжними середовищами, 2015.

33. Лю К., Лі Р., Ху Х., Гу Д.-б., «Побудова семантичних карт для сліпих людей для навігації вдома», з інформатики та електронної інженерії (CEECS), IEEE, 2016, С. 12–17.

34. Гетцельманн Т., « Lucentmaps: 3D-друковані аудіовізуальні тактильні карти для сліпих і людей з вадами зору », у матеріалах 18-ї Міжнародної конференції ACM Sigaccess з комп'ютерів та доступності, 2016 р., стор. 81–90.

35. Глісон К., Гуо А., Лапут Г., Кітані К., Бігхем Дж. П., «Візмапап: доступна візуальна інформація за допомогою реконструкції картки з використанням краудсорсингу», у матеріалах 18-ї Міжнародної конференції ACM SIGACCESS з комп'ютерів та доступності, 2016 р.

36. Vision Workshops C., ICCVW 2017, 2018-Janua, 2018, стор 1480 – 1489.

37. Сато Д., Найто У. О. К., Такагі Х.-р., Кітані К., Асакава К., «Navsog3», у матеріалах 19-ї Міжнародної конференції ACM SIGACCESS з комп'ютерів та доступності — АКТИВИ '17, 2017 р., С. 270–279.

38. Лок Дж., Челняк Г., Беллотто Н.. Портативна система навігації з адаптивним мультимодальним інтерфейсом для сліпих. Весняний симпозіум AAAI - Технічний звіт, SS-17-01 -: 395–400, 2017 р.

39. Козуб Г.О., Чебаненко І.О. СИСТЕМА КОМП'ЮТЕРНОЇ ДОПОМОГИ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ. *Матеріали II Міжнародної науково-практичної конференції «SCIENCE IN MOTION: CLASSIC AND MODERN TOOLS AND METHODS IN SCIENTIFIC INVESTIGATIONS»* (2024, Вінниця, Україна - Відень, Австрія). С.178-180. <https://doi.org/10.36074/grail-of-science.19.01.2024.031>.

ДОДАТКИ

Додаток А



Додаток Б

Лістинг PyCharm коду голосового помічника «Люсі»

```
import pyttsx3 # перетворює текст на мовлення
import datetime # необхідний для вирішення будь-яких запитів щодо
дати та часу
import speech_recognition as sr # необхідний для повернення рядкового виводу
шляхом отримання мікрофонного введення від користувачаimport wikipedia
import wikipedia # необхідний для вирішення будь-яких запитів щодо вікіпедії
import webbrowser # необхідний, щоб відкрити запропоновану програму
у веб-браузері
import os.path # необхідний для отримання вмісту з указаної
папки/каталогу

import smtplib # необхідний для роботи із запитом щодо електронної
ПОШТИ

engine = pyttsx3.init('sapi5') # sapi5 — це API та технологія для розпізнавання та
синтезу голосу, надані Microsoft
voices = engine.getProperty('voices') # надає нам детальну інформацію про поточні
ГОЛОСИ
engine.setProperty('voice',voices[0].id) # 0-жіночий голос

def speak(audio): # функція для помічника говорити
    engine.say(audio)
    engine.runAndWait() # без цієї команди помічника ми не чуємо

def wishme(): # функція для побажання користувачеві відповідно до
денного часу
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak('Доброе утро')

    elif hour>=12 and hour<18:
        speak('Добрый день')

    else:
        speak('Доброго вечора')
```

```

speak('Привіт Илья, Я голосовий помічник Люсі, Чим я можу теобі допомогти ?')

def takecommand():
    # функція для помічника говоритиuser
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Слушаю...')
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        # обробка помилок
        print('Распознаю...')
        query = r.recognize_google(audio, language = 'ru-RU') # використання Google для
        розпізнавання голосу      print(f'User said: {query}\n')

    except Exception as e :
        print('Say that again please...')      # використання google для розпізнавання голосу
        improper voice
        return 'None'
        return query

def sendemail(to,content):
    # функція для надсилання електронної пошти
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.ehlo()
    server.starttls()
    server.login('senders_email@gmail.com','senders_password')
    server.sendmail('senders_email@gmail.com',to,content)
    server.close()

if __name__ == '__main__':
    # Контроль виконання

    wishme ()
    while True:
        query = takecommand().lower()

        # Вся логіка для виконання завдань на основі запиту користувача

        if 'вікіпедія' in query :
            speak('Пошук в вікіпедії...')
            webbrowser.open('https://uk.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE
%D0%B2%D0%BD%D0%B0_%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD
%D0%BA%D0%B0')
            query = query.replace('wikipedia',"")
            results = wikipedia.summary(query, sentences = 5)

```

```

print(results)
speak(results)

elif 'открой youtube' in query :
    webbrowser.open('youtube.com')

elif 'открой радио' in query:
    webbrowser.open('https://play.tavr.media/hitfm/')
elif 'открой аудиокнигу' in query:
    webbrowser.open('https://www.youtube.com/watch?v=nyXy_BG2i0M&t=200s')
elif 'открой бунина' in query:
    webbrowser.open('https://www.youtube.com/watch?v=-kEAs0Y2v5s')

elif 'открой google' in query :
    webbrowser.open('https://www.google.com.ua/?hl=uk')

elif 'двумя зайцами' in query:
    webbrowser.open('https://www.youtube.com/watch?v=0HY8-_kYfJc')

elif 'полный список' in query :
    speak('Озвучиваю полный список Илья')
    speak('Открой ютуб')
    speak('открой гугл')
    speak('открой википедию')
    speak('открой аудиокнигу')
    speak('открой Бунина')
    speak('открой радио')

elif 'скажи время' in query :
    strtime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f'Илья сейчас время {strtime}')

elif 'фильм За двумя зайцами' in query :
    webbrowser.open('https://www.youtube.com/watch?v=0HY8-_kYfJc')

elif 'open free code camp' in query :
    webbrowser.open('freecodecamp.org')

elif 'pycharm' in query :
    codepath = 'pycharm_directory_of_your_computer'
    os.startfile(codepath)

elif 'email' in query :
    try:
        speak('what should i write in the email?')

```

```
content = takecommand()
to = 'iliachebanenlo1706@gmail.com'
sendemail(to, content)
speak('email has been sent')
except Exception as e:
    print(e)
    speak('Sorry, I am not able to send this email')

elif 'выход' in query:
    speak('окей, босс, пожалуйста, позови меня, когда я понадобится')
    quit()
```

Схожість

Джерела з Інтернету

938

1	http://dspace.wunu.edu.ua/bitstream/316497/17483/1/OlikhVY.PDF	63 джерела	2.44%
2	http://dspace.luguniv.edu.ua/jspui/bitstream/123456789/8567/1/2021-mttod.PDF	219 джерел	1.18%
3	https://github.com/Akshat-unt/Coddie-Desktop-A.I/blob/master/coodie.py	19 джерел	1.05%
4	https://er.nau.edu.ua/bitstream/NAU/54490/1/%d0%a4%d0%9a%d0%9a%d0%9f%d0%86_2021_122_%d0%97%d0%b0%d0%b...		0.97%
5	https://medium.com/analytics-vidhya/a-guide-to-your-own-a-i-voice-assistant-using-python-17f79c94704	2 джерела	0.94%
6	https://onecompiler.com/cpp/3xz4gbw63	15 джерел	0.88%
7	https://stackoverflow.com/questions/59325110/how-to-use-multithreading-with-tkinter-in-this-program	2 джерела	0.83%
8	https://stackoverflow.com/questions/58552647/i-am-creating-the-exe-file-of-my-python-file-but-it-always-crash-but-it-is-work		0.82%
9	https://github.com/awesomejuno/A.I.S.H.A.-Artificial-Intelligence-Simulated-Humanoid-Assistant/blob/master/main.py	3 джерела	0.82%
10	https://paste.myst.rs/clone/qtuzbi51	3 джерела	0.81%
11	https://gist.github.com/adiosvirogas0607/e25752373e4ca50bb5a6a19ad9db543a	5 джерел	0.74%
12	http://victoria.lviv.ua/html/wosserman/dodatok_b.htm	3 джерела	0.65%
13	http://technicalnews.net.ua/library/2011/27.pdf		0.42%
14	http://repositsc.nuczu.edu.ua/bitstream/123456789/12445/1/%d0%97%d0%b1%d1%96%d1%80%d0%bd%d0%b8%d0%bd...	123 джерела	0.36%
15	https://cloud.tencent.com/developer/ask/sof/282260	2 джерела	0.33%
16	http://dspace.luguniv.edu.ua/xmlui/bitstream/handle/123456789/2750/10_Kozoob%20.pdf?isAllowed=y&sequence=1	22 джерела	0.32%
17	https://spppc.com.ua/index.php/journal/article/download/1580/1556	35 джерел	0.25%
18	https://istoriya.in.ua/zavdannya-shtuchnogo-intelektu-7-test-po-temi-istoriya-rozvitk.html	44 джерела	0.23%
19	https://er.chdtu.edu.ua/bitstream/ChSTU/3477/1/kmoss2019.pdf	7 джерел	0.23%
20	https://ukrlogos.in.ua/ua_conference_19_01_2024_grail.php		0.2%

[illegible]