Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Аналіз впливу SSI-підходу на точність розпізнавання голосових команд

(тема)

Виконав:

студент II курсу, групи СПм-21-1

Терещенко О.В.
(прізвище, ініціали)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: к.т.н., доц. Барковська О.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Терещенку Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз впливу SSI-підходу на точність розпізнавання голосових команд

затверджена наказом по університету від "7" листопада 2022 р. № 1454Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи

1) Розпізнавання голосу;
2) Розпізнавання обличчя на зображенні;
3) GPGPU: CUDA.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз проблеми та огляд існуючих рішень;
2) вибір технології розробки;
3) розробка алгоритмів розпізнавання рис обличчя у відео;
4) аналіз нейронних мереж;
5) застосування технологій прискорення обробки програми;
6) налагодження та тестування;
7) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) слайд презентація – 12слайдів.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	07.11.22 - 10.11.22	
2	Вибір технології розробки	11.11.22 - 13.11.22	
3	Розробка алгоритмів роботи з голосом	15.11.22 - 20.11.22	
4	Розробка алгоритмів роботи з рисами обличчя	21.11.22 - 25.10.22	
5	Прискорення програми	26.11.22 - 30.11.22	
6	Налагодження та тестування програми	01.12.22 - 03.12.22	
7	Оформлення матеріалів кваліфікаційної роботи	04.12.22 - 06.12.22	
8	Подання кваліфікаційної роботи керівникові та	07.12.22 - 09.12.22	
	її попередній захист		
9	Подання кваліфікаційної роботи на рецензу-	10.12.22 - 12.12.22	
	вання		

Дата видачі завдання 7 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ к.т.н., доц. Барковська О.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 16 рис., 6 табл, 2 дод., 37 джерел.

АВТОМАТИЧНЕ ЧИТАННЯ ПО ГУБАХ, ВІДЕО, ІНТЕРФЕЙС БЕЗМОВНОГО ДОСТУПУ, РОЗПІЗНАВАННЯ МОВИ, МАШИННЕ НАВЧАННЯ, GPGPU.

Метою кваліфікаційної роботи є дослідження впливу використання інтерфейсу безмовного доступу на точність розпізнавання голосових команд у різних звукових оточеннях. Для досягнення поставленої мети у роботі запропоновано узагальнену модель розпізнавання голосових команд, що працює у двох режимах – з однотипними даними на вході (лише звуковий ряд) та двохтипними даними на вході (голосовий ряд та зображення губ). Працездатність запропонованої моделі протестовано та підтверджено завдяки розробленому програмному забезпеченню для розпізнавання голосових команд людини із розширеним функціоналом за рахунок додаткового модуля оброки зображення губ людини для визначення початкової фази звукового ряду, асоційованого із початком мовлення.

У роботі використано технології паралельних обчислень для навчання нейронних мереж, проведено їх аналіз та визначено найкращий спосіб для реалізації поставленої задачі. Визначено основні переваги та недоліки використання інтерфейсу безмовного доступу при розпізнаванні людської мови.

ABSTRACT

Master’s thesis: 78 pages, 16 figures, 6 tables, 2 appendices, 37 sources.

AUTOMATED LIP READING, VIDEO, SILENT SPEECH INTERFACE, SPEECH RECOGNITION, MACHINE LEARNING, GPGPU.

The major goal of this thesis is to research the impact of using a silent access interface on the accuracy of voice command recognition in different sound environments. To achieve the goal, the work proposed a generalized model of voice command recognition that works in two modes - with one-type input data (only the sound series) and two-type input data (voice series and image of lips). The functionality of the proposed model was tested and confirmed thanks to the developed software for recognizing human voice commands with extended functionality due to the additional module of the image of human lips to determine the initial phase of the sound series associated with the beginning of speech.

The work uses parallel computing technologies for neural networks, their analysis was carried out, and the best method for the implementation of the given task was determined. Software was developed based on the required speech recognition functions. The main advantages and disadvantages of the used methods of human speech recognition are determined.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ПК – персональний комп’ютер

МК – мовний корпус, база даних аудіофайлів та транскрипцій текстів, різновид корпусу текстів

ALR – автоматизове читання по губах (англ. Automated Lip Reading)

API – опис способів взаємодії одного ПЗ з іншими (англ. Application Programming Interface)

ASR – автоматичне розпізнавання мови (англ. Automatic Speech Recognition)

CUDA – програмно-апаратна архітектура паралельних обчислень продукції Nvidia (англ. Compute Unified Device Architecture)

DTW – алгоритм динамічної трансформації часової шкали (англ. Dynamic Time Warping)

GPGPU – техніка використання графічного процесора відеокарти, призначеного для комп’ютерної графіки, з метою математичних обчислень (англ. General–Purpose Computing on Graphics Processing Units)

HMM – прихована марківська модель (англ. Hidden Markov Model)

SSI – інтерфейси безмовного доступу (англ. Silent Speech Interfaces)

STT – розпізнавання мовлення, процес перетворення мовленнєвого сигналу в текстовий потік (англ. Speech To Text)

OpenCV – бібліотека алгоритмів комп’ютерного зору, обробки зображень (англ. Open Source Computer Vision Library)

SDK – набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи (англ. Software Development Kit)

RNN – рекурентні нейронні мережі (англ. Recurrent Neural Network)

ВСТУП

Протягом останніх десятиліть набули розвитку технології розпізнавання мови, які використовуються у повсякденному житті людини: від голосових команд пошуку в браузері або побудови маршруту по навігатору, голосового перекладу у телефоні до біометричної перевірки або систем аналітики, які записують телефонні дзвінки та збирають дані для підвищення конверсії. Сучасні системи розпізнавання мови досягають задовільних показників точності в контрольованих умовах [1], та деякі компанії як Google, стверджують, що рівень помилок у словах становить менше 8% під час використання голосових команд і цей показник зменшується з часом для користувачів із різних мовних середовищ [2].

Незважаючи на це, такі системи все ще покладаються виключно на аудіосигнал, на який потенційно впливає шум навколишнього середовища, крім того, звук може сильно відрізнятися залежно від статі чи віку. Наприклад, система обробки мови погано розпізнає голос літніх і дітей. Також розпізнавання мови відіграє важливу роль для військових, які не можуть передавати голосові команди у шумному середовищі або люди після ларингектомії та інших операцій з видалення гортані, голосових зв’язок, або які мають проблеми з мовним апаратом, німі. У таких випадках звичайне розпізнавання мови за допомогою мікрофона неефективне, оскільки рівень шуму або проблема передачі чіткої мови та навіть її відсутність унеможливлює виділення фонеми з аудіосигналу та відповідно не визначає сказаний текст. У таких випадках використовується система обробки звуку. Він заснований на прийомі та обробці звукових сигналів на ранніх стадіях артикуляції.

Сучасні проєкти обробки команд використовують SSI – це інтерфейси безмовного доступу (англ. Silent Speech Interfaces), які використовують не мікрофон, а інші сенсори, не схильні до впливу шумів і є доповненням до

оброблених акустичних сигналів. Наприклад люди здатні інтерпретувати пов’язану контекстну інформацію, таку як рухи губ, голови та тіла, жести рук, міміку та специфічні характеристики мовного сигналу, такі як просодія, які стають важливою частиною процесу спілкування. Хоча деякі з цих інформаційних підказок можуть забезпечувати певну надлишковість або бути менш важливими для спілкування, інші можуть бути цінними для повного розуміння повідомлення. Розуміючи ширше мовленнєву комунікацію, SSI допомагає розпізнати контекстну інформацію без звуку, таким чином, мовний аудіосигнал є лише кінцевим результатом більшого набору подій, які система розпізнавання мови на основі інтерфейсів безмовного доступу може обробити.

Всупереч значній роботі у цій галузі, у літературі про SSI все ще бракує комплексного погляду на ключові аспекти та технології безмовного доступу, які сильно відрізняються один від одного, але кількість проєктів збільшується, наприклад: пристрій, заснований на постійній магнітній артикулографії [2], або візуальне мовлення за допомогою сенсора глибини Kinect [3]. Таким чином, ресурси та набори даних стають відкрито доступними для дослідницької спільноти, але у деяких випадках вони дорогі у виробництві або більшість досліджень розпізнавання мови в аерокосмічній галузі пов’язана з військовим ПЗ, тому результати часто засекречені або важкодоступні. У цій роботі буде використано поєднання мікрофона та відеокамери, оскільки ці пристрої є поширеними та існує велика кількість програмних бібліотек у загальному доступі для обробки кадрів, аудіо і поширений мовний корпус із великої кількості відеозаписів для порівняння з іншими проєктами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування актуальності обраної теми

Обробка мови – це загальний напрямок штучного інтелекту та математичної лінгвістики та вивчає проблеми комп’ютерного аналізу та синтезу природних мов. Розв’язання цих проблем означає створення зручнішої форми взаємодії людини та комп’ютера, тому існують системи на основі SSI, які використовують ультразвук або оптичні камери та фіксують рухи обличчя або шиї як сигнал, незважаючи на вхідний звук.

1.1.1 Використання голосових команд

Голосові команди добре вписується в концепцію побудови природномовного інтерфейсу користувача. Однак не тільки інтерфейс користувача може бути метою для вивчення обробки природної мови. Крім цього, такі технології вже набули широкого поширення у різних сферах життя (рисунок 1.1), наприклад:

- телекомунікації: автоматизація обробки вхідних та вихідних дзвінків за допомогою створення голосових систем самообслуговування зокрема для проведення опитувань, анкетування, збору інформації, інформування, отримання довідкової інформації та консультування, зміни параметрів послуг, замовлення товарів та будь-яких інших сценаріїв.
- система "Smart house": інтерфейс керування системою дому;
- побутова техніка та роботи: інтерфейс електронних роботів, голосове керування побутовою технікою тощо;
- автомобілі: керування системою навігації, радіо, дзвінками та іншим у салоні автомобіля не відволікаючись від дороги;
- безпека, включаючи використання з іншими біометричними

сканерами для багатофакторної автентифікації;

- авіація: керування кабіною, створення повідомлень у надзвичайних ситуаціях або у шумному оточенні (літаки, гелікоптери тощо);
- ПК: голосове введення в комп’ютерних іграх та інших програмах;
- сервіси, спрямовані на полегшення життя людям з обмеженими можливостями (глухонімота, афазія, алалія тощо);
- смартфони та інші портативні пристрої: команди для навігації, посилання SMS або інших повідомлень, набір номера, перекладач, нагадування, створення нотаток і налаштування будильника тощо.

Також відомо, що уряди деяких країн також досліджують використання технології розпізнавання голосу з метою безпеки, наприклад Агентство національної безпеки США розробило власну систему розпізнавання у 2004 році та неодноразово випускає нові та покращені версії [4].

Рисунок 1.1 – Приклади використання голосових команд

Цей перелік можна постійно поповнювати, але найбільшого поширення такі системи набули як голосові помічники у вигляді програмних додатків на смартфонах, ПК або спеціальних пристроїв, схожих на аудіоколонки.

У цій роботі детально розглядаються сфери, де існує проблема високого рівня шумів або відсутності звукового сигналу, наприклад для людей з обмеженими можливостями, авіації, автомобілів з недостатньою шумоізоляцією, а також відновлення діалогів у німому кінематографі.

1.1.2 Обробка мови людей із обмеженими можливостями

Мова – це складний процес, який включає як руху рота, так і вібрації смужок пружної м’язової тканини, званих голосовими зв’язками, в горлі. Людина може розмовляти завдяки наявності гортані — ділянки дихальної системи, де розташовані органи, відповідальні за утворення голосу. Перед відтворенням будь-якого звуку, голосові зв’язки людини стискаються і вібрують під натиском повітря, що виходить з легких. З цієї звукової хвилі, завдяки рухам губ, утворюються літери, слова та речення.

Люди з обмеженими можливостями можуть скористатися спеціальними ПЗ та пристроями для розпізнавання команд. Для глухих або людей з вадами слуху система розпізнавання мови використовується для автоматичного створення субтитрів розмов, таких як обговорення в актових залах або лекцій у класах. Працівники з проблемами слухового апарату хочуть брати участь в обговореннях на зустрічах і лекціях у реальному часі, а не обмежуватися читанням обробленої стенограми через години або дні. Розпізнавання мови також необхідне для людей, у яких є труднощі з використанням рук, наприклад легкі стресові травми або інвалідність, яка не дозволяє використовувати звичайні пристрої введення для комп’ютера. Фактично, люди, які мають пошкодження частини опорно-рухового апарату та нервової системи, стали важливою групою користувачів для систем розпізнавання мови без використання клавіатури та схожих пристроїв.

Розпізнавання мови використовується в телефонії для глухих, наприклад при перетворенні голосової пошти в текст, службах ретрансляції та телефонії з субтитрами, але часто вона обробляється неправильно, що призводить до іншого результату на папері.

Також відомо, що за даними Всесвітньої організації охорони здоров’я, на наразі у світі налічується близько пів мільярда людей, які позбавлені можливості говорити [5]. Їхня вада зазвичай пов’язана з вродженою глухотою — не чуючи мови дорослих людей, глухі діти просто не можуть

навчитися говорити. Також основною причиною проблеми — це ушкодження голосових зв’язок і людина не може озвучити слова, хоча знає, як вимовляти слова.

Людина з пошкодженими голосовими зв’язками не може відтворювати звукову хвилю, тобто практично позбавлена голосу. У деяких випадках вони можуть вимовляти слова, використовуючи так званий «голосоутворюючий пристрій» на шиї, який створює звукові хвилі замість голосових зв’язок, або інфрачервоний датчик, встановленого на окулярах, який вловлює найменший рух м’язів щоки [6]. Також існують джойстики, які допомагають людини за допомогою руху щелеп керувати комп’ютером, який відтворює звуки, але це незручно. Ці пристрої мають два недоліки: відтворювані звуки дуже схожі на голос робота, а також користувачі повинні весь час носити їх з собою і притискати до шиї чи щоки кожного разу, коли потрібно щось сказати. Тому покращення таких систем на основі інших датчиків або систем є актуальною проблемою на сьогодні.

1.1.3 Розпізнавання мови пілотів

Кабіни пілотів стають складними та містять велику кількість пристроїв керування. Це складне і стресове середовище, яке вимагає високі вимоги до уваги пілота. Також відомо, що рівень шуму в кабіні середнього винищувача може перевищувати 100 dB. Впровадження технології розпізнавання мови як альтернативна стратегія керування літаком звільняє руки та очі пілота, тим самим знизивши робоче навантаження і дозволивши йому краще зосередитися на головному завданні.

Проблеми досягнення високої точності розпізнавання мови за умов стресу і шуму особливо актуальні серед ПЗ для гелікоптерів та реактивних винищувачів. Програма розпізнавання мови використовується для встановлення радіочастот, керування системою автопілота, встановлення координат та параметрів випуску зброї, а також керування дисплеєм.

Система розглядається як головна конструктивна особливість сучасних літаків у зниженні навантаження на пілота, наприклад для широкого спектра функцій кабіни та дозволяє пілоту призначати цілі голосовими командами.

Відомо, що розпізнавання погіршується зі збільшенням перевантажень, наприклад у звіті з обробки мови шведських пілотів [7], зроблено висновок про те, що спеціальна техніка дихання значно покращувала результати розпізнавання у всіх випадках, але існує значна різниця в артикуляції між тими, хто говорить при високих навантаженнях через індивідуальну адаптацію та стиль мовлення. Методи боротьби з шумом включають тренування пілотів у шумному середовищі та шумозаглушення кабіни. Висновком було те, що спонтанна мова викликала проблеми у ПЗ літака, але обмежений словниковий запас і використання правильного синтаксису істотно підвищувала показники точності розпізнавання.

Проблема акустичного шуму є більш серйозною для гелікоптера не тільки через високий рівень шуму або вібрації, а й тому, що часто пілоти не використовують кисневу маску, яка зменшує акустичний шум у мікрофоні, але у таких випадках використовують мікрофони з пасивним або активним шумозаглушенням для уловлювання низькочастотного звуку та нейтралізації його до того, як він досягне вуха або SSI у вигляді електромагнітних пристроїв для відстеження рухів язика та губ. Також у сучасних гелікоптерах існують голосові команди управління радіозв’язком, налаштування навігаційною системою та керування системою перемикання цілей.

1.2 Огляд існуючих аналогів. Аналіз переваг та недоліків

З погляду технології, розпізнавання мови має довгу історію з кількома хвилями великих інновацій. Зовсім недавно було отримано значні досягнення у глибокому навчанні та обробці великих даних. Про успіхи свідчить кількість академічних публікацій, а також впровадження різних методів для розпізнавання мови, які мають свої переваги та недоліки.

1.2.1 Історія створення систем розпізнавання мови

За останні десятиліття спостерігається експонентне зростання технологій розпізнавання голосу. Ключовими областями зростання були: розмір словникового запасу, незалежність від того, хто говорить і швидкість обробки інформації.

Перші системи розпізнавання мови могли розуміти лише цифри (з огляду на складність мови). Bell Laboratories розробили систему Audrey, яка розпізнавала цифри, сказані одним голосом. Через 10 років, у 1962 році, IBM продемонструвала свою розробку – систему Shoebox, яка розуміла 16 слів англійською. Радж Редді був першим, хто застосував безперервне розпізнавання мови, бувши аспірантом Стенфордського університету наприкінці 1960-х років, а також його розробка видавала голосові команди для гри в шахи. Попередні системи вимагали від користувачів паузи після кожного слова. Відомо, що лабораторії США, Японії, Англії та СРСР розробили ще кілька апаратів, які розпізнавали окремі сказані звуки, розширивши технологію розпізнавання підтримкою інших мов та звуків [8].

Наприкінці 1960-х років Леонард Баум розробив математику марківських ланцюгів та алгоритм Баума-Велча для пошуку невідомих параметрів прихованої марківської моделі (HMM) в Інституті оборонного аналізу, яку потім застосували студенти для знаходження оцінки максимальної правдоподібності параметрів фонем у системі розпізнавання мови, але такі моделі надто спрощені, щоб пояснити багато спільних рис людських мов. Використання HMM дозволило дослідникам об’єднати різні джерела знань, такі як акустика, мова та синтаксис у єдину модель [9].

Приблизно у цей час радянські дослідники винайшли алгоритм динамічної деформації часу (DTW) і використовували його для створення системи розпізнавання мови, здатного працювати зі словником з 200 слів. DTW обробляв мову, поділяючи її на короткі кадри, наприклад сегменти по 10 мс, обробляючи кожен кадр як єдине ціле. Хоча DTW буде замінено

іншими алгоритмами, цей метод досі застосовують [10].

Починаючи з 1976 року комп'ютери IBM могли обробляти трохи більше ніж 1000 слів. Це число збільшилось приблизно до 20000 слів у 1980-х роках для пристрою з голосовим управлінням під назвою IBM Tangora або продукції Dragon Systems. У 1980-ті роки була представлена мовна модель n-грам разом з моделлю відкату Каца для отримання кращих результатів розпізнавання. Також відомо, що HMM виявився дуже корисним для моделювання мови та замінив алгоритм DTW, ставши панівним алгоритмом розпізнавання мови у 1980-х роках у поєднанні з ймовірнісним методом розпізнавання патернів часу. Окрім цього, у 1986 році було опубліковано новий клас рекурентної нейронної мережі (RNN), яку використовують для сучасних систем обробки мови [11].

У 1990-х роках словниковий запас типової комерційної системи розпізнавання мови був більшим, ніж середній словниковий запас людини, але такі пристрої були дорогими та користувачам потрібно було тренувати програму протягом години перед використанням. У 1992 році відбулося дві важливих події: AT&T розгорнула службу обробки дзвінків з розпізнаванням для маршрутизації телефонних дзвінків без участі людини-оператора та була представлена перша система Sphinx-II, яка не залежить від того, хто говорить, з великим словниковим запасом, безперервним розпізнаванням мови та вона показала кращі характеристики за оцінкою DARPA [12]. Обробка безперервного мовлення з великим словниковим запасом стала важливою подією в історії, оскільки минулі розробки вимагали паузи після кожної вимови. У 1997 році було запропоновано нову архітектуру рекурентних нейронних мереж – довга короткочасна пам'ять (LSTM), яку застосовують технологічні компанії як основу для автоматичного розпізнавання мови та має перевагу в обробці над RNN, HMM та іншими методами.

До 2001 року розпізнавання мови досягло 80-відсоткової точності, але прогрес якості розпізнавання зупинився на декілька років та компанії були

зацікавлені у збільшенні продуктивності пристроїв та підтримці великої кількості мов, наприклад арабської чи китайської. Також на початку нового тисячоліття технологія розпізнавання дозволила аналітикам з державних установ шукати у великих обсягах записаних розмов і ізолювати згадки ключових слів для виявлення злочинців. Записи можуть бути проіндексовані та аналітики можуть виконувати запити до бази даних, щоб знайти цікаві розмови. Деякі урядові дослідницькі програми були зосереджені на розвідувальних програмах розпізнавання мови, наприклад, програма EARS DARPA та Babel IARPA у США [13]. Також у ці роки розпізнавання мови за допомогою штучних нейронних мереж (наприклад, рекурентні мережі) були гірше у використанні, ніж метод неоднорідної змішаної моделі Гаусса та прихованої марківської моделі (GMM-HMM), оскільки на той час не вистачало великих навчальних даних та обчислювальних потужностей.

1.2.2 Приклади сучасних систем розпізнавання мови

Ефективність розпізнавання мови сьогодні є результатом десятиліть досліджень сотень вчених та інженерів, які працюють у галузі статистики, лінгвістики, семантики, алгоритмів прогнозування та обробки звуку. Більшість дослідників розпізнавання мови, які розуміли такі бар'єри як обчислювальні потужності, згодом відмовилися від нейронних мереж, щоб використати підходи до генеративного моделювання до недавнього відродження глибокого навчання, що розпочалося приблизно у 2010-х роках.

Станом на 2016 рік основні технологічні компанії, включно з Google, Apple, Amazon, Microsoft та Baidu, використовують мережі LSTM як основу для нових продуктів. Наприклад у 2015 році розпізнавання мови Google зазнало різкий стрибок продуктивності на 49% завдяки LSTM, навченому на основі коннекціоністської часової класифікації (CTC), який тепер доступний через Google Voice для всіх користувачів смартфонів та ПЗ цієї компанії [14]. Також вони мають доступ до великомасштабних обчислювальних ресурсів,

яке змушує компанію змінювати архітектуру та алгоритми розпізнавання мови та експериментувати з методами, які в минулому вважалися надмірно дорогими. Окрім цього, технологію використовують для розпізнавання рукописного тексту, що спрощує роботу з документами або перекладу.

У 2017 році дослідники Microsoft досягли історичної віхи, розв'язавши задачу розпізнавання розмовної мови Switchboard [15], а також використовують рішення для віртуальної голосової помічниці Cortana, голосового пошуку Bing, диктування тексту у Microsoft Office тощо. Задача включає розшифровку розмов англійською між незнайомцями, які обговорюють такі теми, як спорт і політика. Switchboard – це набір записаних телефонних розмов, які спілка дослідників мови використовувало понад 20 років для порівняння систем розпізнавання мови. Microsoft знизили рівень помилок приблизно на 12 відсотків порівняно з минулим рівнем точності, використовуючи низку покращень акустичних та мовних моделей на основі нейронних мереж та представили нову модель згорткової нейронної мережі у поєднанні з двонаправленою довготривалою короткочасною пам'яттю (CNN-BLSTM) для покращення акустичного моделювання. Крім того, розробники посилили мовну модель розпізнавача, використовуючи всю історію сеансу діалогу, щоб передбачити, що може статися далі, що дозволяє моделі ефективно адаптуватися до теми та локального контексту розмови. Але залишається робота не тільки з навчання комп'ютерів як транскрибувати вимовлені слова, а й розуміти їх сенс і наміри.

Основною проблемою для дослідження систем розпізнавання від технологічних компаній є закритий доступ до коду та деталей алгоритмів, оскільки це є комерційною таємницею, але існують інші рішення від наукових спільнот, ентузіастів чи некомерційних організацій. Прикладами популярних та загальнодоступних систем розпізнавання є CMUSphinx, HTK, які використовують приховану марковську модель (HMM) та створені університетами США або система Julius, яка є покращеною версією HTK із застосуванням триграм і має підтримку японської мови [16]. Також

розробники використовують Kaldi – це набір інструментів для розпізнавання мови з відкритим кодом, який створений на основі глибоких нейронних мереж з попередньою обробкою форми хвилі. Ще одним набором інструментів для розпізнавання мови є RWTH ASR, яка була створена групою вчених із університета в Аахені. Він складається з інструментів для розробки акустичних моделей та декодерів, а також компонентів для адаптивного, неконтрольованого або дискримінаційного навчання.

Сучасною системою обробки мови є безшумний мовний інтерфейс (SSI), який реалізується автоматизованим читанням по губах (ALR), аудіовізуально автоматичним розпізнаванням мови (AV-ASR) або субвокальним розпізнаванням (SVR). Для цих методів використовують інші пристрої, які відрізняються від мікрофона та фіксують рухи голосового апарату (губи, язик, гортань із голосовими зв'язками тощо). Також для таких систем застосовують згорткову нейронну мережа (CNN) для виділення ознак зображення.

1.2.3 Проєкти із застосуванням SSI

SSI покладається на неакустичні біосигнали, що генеруються людським тілом під час відтворення мови, щоб забезпечити спілкування, коли вербальне неможливе або небажане через шум. Пристрої з використанням SSI можуть обробляти різні біосигнали для забезпечення безшумної комунікації, такі як електрофізіологічні записи нейронної активності, електроміографічні (ЕМГ) записи рухів голосового тракту або пряме відстеження рухів артикулятора з використанням методів візуалізації. Залежно від ситуації, деякі методи обробки сигналів можуть краще за інших підходити для збору інформації, пов'язаної з мовою. Наприклад, методи ЕМГ та візуалізації працюють для пацієнтів після ларингектомії, які втратили можливість говорити після видалення голосових зв'язок, але не підходять для пацієнтів з важким паралічем [17]. З цих сигналів декодують передбачуване повідомлення, використовуючи алгоритми розпізнавання чи синтезу мови.

Окрім використання спеціальних медичних пристроїв, як правило, які закріплюються на шкірі, існують системи із використанням оптичної камери, які можуть аналізувати відеозображення людини, що говорить. Метою проєкту LipNet є розуміння природного мовлення, яке в основному складається з фонем, що зробило необхідним використання мовного корпуса, бази даних, що містить слова, фрази та моделі, які можуть ефективно працювати з ними. Форми, створені губами або мімікою людини можна обробити, а потім перетворити на звуки, група яких порівнюється зі словами. Ця технологія була успішно використана для аналізу старих відеозаписів, які не мали звукової доріжки або для розпізнавання розмови людей у галасливому оточенні [18]. В ALR існує фундаментальне обмеження продуктивності через омофони – це набори слів, які звучать по-різному, але включають однакові рухи губ, тому їх неможливо розрізнити лише на зображенні, але розв’язанням цієї проблеми є використання нейронних мереж для обробки змісту, який може уточнити значення цих слів або комбінування з обробкою аудіо.

Alter Ego – це проєкт, який розроблено у 2019 році дослідником Массачусетського технологічного інституту Арнавом Капуром, який ефективно служить нейронним інтерфейсом для роботи з комп’ютером і вимагає для роботи лише тонкої стимуляції мовних м’язів, не відкриваючи рота і без видимих ззовні рухів [19]. Зворотний зв’язок із користувачем здійснюється через звук, застосовуючи кісткову провідність, не порушуючи звичайне слухове сприйняття користувача. Щоб використовувати гарнітуру для виконання простого завдання, наприклад, пошуку погоди на комп’ютері, спочатку формується запит у розумі. Сенсори гарнітури зчитують біосигнали, які надсилаються з мозку в області голосового апарату для вимови вголос, наприклад на задню частину язика та піднебіння. Потім через пристрій обробляє ці сигнали як слова і виконує запит на ПК через вебз'єднання. Дослідники виявили, що ця система може розуміти свого власника у 92% випадків. Інтерфейс тестується в обмежених лікарняних

умовах, де він допомагає спілкуватися пацієнтам з розсіяним склерозом.

SpeakUP – це проєкт з використанням SSI, який створив Варун Чандрашекхар у 2021 році [20]. Цей пристрій може використовуватися пацієнтами з порушеннями мови для безшумного спілкування англійською мовою, просто вимовляючи слова або речення в роті без жодних звуків. SSI записує сигнали ЕМГ, які потім класифікуються у режимі реального часу з використанням навченої моделі машинного навчання. Виявлено, що ця система забезпечує точність розпізнавання слів на 90,1%, схожа на Alter Ego і є дешевою у виробництві.

Ouisper – це ще один спосіб отримання безмовної інформації за допомогою візуалізації [21]. Ультразвукове дослідження – це неінвазивна та клінічно безпечна процедура, яка дозволяє в реальному часі візуалізувати один із найважливіших артикуляторів системи мовлення – язик. Розміщений під підборіддям ультразвуковий сенсор Ouisper може забезпечити огляд язика і використовується для розпізнавання мови, який у цьому випадку називається безшумним вокодером.

1.2.4 Недоліки систем розпізнавання мови

Хоча дослідники із Microsoft та інших організацій досягли людського рівня розпізнавання розмовної мови, ґрунтуючись на результатах Switchboard [15], але залишається велика кількість нерозв’язних проблем. Тест від Switchboard простий, тому що кожен, хто говорить, записаний на окремий мікрофон та в одному каналі аудіо не перекриваються різні голоси. Люди можуть розуміти кількох людей, які розмовляють одночасно. Хороша система розпізнавання мови повинна сегментувати аудіо на підставі того, хто говорить (діаризація). Також цікавить показник семантичних помилок, що рідко вказують у звітах з тестування систем, хоча на практиці користувачі часто стикаються з цим у рішеннях від технологічних компаній, наприклад помилки розпізнавання голосових команд Google Assistant. Окрім цього,

система повинна розуміти аудіо від кількох розмовляючих (поділ джерел), але такі системи у розробці та мають посередні результати. Також залишаються проблеми з фоновим шумом для систем без застосування SSI. Існує ще один недолік, який стосується рішень із загальним доступом (Kaldi, Julius тощо), оскільки вони не вміють обробляти слова у контексті – минулі розмови, вирази обличчя, знання про людину, яка говорить. Також багато рішень вимагають постійного підключення до мережі, що робить їх марними для багатьох місць через обчислювальну потужність або обсяг навчених моделей, які необхідно виносити за межі пристрою розпізнавання на окремі сервери (рішення від Google, Microsoft, проєкт Alter Ego, ПЗ Kaldi та HTK).

Основним недоліком багатьох рішень субвокального розпізнавання у тому, що один той самий користувач з високим рівнем точності за один день, може мати посередній рівень наступного дня, а між двома користувачами відрізняється ще більше [19, 20]. Окрім цього, пристрої можуть бути незручними або дивно виглядати, оскільки потрібно кріпити набір електродів до шкіри горла і їх кількість впливає на точність розпізнавання. Також у багатьох системах існує обмежений набір слів, який потрібно тренувати по декілька годин, чого недостатньо для спілкування з іншими, але вистачає для управління технікою. Підсумком можна назвати те, що ця технологія є найбільш прикладною для пацієнтів, пілотів та військових, що вже застосовують, але не для звичайних користувачів, яким достатньо методів розпізнавання на основі мікрофона.

У більшості робіт присвячених розпізнаванню мови з ЕМГ або читання по губах розглядається свій dataset, часто з дуже обмеженим набором слів або навіть записаний власноруч (таблиця 1.1). Окрім цього, аналіз мови за візуальними ознаками набагато чутливіший до зміни користувача, різних наборів слів, акцентів. Однак розпізнавання мови, засноване на відео у загальному випадку складніше за аналіз проєктів обробки звукових сигналів.

Людська мова містить близько 50 фонем (мінімальна помітна одиниця аудіопотоку) у той час, як по губах можна розрізнити 10-15 візем (груп

візуально нерозрізнених фонем). Таким чином, послідовність візем часто може не відповідати конкретному слову і точність читання по губах сильно залежить від контексту. Крім того, навіть серед людей, що говорять на одному діалекті, відповідність між рухами губ і вимовленими віземами може дуже відрізнятися, що робить майже неможливим побудову універсальної моделі розпізнавання без інформації про форми руху губ або інших органів. Ще одним недоліком для таких систем є проблеми з неекспонованими зображеннями при слабкому освітленні, хоча існують проєкти, які використовують глибокі нейронні мережі для покращення освітленості та видалення шумів на зображенні.

Таблиця 1.1 – Порівняння систем розпізнавання мови

Система	SSI	Технології	Dataset	Тип команди	Точність
Kaldi	-	GMM/HMM	Switchboard	Речення	~80%
Sphinx	-	HMM	EUSTACE	Речення	71,99%
Microsoft	-	RNNLM	Switchboard	Речення	93,8%
Ouisper	+	УЗД	Власний	Слово	~70%
SpeakUP	+	ЕМГ	Власний	Словосполучення	80,1%
Alter Ego	+	ЕМГ	Власний	Словосполучення	91,2%
Fu et al (2008)	+	Кохлеарний імплант	AVICAR	Числа	37,9%
Gergen et al (2016)	+	ALR	GRID	Слово	86,4%
LipNet	+	ALR	GRID	Речення	95,2%

Також часто автори наводять точність своїх моделей, навчених на аудіо або візуальних ознаках окремо, але неможливо порівняти з моделями, які поєднують обидва способи. Тому порівнювати системи між собою складніше, ніж зі звуковою інформацією. Наприклад, у проєкті дослідникам вдалося досягти точності 79%, використовуючи лише візуальні ознаки [22].

1.3 Обґрунтування доцільності вдосконалення існуючих рішень

Найчастіше під розпізнаванням мови передбачають перетворення аудіопослідовності запису голосу людини на текстові дані. Однак, використання не тільки звукової інформації, а й відео може значно покращити якість розпізнавання або навіть замінити аудіо.

Основною проблемою для великої кількості проєктів розпізнавання команд з SSI-підходом полягає у тому, що в них застосовуються незручні прилади, які потрібно закріплювати на шкірі, що не знайшло масового застосування, лише в експериментальних дослідженнях на пацієнтах, військових чи астронавтах. Також у багатьох системах обмежена кількість команд, наприклад пару десятків у дослідному проєкті NASA, а також мають змінну точність для одних тих самих слів одного користувача через зміщення сенсорів на тілі у повсякденному застосуванні. Тому для зручності краще використовувати інші пристрої (відеокамера або камера глибини).

Проблему високого рівня шумів можна вирішити за допомогою алгоритму автоматизованого читанням по губах з відеопотоку, оскільки звуковий сигнал не використовується, а тільки віземи на кадрах. З іншої сторони, у таких системах залишається проблема омофонів, які включають однакові рухи губ для різних слів, яку можна вирішити додаванням обробки звуку, але такий спосіб погано працює у галасливому оточенні без застосування додаткової фільтрації.

Відомо, що у деяких дослідженнях застосовували вже розпізнаний текст у контексті для передбачення наступних слів, що значно покращило точність розпізнавання мови в аудіофайлах [15]. Такий метод можна спробувати для розпізнавання візем, наприклад використовувати зображення форм губ для різних фонем у контексті. Окрім цього, досягти ідеального розпізнавання неможливо – людина може вимовити не пов'язані з собою слова в одному реченні, але така ситуація є досить рідкісною у повсякденному застосуванні.

1.4 Постановка задачі

Метою кваліфікаційної роботи є дослідження впливу використання інтерфейсу безмовного доступу (SSI), що забезпечує визначення початкової фази звукового ряду, асоційованого із початком мовлення, на точність розпізнавання голосових команд у різних звукових оточеннях. Також у цьому проєкті розглядається алгоритм з використанням камери та стандартизований набір даних із відеороликів для навчання та порівняння з іншими системами розпізнавання.

Для досягнення поставленої мети мають бути вирішені такі задачі:

- аналіз методів попередньої обробки та розпізнавання мови;
- аналіз методів розпізнавання візем людини, що розмовляє;
- створення моделі розпізнавання голосових команд, вдосконаленої завдяки аналізу віземи людини, що розмовляє;
- реалізація моделі розпізнавання голосових команд на основі аналізу звукового ряду;
- реалізація моделі розпізнавання голосових команд на основі комбінації аналізу звукового ряду та зображення губ диктора;
- виконання експериментів із зашумленим мовним корпусом;
- аналіз отриманих результатів.

Для подальшого розвитку цієї роботи можна застосувати алгоритми зменшення шуму для аудіосигналу або розв’язати задачу визначення візем в умовах недостатньої яскравості або положення обличчя під іншим кутом.

Також можна реалізувати проєкт із застосуванням інших пристроїв, наприклад камери глибини, оскільки вона може фіксувати різну відстань до рота, що має вплинути на точність розпізнавання візем або скомбінувати з оптичною камерою та мікрофоном. Крім цього, для поліпшення цієї роботи можна використовувати не одну камеру, а декілька, які розташовані під різним кутом, щоб поліпшити розпізнавання сказаних фонем, але для цього потрібні відповідні відеодані.

2 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РОЗПІЗНАВАННЯ ГОЛОСУ

2.1 Визначення апаратної бази для виконання експериментальної частини проекту

Апаратна платформа – це набір сумісного обладнання, на якому можна запускати таке програмне забезпечення, як розпізнавання голосових команд. Для обробки голосу або жестів необхідно використовувати сумісні пристрої для фіксування цих сигналів та виконання програмного коду, який повинен виконувати швидкі обчислення і забезпечувати точність розпізнавання.

2.1.1 Мікрофон

Мікрофон – це прилад, що перетворює звукові коливання на коливання сил електричного струму. Записаний звук складається з безлічі звукових хвиль, що одночасно потрапляють на датчик мікрофона у деякий проміжок часу, в результаті чого отримується довгий вектор з чисел – це амплітуди (гучність) сигналу протягом невеликого часу. Крім того, мікрофони неоднаково чутливі до звукового тиску та можуть сприймати різні рівні без спотворень. Також чутливість впливає на те, наскільки якісно мікрофон перетворює акустичний тиск у вихідну напругу.

Через відмінності у конструкції мікрофони мають характерні реакції на звук. Найбільшого поширення у звукозаписі мають електродинамічні (котушкові, стрічкові) і конденсаторні мікрофони. Динамічні мікрофони забезпечують якісні електроакустичні параметри, а саме: великий динамічний діапазон, стійкість до механічних, кліматичних навантажень, низький рівень шумів, але поступаються конденсаторним мікрофонам за чутливістю, нерівномірністю та рівнем перехідних процесів. Конденсаторні (електретні) мікрофони мають низку переваг, які дозволяють широко

використовувати їх у студійній практиці. До основних можна віднести: низький рівень перехідних спотворень (через малу масу діафрагми), широкі частотний і динамічний діапазони, висока чутливість, мала чутливість до магнітних перешкод тощо.

Загальний спосіб зберігання цифрового звуку – це відображення напруги звуку, яке при відтворенні відповідає рівню сигналу на окремому каналі з певною роздільною здатністю, кількістю бітів на вибірку через проміжки часу (частоти вибірки).

Існує три основні групи форматів аудіофайлів:

- нестиснені аудіоформати, такі як WAV, AIFF, AU;
- формати зі стисненням без втрат: FLAC, TTA, MPEG-4 SLS/ALS;
- формати зі стисненням з втратами: MP3, Ogg Vorbis, AAC, WMA.

Формат без втрат вимагає більше часу на обробку, ніж стиснуті формати, але займає більше місця. Нестиснені аудіоформати кодують як звук, так і тишу з однаковою кількістю бітів в одиницю часу. При кодуванні хвилини тиші в форматі нестиснутому виходить файл того ж розміру, що і при кодуванні хвилини оркестрової музики в файлі.

У форматах зі стисненням з втратами обробляються дані шляхом відкидання їх частин. Процес намагається звести до мінімуму обсяг даних, що зберігаються у файлі, зменшуючи його розмір та якість. Стиснення з втратами, з іншого боку, видаляє або зменшує певні типи інформації під час обробки даних, наприклад урізання максимальної частоти спектра.

Сучасні бібліотеки обробки звуку підтримують більшість форматів зі стисненням з втратами, однак слід уникати їх, оскільки відсутність даних може вплинути на точність, тому використовують такі формати як WAVE або WAF з великим обсягом файлів. Відомо, що достатніми параметрами для розпізнавання є частота дискретизації – 16000 Гц або вище та один звуковий буфер розміром близько 100 мілісекунд для збалансованої затримки (затримка між створенням звуку та його записом) [23]. Також потрібні навчальні дані у вигляді колекції пари аудіозапису з текстом.

2.1.2 Камера та відеоформати

Цифрова відеокамера – це оптичний пристрій, який записує кадри у цифровому вигляді. При влученні світла на елемент матриці відеокамера виробляє електричний сигнал, пропорційний кількості світла, що потрапило. Потім аналогові сигнали з елементів матриці зчитуються і перетворюються у цифрову форму цифро-аналоговим перетворювачем. Також важливою характеристикою камер є роздільна здатність матриці – здатність пристрою передавати дрібні деталі зображення. Висока роздільна здатність забезпечує більш точне відображення оригіналу, але у методах автоматизованого читання по губах застосовують низьку, наприклад кадри розміром 360 на 240 пікселів або більше, чого достатньо для виявлення обличчя або рота на зображенні [24].

Також існують камери глибини, що знімають відео, у кожному пікселі якого зберігається не колір, а відстань до об'єкта у цій точці. Вони засновані на скануванні структурованим світлом – за допомогою проєктора (зазвичай інфрачервоного) формується проєкція світлової сітки на об'єкті та за допомогою камери фіксуються спотворення сітки. Вона працює у неяскравих приміщеннях або навіть без світла, але існують проблеми шумів у яскравих місцях та нестабільність глибини у часі. Такі камери використовують для ідентифікації особи, у системах розпізнавання команд, наприклад жестів, як було реалізовано у Microsoft Kinect або у наукових проєктах розпізнавання мови із SSI-підходом [3].

Формат відеофайлу – це тип формату файлу для зберігання цифрових відео даних у комп’ютерній системі. Відео майже завжди зберігається за допомогою стиснення з втратами для зменшення розміру файлу. Оскільки відеофайли можуть мати великий розмір, були розроблені програми, які називаються кодеками, які спрощують їх зберігання та обмін ними. Кодеки кодують дані, стискаючи їх для зберігання. Потім вони декодують ці дані, щоб розпакувати їх для перегляду та редагування. Найбільш поширеним

кодеком для стиснення відео є H.264, H.265 або AVC. Вони мають високі показники стиснення і продуктивності, а також підтримуються більшістю сучасних пристроїв, тому їх часто використовують у системах розпізнавання. Також розпізнавання мови сильно ускладнюється низькою яскравістю більшості практичних відеоматеріалів, які не дозволяють точно зчитувати просторово-часові характеристики людини під час розмови.

2.1.3 Одноплатний комп’ютер Raspberry Pi

Raspberry Pi – це серія невеликих одноплатних комп’ютерів (SBC), розроблених у Великій Британії Raspberry Pi Foundation спільно з Broadcom. На сьогодні існує три серії Raspberry Pi та випущено кілька поколінь кожної. Ці комп’ютери мають систему на кристалі (SoC) із вбудованим ARM – сумісним центральним процесором (CPU) та вбудованим графічним процесором (GPU). До нього можна під’єднати різні пристрої через USB, GPIO або інші порти, а операційна система та програми зберігаються на картах пам’яті microSD. Такі одноплатні комп’ютери використовують для домашньої чи промислової автоматизації та існують проєкти із розпізнаванням мови, оскільки його достатньо для обробки аудіо, але для виявлення візем краще застосовувати інші пристрої або під’єднатийого до іншого комп’ютера [25].

У цій роботі до Raspberry Pi 4 буде приєднано відеокамеру з мікрофоном, а також до локальної мережі для передачі аудіо та відеофайлів на інший ПК – сервер, на якому виконується ПЗ розпізнавання команд. Обчислювальна потужність складних алгоритмів із застосуванням машинного навчання вимагає використання паралелізму, який найшвидше буде виконуватися на ПК за допомогою обчислень загального призначення на GPU. Аналогом для побудови такої системи без окремого сервера є Nvidia Jetson, який за розмірами та SoC схожий на Raspberry Pi та додатково має підтримку CUDA для прискорення обчислень, але коштує дорожче.

2.1.4 Графічна карта з підтримкою CUDA

Compute Unified Device Architecture (CUDA) – це паралельна обчислювальна платформа компанії Nvidia та інтерфейс прикладного програмування (API), який дозволяє програмному забезпеченню використовувати певні типи графічних процесорів (GPU) для обробки загального призначення, підхід, що називається обчисленнями загального призначення на GPU (GPGPU). Вона надає прямий доступ до віртуального набору інструкцій GPU та паралельних обчислювальних ядер.

Обчислення на графічних процесорах дозволяють алгоритмам, що використовують паралельну обробку даних, досягати швидких результатів, наприклад, при виконанні серії математичних операцій над великими обсягами даних. При цьому кращі результати досягаються, якщо відношення числа арифметичних інструкцій до звернень до пам’яті досить велике. У результаті всіх описаних відмінностей, теоретична продуктивність GPU значно перевищує продуктивність CPU. Використовуючи як CPU, так і GPU у процесі розпізнавання мови, можна виконувати розпізнавання з використанням великих, а в деяких випадках і декількох моделей, отримуючи високу точність навіть у вбудованих та мобільних системах. Крім того, заздалегідь записаний мовний контент може бути розпізнано набагато швидше, наприклад у декілька десятків разів. Порівняно з однопотоковою реалізацією CPU, яка є стандартною архітектурою і використовується у розпізнаванні мови, створювати моделі набагато ефективніше, використовуючи кілька графічних процесорів, а не кластери CPU, оскільки акустична модель, для навчання якої зазвичай потрібно понад 1000 годин, може бути навчена за 10 годин на одному графічному процесорі [26].

Також існують системи із застосуванням цієї платформи, наприклад HYDRA – це науковий проєкт Університету Карнегі-Меллона, у якому досліджують нові високопаралельні обчислювальні платформи для задачі розпізнавання мови, наприклад гетерогенної платформ CPU-GPU.

2.2 Аналіз технологій для вирішення задачі

Для створення системи розпізнавання мови людини необхідно застосувати технології за такими вимогами: підтримка керування камерою та мікрофоном, відображення результату розпізнавання у вигляді тексту або відеопотоку з розпізнаними віземами, невелике споживання оперативної пам’яті, висока продуктивність.

2.2.1 Бібліотека OpenCV

OpenCV (Open Source Computer Vision Library) – це безкоштовна бібліотека комп’ютерного зору, спочатку розроблена компанією Intel. Вона має понад 2500 оптимізованих алгоритмів комп’ютерного зору, обробки зображень та чисельні алгоритми загального призначення з відкритим кодом. Для підтримки деяких з областей, наприклад, виявлення об'єкта, OpenCV включає бібліотеку статистичного машинного навчання [27]. Хоча вона реалізована мовами C та C++, вона може застосовуватись у C#, Java, Python, Go тощо. Її популярність обумовлена тим, що:

– безкоштовна, випущена під ліцензією BSD, що дозволяє вільно використовувати її в дослідних та комерційних цілях;

– кросплатформеність для операційних систем GNU/Linux, Mac OS X, Windows і Android, а також для різних апаратних архітектур, таких як x86, x64 (ПК), ARM (телефони, Raspberry Pi тощо);

– детальна документація, організація активно дбає про те, щоб довідкова документація для розробників була повною та актуальною.

Для зручної роботи із кадрами відеопотоку для розпізнавання рота було обрано OpenCV. Також існують альтернативні бібліотеки, наприклад SimpleCV, Accord.NET Framework, FastCV Computer Vision тощо, однак вони не є настільки поширеними та деякі з них залежать від однієї мови програмування або апаратної архітектури.

2.2.2 Бібліотека CMUSphinx

CMUSphinx на сьогодні є найбільшим проєктом з розпізнавання мови людини та розроблений в Університеті Карнегі-Меллона (CMU), в якому протягом кількох десятиліть вирішувалося задача розпізнавання мови. Також проєкт з 2000 року має відкритий код кількох компонентів розпізнавання і підтримку процесорів ARM. Бібліотека містить такі програми та бібліотеки, як Pocketsphinx, яка приймає на вхід довільні акустичні моделі, граматики та словники, а також звуковий потік з мікрофона або файлу, а результатом є розпізнаний текст. Окрім цього містить програмне забезпечення для навчання акустичної моделі, компіляції мовної моделі та загальнодоступний словник вимови, cmudict. Остання версія має підтримку багатопроцесорного навчання для прискорення розпізнавання та алгоритми зниження шуму, проте збільшення точності не перевищує 10% [28].

2.2.3 Мовний корпус GRID

Для навчання системи розпізнавання потрібен мовний корпус, приклади яких відображені на рисунку 2.1. GRID – це колекція з десятків тисяч коротких відеороликів, у яких 34 добровольці читають англійською безглузді речення та накладені підписи до них. Кожен файл триває три секунди, а кожне речення відповідає шаблону: команда, колір, прийменник, літера, цифра, прислівник. Приклади таких речень: «place blue in M one soon», «set blue by A four please» і «place red at C zero again».

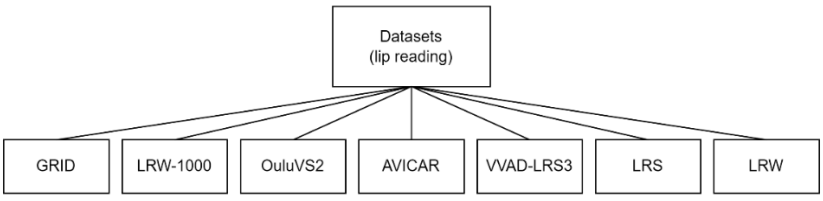


Рисунок 2.1 – Приклади корпусів для ALR

До переваг цього корпусу можна віднести велику кількість відеороликів, на яких у світлому приміщенні добре видно рух губ різних людей, чого буде достатньо для побудови системи розпізнавання, але для практичного застосування краще використовувати інші корпуси, де положення голови буде не анфас, не шаблонні речення та темне освітлення. Для виконання цих умов можна віднести для аналізу не різкого руху рота корпус OuluVS2, який записано шістьма камерами з п’яти різних видів, розташованих між фронтальним та профільним видами 50 людей, але головною проблемою буде реалізація алгоритму ALR з кадрів, де обличчя людини у профіль

LRW-1000 раніше називали CAS-VSR-W1k і часто застосовують для розпізнавання, але для даної роботи складно використовувати, оскільки не володію китайської мовою, хоча до LRW-1000 входить 18018 зразків відео від приблизно 2000 людей і це гарний корпус для практичного застосування, оскільки містить різне освітлення, положення голів та не лабораторні речення.

Найкращий мовний корпус для порівняння з іншими в різних звукових умовах – це AVICAR, оскільки записано обличчя 86 людей у різних положеннях усередині автомобіля та з 7 мікрофонів, а також має 5 рівнів шуму залежно від швидкості руху авто. Але проблемою є складність отримання всіх даних, крім обмеженої кількості людей, оскільки посилання на файли у момент написання цієї роботи були недоступні (останнє оновлення у 2004 році) або потрібне підтвердження від дослідників з Університету Іллінойсу.

Також поширена група корпусів LRS, яка має різні версії (LRS, LRS2, LRS3-TED, MV-LRS тощо) та створена телебаченням BBC, а кожне речення має довжину 100 символів. Через велику кількість положень камер, об’єм файлів та різноманітність контенту набір даних LRS2 є складнішим (75,2% кадрів з не фронтальним обличчям), ніж набір даних LRS або MV-LRS та рекомендується для проєктів з найкращими алгоритмами розпізнавання.

Окрім цих даних, не вистачає стандартизованого набору анал огічних відеороликів українською мовою, за допомогою яких можна було порівняти результати для різних систем без створення власного корпусу.

2.2.4 Бібліотека dlib

Бібліотека dlib надає функції, які можна використовувати для розпізнавання облич та створена мовою програмування Python. Для мережі розпізнавання облич dlib вихідним вектором ознак є 128-d (тобто список зі 128 дійсних чисел), який використовується для кількісної оцінки особи, а навчання мережі виконується за допомогою триплетів [29]. Також для роботи з цією бібліотекою потрібно використовувати OpenCV, має підтримку CUDA та містить два методи виявлення осіб:

- детектор осіб HOG + Linear SVM, точний та ефективний у обчислювальному відношенні;
- детектор осіб Max-Margin (MMOD) CNN, що відрізняється високою точністю та надійністю, здатний виявляти обличчя при різних кутах огляду, умовах освітлення та оклюзії.

MMOD CNN може допомогти для виявлення візем у темному приміщенні, що може покращити точність розпізнавання у повсякденному застосуванні. Крім цього, пакет містить швидкі методи обробки зображень, що є критичним для створення програмного забезпечення. Крім цього, пакет містить швидкі методи обробки зображень, що є критичною характеристикою для створення ПЗ.

2.2.5 Мова програмування Python

Python – це високорівнева мова програмування загального призначення. Вона має динамічну типізацію та garbage collector, підтримує кілька парадигм програмування, включаючи структурне, об'єктно-

орієнтоване та функціональне програмування. Python використовує качину типізацію та має типізовані об'єкти, але нетипізовані імена змінних. Стандартна бібліотека надає інструменти, що підходять для багатьох задач: автоматизація, аналітика, наукові обчислення, обробка тексту тощо [30].

До особливостей цієї мови можна віднести простоту, популярність серед розробників у сфері машинного навчання та велику кількість пакетів для роботи з нейронними мережами та зображеннями, що стало головним параметром для створення системи розпізнавання. Перша спроба була із застосуваннями мови програмування Go, але через відсутність повної підтримки OpenCV і CUDA унеможливило створення повноцінної системи розпізнавання мови з SSI-підходом у реальному часі.

2.2.6 Бібліотека PyCUDA

Для створення системи розпізнавання мови у реальному часі із застосуванням відеопотоку необхідно обробляти зображення розміром кілька мільйонів пікселів за мілісекунди. Для розв'язання цієї задачі існує бібліотека PyCUDA, яка надає простий Pythonic-доступ до API паралельних обчислень CUDA від Nvidia. Базовий шар PyCUDA створений мовою C++, тому практично всі функції без накладних витрат та ефективно виконуються як мовами C /C++ . Всі помилки CUDA автоматично перетворюються на омилки у Python та абстракції PyCUDA роблять програмування на CUDA навіть зручнішим, ніж середовище виконання Nvidia на основі C.

Очищення об’єктів прив’язана до часу життя об’єктів – це ідіома, яку у C++ часто називають RAII, значно спрощує написання правильного коду без витоків пам’яті та збоїв. Також ця бібліотека містить механізм обробки залежностей, тому вона не буде від’єднуватись від контексту доти, доки вся виділена у ньому пам’ять також не буде звільнена, що спрощує процес розробки програм із підтримкою паралелізму. Також існує важлива перевага PyCUDA у порівнянні з API CUDA – вона автоматично обирає найкраще

рішення за швидкістю під час виконання коду без ручного налаштування. Аналогічний метод використовується у ряді відомих обчислювальних пакетів, у тому числі в ATLAS та FFTW. Хоча для цього потрібні складні процедури драйвера оптимізації, можна керувати PyCUDA, не виходячи з Python [31].

2.3 Аналіз методологічного підґрунтя для рішення задачі

Для створення системи розпізнавання голосових команд необхідно застосувати алгоритми, які можуть визначити фонему з аудіо або візему кадру відеопотоку у режимі реального часу.

2.3.1 Алгоритм прихованих марківських моделей (HMM)

Марківські моделі є потужним засобом моделювання різних процесів та розпізнавання образів. Ці моделі дозволяють враховувати безпосередньо просторово-часові характеристики сигналів і тому набули широкого застосування у системах розпізнаванні мови, а останнім часом – зображень. HMM визначається за наступною формулою:

$$\lambda = (A, B, \pi), \tag{2.1}$$

де A – матриця ймовірностей переходів;

B – матриця ймовірностей спостережень вихідних значень;

π – вектор ймовірностей початкових станів.

Матриця A складається з елементів a_{ij} - ймовірностей переходу зі станів i у j. Матриця B містить елементи $b_i(o_k)$ - ймовірність спостереження у стані i вектора ознак o_k. π складається з компонентів π_i - ймовірностей знаходження в i-му стані у початковий момент часу [32].

За допомогою HMM складають статистичні моделі фонем, слів та

цілих фраз. Вибір конкретного мовного об’єкта залежить від завдань, які повинна вирішувати система розпізнавання мови, що розробляється. Також можна виділити такі підходи до створення HMM (вони можуть бути як взаємозаперечними, так і взаємодоповнювальними):

- складають моделі фонем, які можна поєднувати у слова;
- складають окремі HMM для кожного слова зі словника та при розпізнаванні обирають «найбільш відповідну». Такий підхід підійде для розпізнавання слів, що окремо стоять у реченні;
- фонеми моделюються за допомогою трьох станів – початкового, середнього та кінцевого (рисунок 2.2), оскільки мовний тракт неспроможен змінювати свої характеристики миттєво та при переході від фонеми до фонеми відбувається його "перемикання" через проміжні стани;
- складають одну HMM, склеюючи HMM для слів через проміжні стани (наприклад, тишу) згідно з граматикою мови. Це необхідно для розпізнавання злитого мовлення;
- фонеми звучать по-різному серед різних фонем і цей ефект називається коартикуляцією.

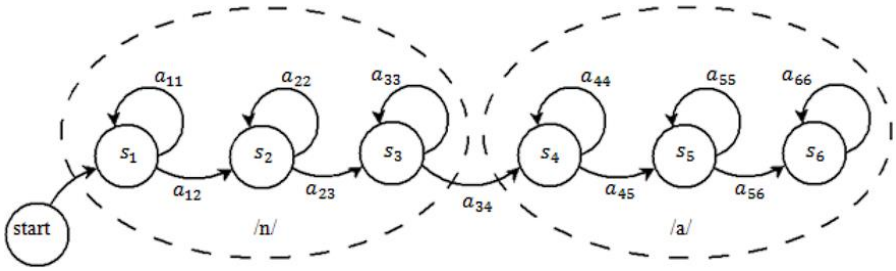


Рисунок 2.2 – Приклад алгоритму HMM для фонем /n/ і /a/

Залежно від того, чи враховуватиметься, чи ігноруватиметься явище коартикуляції, існує два типи фонем:

1) монофони – коартикуляція ігнорується, складаються моделі окремих фонем. Цей підхід має важливу перевагу: фонем у мові зовсім

небагато (в українській - 38), та з них можна скласти будь-які слова, так що розпізнавання буде зводитися до визначення ланцюжка сказаних фонем, а словник фактично необмежений, але недоліком є невисока точність.

2) трифони – коартикуляція враховується шляхом складання окремих моделей для фонем серед інших фонем. Наприклад слово "назад": використовуючи Міжнародний фонетичний алфавіт, його можна описати ланцюжком фонем "n-a-z-a-t". Тут фонема /a/ зустрічається двічі, але через коартикуляцію для неї потрібно скласти дві окремі моделі: "na+z" і "z-a+t". Це набагато складніший підхід, але точність розпізнавання вище, ніж використання монофонів [9].

2.3.2 Алгоритм динамічної трансформації часової шкали (DTW)

У системах розпізнавання мови, що містять слова, розпізнавання потребує порівняння між вхідним словом та різними словами у словнику. Ефективне розв'язання проблеми лежить у динамічних алгоритмах порівняння, метою якого є обчислення оптимальної послідовності трансформації часу між двома часовими рядами. Шлях трансформації або деформації – це відстань між порівнянням двох звукових хвиль. Чим менший шлях деформації утворюється, тим дві хвилі можна вважати однаковими. Алгоритми такого типу називаються динамічними алгоритмами трансформації часової шкали (DTW).

Одним із варіантів використання є виявлення звукового патерна того ж типу, наприклад розпізнавання аудіофайлу, який містить слово “hello”. Проте, люди вимовляють одне і те ж слово по-різному, якщо користувач вимовляє слово у набагато повільнішому темпі, наприклад “heeeeeeellooooooo”, знадобиться цей алгоритм, щоб зіставити звуки різної довжини та визначити, що вимовлені слова від одного і того ж користувача.

Алгоритм DTW призначений для вирівнювання двох векторних послідовностей шляхом багаторазового повороту осі часу, доки не буде

знайдено оптимальну відповідність між двома послідовностями. Цей метод працює як лінійне відображення осі для вирівнювання двох сигналів. Припустимо, що у двовимірному просторі (рисунок 2.2.3) є дві векторні послідовності, як у формулі (2.2).

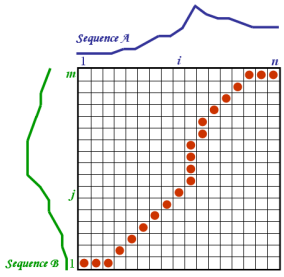


Рисунок 2.3 – Матриця трансформації

$$x = [x_1\ x_2\ \dots\ x_n\ ]\ \text{і}\ y = [y_1\ y_2\ \dots\ y_m] \quad (2.2)$$

Дві послідовності вирівняні з боків: одна зверху ( ), а інша зліва ( ). Обидві послідовності починаються в нижньому лівому куті матриці. У кожній клітинці розміщується міра відстані, порівнюючи відповідні елементи двох послідовностей. Відстань між двома точками обчислюється через евклідову відстань за формулою (2.3).

$$Dist\ (x, y) = |\ x - y\ | = [\ (x_1 - y_1)^2 + (x2 - y2)^2 + \dots + \ (xn - ym)^2]^{\frac{1}{2}} \quad (2.3)$$

Наступним етапом створюється матриця трансформацій (деформацій), де кожен елемент обчислюється за формулою (2.4).

$$D_{i,j} = d_{i,j} + \min\ (D_{i-1}, D_{i-1,j-1}, D_{i,j-1}) \quad (2.4)$$

Останнім кроком визначається оптимальний шлях трансформації (деформації) за формулою (2.5), де K – довжина шляху.

$$max(m,n) \leq K < m + n \tag{2.5}$$

Найкраща відповідність між цими двома послідовностями – це шлях через матрицю, який мінімізує загальну відстань між ними та називається глобальною відстанню. Вона обчислюється шляхом пошуку та проходження всіх можливих маршрутів у сітці, кожен з яких обчислює загальну відстань.

Слід зазначити, що шлях трансформації W містить усі точки обох часових рядів, пересувається не більше ніж на один крок за один раз і не повертається назад до пройденої точки.

DTW-відстань або вартість шляху між двома послідовностями розраховується на основі оптимального шляху трансформації за допомогою наступної формули:

$$DTW(Q,C) = min\left\{\frac{\sum_{k=1}^{K} d(w_k)}{K}\right\} \tag{2.6}$$

де, K у знаменнику використовується для врахування того, що шляхи трансформації можуть бути різної довжини.

Використання евклідової відповідності має суттєвий недолік: якщо два часові ряди однакові, але один з них трохи зміщений у часі (вздовж осі часу), то ряди відрізняються один від одного, що унеможливлює розпізнавання слів. DTW-алгоритм було введено для того, щоб подолати цей недолік та надати наочний вимір відстані між рядами, не звертаючи уваги як на глобальні, так і загальні відстані за шкалою часу [33].

Різний час вирівнювання мови є основною проблемою вимірювання відстані при розпізнаванні слів. Невеликі зсуви призводять до неправильної ідентифікації. DTW є ефективним методом розв'язання проблем вирівнювання часу, тому цей алгоритм використовують для вимірювання подібності шаблону (збіг двох слів). Ілюстрація порівняння DTW з евклідовою відповідністю відображена на рисунку 2.4.

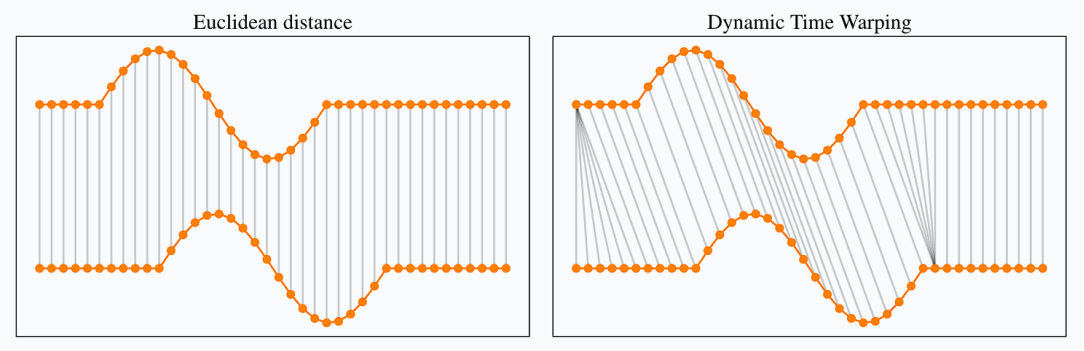


Рисунок 2.4 – Порівняння евклідової відстані і DTW

До переваг DTW-алгоритму відноситься те, що результат порівняння залежить від швидкості відтворення і довжини уявлення двох порівнюваних часових послідовностей. Також цей метод є простим у реалізації та не залежить від кількості класів даних, але недолік у його застосуванні на реальних даних, оскільки потрібне згладжування чи фільтрація.

2.3.3 Алгоритми із застосуванням нейронних мереж

В епоху глибокого навчання нейронні мережі продемонстрували значне поліпшення задачі розпізнавання мови, наприклад згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та останнім часом мережі довгої короткострокової пам'яті (LSTM) досягли високої продуктивності. Нейронні мережі, як прямі, так і рекурентні, можна використовувати лише для кадрової класифікації вхідного аудіо.

RNN виконують обчислення у часовій послідовності, оскільки їх поточний прихований стан залежить від усіх попередніх прихованих станів. Зокрема, вони призначені для моделювання сигналів часових рядів, а також для фіксації довгострокових і короткострокових залежностей між різними часовими кроками вхідних даних. Що стосується програм розпізнавання мовлення, вхідний сигнал передається через RNN для обчислення прихованих і вихідних послідовностей. Одним з головних недоліків простої

форми RNN є те, що вона генерує наступний вихід на основі лише попереднього контексту. RNN обчислюють послідовність векторів за формулами (2.7) і (2.8).

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h); \quad (2.7)$$

$$y_t = W_{hy}h_t + b_y, \quad (2.8)$$

де W – вагові коефіцієнти;

h – вектори прихованого шару;

b – вектори зміщення;

H – нелінійна функція.

Однак, інформація для наступних слів у контексті є так само важливою, як вже сказаних слів, тому замість використання однонаправленої RNN зазвичай обирають двонаправлені (BiRNN), щоб визначити такі дані. BiRNN обробляють вхідні вектори в обох напрямках та зберігають приховані вектори стану для кожного напрямку (рисунок 2.5).

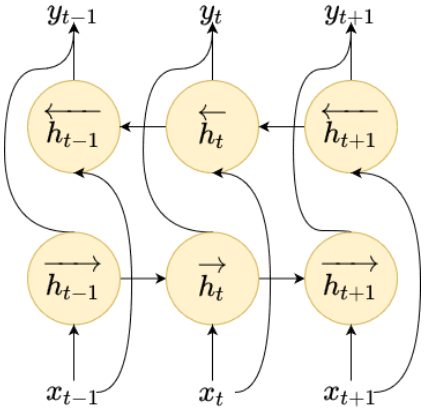


Рисунок 2.5 – Двонаправлений RNN (BiRNN)

Важливу роль у точності розпізнавання мови сучасного ПЗ належить LSTM або GRU – це складні рекурентні нейронні мережі, які здатні до навчання довгострокових залежностей [34]. Замість одного шару нейронної

мережі LSTM містить декілька, а також має можливість скидати дані зі стану комірки, а GRU простіше у реалізації та швидше у навчанні. Для видалення інформації застосовують фільтри.

В останні роки дослідники часто використовують згорткові нейронні мережі (CNN) для виділення ознак. У своїй основі згорткову нейронну мережу можна розглядати як нейронну мережу, що використовує безліч ідентичних копій одного і того ж нейрона. Це дозволяє мережі мати обмежену кількість параметрів при обчисленні великих моделей. Назва мережевої архітектури походить від операцій згортки, оскільки кожен фрагмент зображення множиться на матрицю (ядро) згортки, після чого результат підсумовується та зберігається у відповідній позиції вихідного зображення. Convolution (згортка) містити свій фільтр для кожного кольору або каналу, ядро згортки якого обробляє попередній шар за фрагментами.

RNN застосовують у тих випадках, де потрібно визначити фрагмент або всю послідовність цілком і зробити певний висновок із цього. Наприклад алгоритм може бути успішно застосований у RNN для пошуку зв’язку слів у контексті, наприклад переклад слів, а CNN часто використовується у задачах, пов’язаних з фотографіями або відео, оскільки використовує двовимірні матриці – зображення або кадри з відео [35].

2.3.4 Метод автоматизованого читанням по губах (ALR)

Метод автоматичного читання по губах (ALR) є одним зі способів розв’язання задачі розпізнавання мови із застосуванням SSI-підходу. Особливо у шумних середовищах візуальні сигнали можуть видаляти надмірну або доповнювати мовленнєву інформацію, скорочувати час і робоче навантаження людини, а також покращувати автоматичне розпізнавання мови за допомогою візем. Цей метод охоплює багато наукових галузей, такі як розпізнавання образів, комп’ютерний зір, розуміння природної мови та машинне навчання.

Візема – це візуальний опис фонеми у мовленні. Вона визначає положення особи та рота у той чи інший момент промови. Кожна візема відображає розташування рис обличчя для певного набору фонем.

Традиційні ALR системи зазвичай складаються з двох етапів: виділення ознак та класифікація. На першому етапі більшість методів виділення ознак застосовують значення пікселів, отримані з області рота людини, як візуальної інформації. Потім абстрактні ознаки зображення отримуються за допомогою дискретного косинусного перетворення (DCT), дискретного вейвлет-перетворення (переводить сигнал з часового подання до частотно-часового) та аналізу головних компонентів (PCA).

Оскільки зміни між кадрами відеопотоку автоматичного читання по губах безперервні та відбуваються у часових рядах, дослідники використовують мережу довготривалої короткочасної пам’яті (LSTM), яка може знаходити приховану асоціативну інформацію даних часових рядів. Багатошарова структура нейронної мережі з каскадним рівнем прямого зв'язку та LSTM використовується для класифікації на рівні слів.

Також розглядається система LipNet, яка заснована на двосторонній згортковій нейронній мережі, LSTM та об’єднанні ознак. Традиційний метод поєднується з методом глибокого навчання та створюється мережева структура для вилучення просторово-часових характеристик відео з губ. Читання по губах декодує текстовий вміст відповідно до візуальної інформації про рух губ людини, що говорить. Послідовність кроків з розпізнавання слів на відео виглядає наступним чином:

– метод об’єднання рангів використовується для перетворення відеопослідовності губ на зображення та вся інформація про відеокадри зберігається у пам’яті. Цей підхід може зменшити вхідну розмірність глибокої нейронної мережі та за допомогою акустичної моделі для кожного кадру розраховується розподіл ймовірностей сказаних візем;

– двостороння згорткова нейронна мережа обробляє пряме динамічне зображення (DI) та зворотне DI, оскільки зменшує відхилення у різких рухах;

– функція форми зовнішнього вигляду та функція глибини об’єднані, оскільки цей метод комбінування ознак зберігає просторово-часову інформацію та покращує розпізнавання форми губ в залежності від рухів.

Архітектура LipNet складається з двох частин: перша – це виявлення класичних ознак, а друга – вилучення глибоких ознак, як проілюстровано на рисунку 2.6.

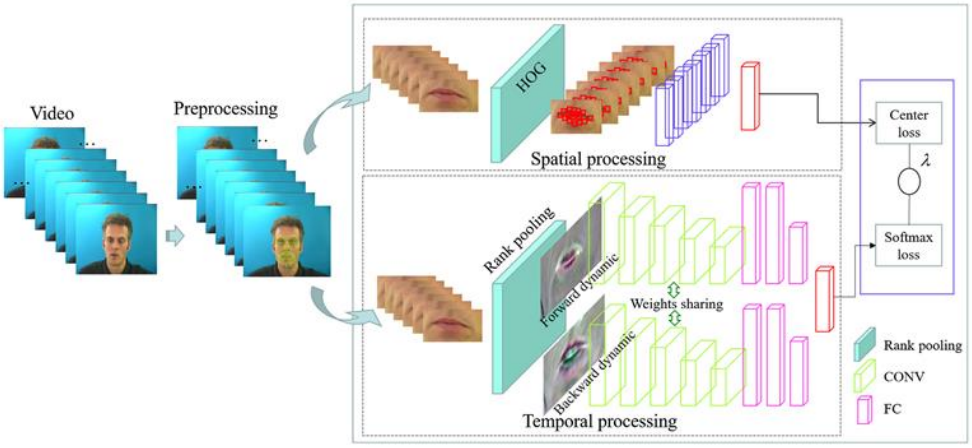


Рисунок 2.6 – Архітектура LipNet

Гістограма спрямованих градієнтів (HOG) використовується для виявлення форми губ на кадрах. Інформаційні зображення, отримані ранговим об’єднанням, відправляються у двосторонню нейронну мережу (BiRNN). Всі виявлені ознаки об’єднуються через спільну функцію оцінки вагів, щоб отримати більш вирішальні ознаки шляхом вивчення згорткового шару для покращення точності [36].

3 РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ З SSI-ПІДХОДОМ

3.1 Реалізація системи на основі ALR

Візуальне розпізнавання мовлення або читання з губ відіграє важливу роль у спілкуванні між людьми, особливо в шумному середовищі та може бути надзвичайно корисним для людей із вадами слуху, тому було обрано технологію ALR для розпізнавання команд з використанням камери. Для ідентифікації слова або речення, система має бути навчена за допомогою даних, зібраних певною мовою та словником, але для ALR замість фонем застосовують віземи.

ALR або автоматизове читання по губах – це декодування тексту за рухом рота мовця. Машинне зчитування з губ є складним, оскільки вимагає вилучення просторово-часових характеристик із відео (оскільки положення та рух губ важливі). Також ускладнює процес розпізнавання візуальних ознак язика та зубів, оскільки у великій кількості випадків вони приховані через закритий рот, тому їх важко розпізнати без контексту.

Також останнім часом відбувся сплеск наскрізних підходів до глибокого навчання для читання з губ, які зосереджені на прогнозуванні на рівні слів із використанням комбінації згорткових і рекурентних мереж [36].

У запропонованій системі (рисунок 3.1) потрібно перетворити відео у послідовність кадрів, які містять зображення губ. Наступним кроком ці кадри застосовуються як вхідні дані для згорткової нейронної мережі, яка до цього була навчена на аналогічних даних. Потім дані з CNN проходять через повноз’єднані шари для формування вхідного вектора візем до LSTM. Вихід одного шару стає входом наступного рекурентного шару. Останнім кроком вектор розподілу ймовірностей потенційних візем декодується в LSTM, а у результаті формується послідовність символів, які об’єднуються у слова.

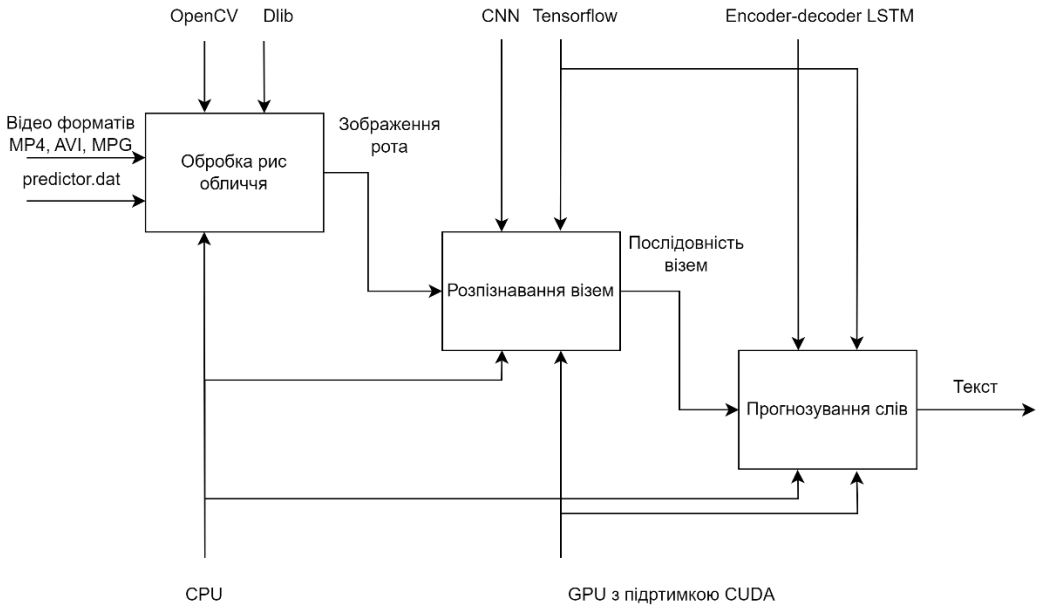


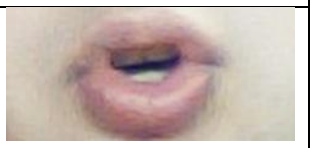
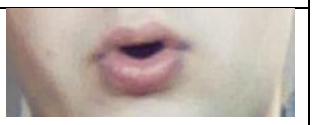
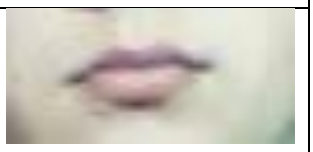
Рисунок 3.1 – IDEF0 нотація запропонованої системи розпізнавання

Для розв’язання задачі, потрібно визначити які слова чи фрази вимовляються з фіксованого набору відомих. Компоненти системи застосовують послідовність зображень як вхідні дані, а результатом - слова. У таблиці 3.1 відображено 11 візем і стан мовчання, які відповідають фонемам англійської мови та можна запрограмувати у вигляді словника, оскільки групи фонем не відрізняються за візуальними ознаками.

Таблиця 3.1 – Віземи та фонеми англійської мови з власними прикладами

Приголосні			Голосні		
Візема	Фонема	Приклад	Візема	Фонема	Приклад
$V_{J,C,H}$	/dʒ/ /tʃ/ /ʃ/ /ʒ/		V_A	/ɑ:/ /aʊ/ /aɪ/ /ʌ/	
$V_{P,M,B}$	/p/ /b/ /m/		V_E	/e/ /eɪ/ /æ/	

Продовження таблиці 3.1

Приголосні			Голосні		
Візема	Фонема	Приклад	Візема	Фонема	Приклад
$V_{F,V}$	/f/ /v/		V_I	/i:/ /ɪ/	
$V_{D,T,S}$	/d/ /t/ /s/ /z/ /θ/ /ð/		V_O	/ɔ:/ /ɔɪ/ /əʊ/	
$V_{R,W}$	/r/ /w/		V_U	/ʊ/ /u:/	
$V_{G,K,N}$	/g/ /k/ /n/ /ŋ/ /l/ /y/ /h/		Silent		

Точність розпізнавання візем буде низькою у випадках, коли відмінною рисою є положення язика, наприклад $V_{D,T,S}$ та $V_{G,K,N}$, для чого необхідно застосовувати середовище з гарним освітленням, а розпізнавання у темряві буде неможливе. Для таких умов краще використовути камеру глибини, оскільки світло не впливає на дані та існують проєкти, які використовують цей пристрій, наприклад Microsoft Kinect [3].

3.2 Розпізнавання візем

Базове виявлення візем полягає в аналізі геометрії обличчя. У разі відкритого рота відстань між куточками рота збільшується. Незважаючи на те, що люди мають різні розміри рота, можна нормалізувати цей показник,

розділивши його на відстань між щелепами та отримати загальне співвідношення, яке можна використовувати для різних облич.

Кожне зображення містить велику кількість вихідної інформації, яка не використовується у розпізнаванні мови. Тому потрібно обробляти кожне зображення та виділяти область губ. У цій роботі як основний алгоритм виділення ознак було використано функції бібліотеки Dlib. Її за точністю та швидкістю можна порівняти з іншими рішеннями: MTCNN, Openface, LFW landmarks, хоча в останньому ПЗ процес обробки зображення буде швидше, а точність гірше без застосування модифікацій. Також код існує у загальному доступі, що є важливим для редагування алгоритмів або зміни параметрів для розпізнавання.

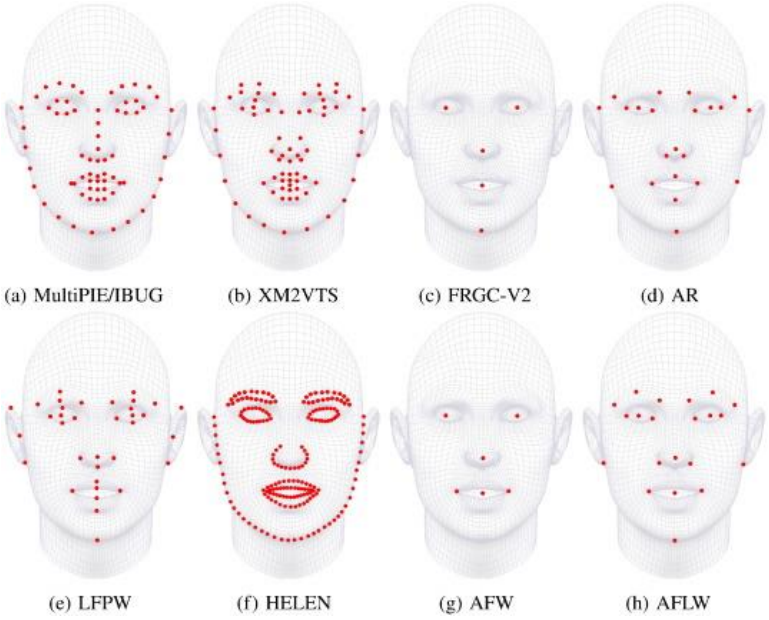


Рисунок 3.2 – Приклади розмітки рис обличчя

Детектор рис обличчя у dlib розроблено з використанням класичної функції гістограми орієнтованих градієнтів (HOG) у поєднанні з лінійним класифікатором, пірамідою зображення і схемою виявлення ковзного вікна.

Гістограма орієнтованого градієнта – це алгоритм обробки зображення, який виконує функцію виділення ознак. Dlib містить інформацію про

розмітку крапок у контурі обличчя та застосовує їх для вхідного кадру, а у вихідному позначає ці крапки, якщо на зображені є рот, очі або інші риси обличчя.

Для розпізнавання контурів обличчя, у тому числі губ, застосовується shape_predictor_68_face_landmarks.dat, яка навчена на колекції зображень iBUG 300W (рисунок 3.2). Також можна використати інші файли, наприклад на основі HELEN, оскільки в ній існує велика кількість крапок, які виділяють верхню та нижню губу, а також відкритий рот. Використовуючи ці ознаки, алгоритм отримує відцентровані по губах зображення розміром 100 × 50 пікселів, чого буде достатньо для подальшої обробки нейронними мережами. Також область з виявленими губами збільшується на 10 або більше пікселів з кожного боку, щоб губи не вийшли обрізаними.

Лістинг 3.1 – Фрагмент із застосуванням dlib і OpenCV

```
detector = dlib.get_frontal_face_detector() predictor =
dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')

image = cv2.imread("depth_002.png")
plt.imshow(image)

image = imutils.resize(image, width=100, height=50)
plt.figure()

cv2.imwrite('test.jpg',image)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
rects = detector(gray, 1)

for (i,rect) in enumerate(rects):
     shape = predictor(gray, rect)
     shape = face_utils.shape_to_np(shape)
```

Кожен кадр відео змінюється у градації сірого, оскільки колір не відіграє суттєвої ролі при виявленні обличчя на зображені, наявність якого негативно впливає на швидкість роботи алгоритму. Для цього потрібно використати метод конвертації кольору з бібліотеки OpenCV і використовувати чорно-біле зображення для наступних модулів.

3.3 Застосування нейронних мереж

Для створення системи розпізнавання мови застосовується архітектура на основі нейронних мереж для обробки мови, яка відображає послідовності відеофрагментів змінної довжини у текстові послідовності.

3.3.1 Застосування CNN

Вхідні дані у вигляді відеофайлів застосовують для розв’язання задачі аналізу обличчя за допомогою згорткових нейронних мереж (CNN), а саме для розпізнавання рис обличчя та візем.

Згорткові нейронні мережі (CNN) – це архітектура штучних нейронних мереж, націлена на ефективне розпізнавання зображень (рисунок 3.3). Ядро згортки – це матриця ваг невеликого розміру, яку "рухають" всьому оброблюваному шару вхідного зображення, формуючи після кожного зсуву сигнал активації для наступного нейрона шару з аналогічною позицією. Для організації згорткової нейронної мережі застосовується:

– згортка (CONV);
– агрегування (POOL);
– функція активації ReLU;
– повноз’єднаний шар (FC).

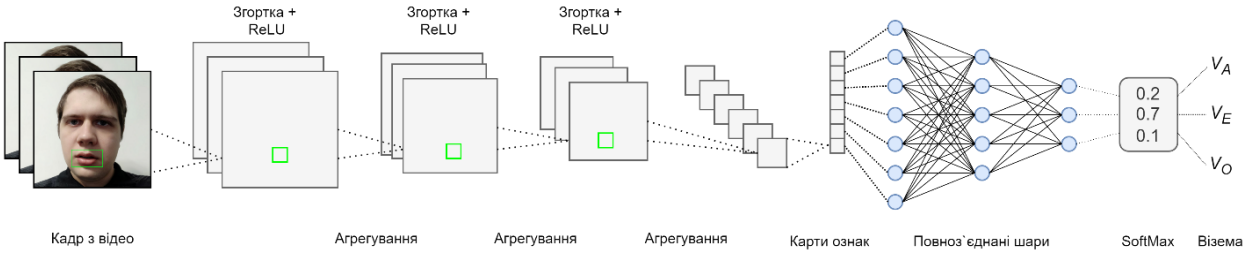


Рисунок 3.3 – Архітектура CNN для розпізнавання візем

Операція згортки зображення – це операція між матрицею зображення і ядром згортки (фільтром), коли кожен елемент (піксель) у вихідному зображенні є сумою добутку значення елемента ядра на значення відповідного елемента матриці вхідного зображення. Приклад операції згортки наведено у лістингу 3.2.

Процес згортки є ітеративним. Спочатку до ділянки вхідного зображення застосовується фільтр і записується вихідне значення. Потім фільтр зміщується на одну позицію, якщо крок дорівнює 1 або на кілька позицій, якщо для кроку встановлено більше число і процес повторюється до тих пір, поки не буде завершена згорнута функція.

Лістинг 3.2 – Приклад додавання шару згортки у модель

```
model.add(Conv2D(filters=64,kernel_size=(3,3),strides=(1,1),
padding='same',activation='relu',name='2D-Convolutional-Layer'))
```

Потрібно налаштовувати кілька згорткових шарів для покращення мережі. Переваги виникають через те, що наступні згорткові шари виявляють додаткову складність у зображенні. Перший рівень у глибокій мережі згортання (DCN) має тенденцію знаходити ознаки низького рівня (наприклад, вертикальні, горизонтальні, діагональні лінії…). Глибокі шари можуть ідентифікувати характеристики вищого рівня, такі як складні форми, які мають візуальні ознаки при вимові певних звуків, наприклад прихована нижня губа для фонем /f/ або /v/.

Зазвичай шар об’єднання додається після шару згортки. Його мета – зменшити розмір згорнутих об'єктів, підвищивши ефективність обчислень. Також це може допомогти усунути шум, зберігши найсильніші активації.

Окрім цього, застосовується шар виключення (dropout), який випадково встановлює вхідні одиниці в 0 в залежності від наданої швидкості. Це означає, що випадкові вхідні дані (ознак/вузлів) будуть обнулені для запобігання перенавчання мережі та значно підвищує швидкість навчання.

3.3.2 Застосування LSTM

Рекурентна нейронна мережа зазвичай використовується для пошуку регулярного шаблону, наприклад слів у реченні, а мережа LSTM пристосована до навчання на задачах класифікації, обробки та прогнозування часових рядів у випадках, коли важливі події розділені тимчасовими затримками з невизначеною тривалістю. Щоб краще виявляти та використовувати довгострокові залежності від даних послідовності (відео або аудіо), комірка пам'яті запам'ятовує пов'язану інформацію, яку потрібно зберігати у довгій послідовності, та видаляє частину непотрібної інформації.

Будь-яка рекурентна нейронна мережа має форму ланцюжка модулів нейронної мережі, що повторюються. У звичайній RNN структура одного модуля дуже проста, наприклад, він може бути одним шаром з функцією активації гіперболічного тангенса, а для LSTM мають інший вигляд. На рисунку 3.4 зображено ключовий компонент LSTM - це стан комірки (cell state) - горизонтальна лінія, що проходить по верхній частині LSTM unit та бере участь у лінійних перетвореннях і формує результат.

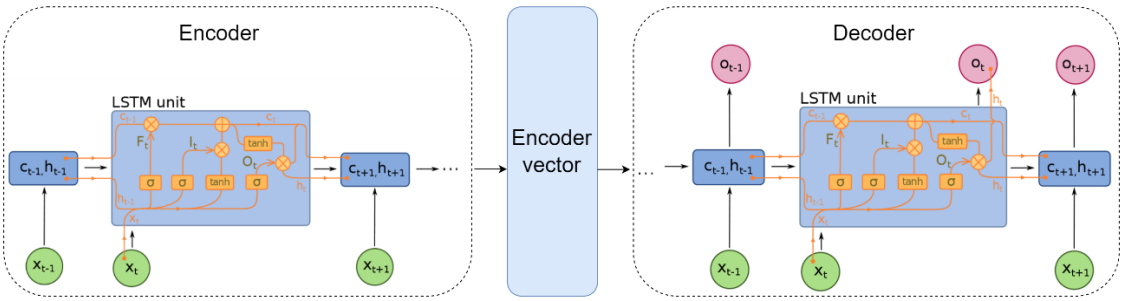


Рисунок 3.4 – Схема encoder-decoder LSTM

Рекурентні нейронні мережі RNN працюють шляхом ітерованого оновлення прихованого стану h, який є вектором, що може мати довільний розмір. Також застосовують зсуви для h_t та y_t. Варто враховувати, що на будь-якому заданому етапі t:

- наступний прихований стан h_t розраховується за допомогою попереднього h_{t-1} та наступним входом x_t;

- наступний вихід y_t розраховується через h_t.

У цьому випадку рекурентна нейронна мережа складається з трьох параметрів ваги та двох зсувів, приклад такої комірки наведено у лістингу 3.3. Для ваги використовується матричне множення, після чого вектори вносяться до кінцевого результату. Потім застосовується гіперболічна функція або сигмоїда для активації.

Лістинг 3.3 – Модель комірки LSTM

```
def RNN(x, weights, biases):
    x = tf.reshape(x, [-1, n_input])
    x = tf.split(x,n_input,1)
    rnn_cell = rnn.BasicLSTMCell(n_hidden)
 outputs, states = rnn.static_rnn(rnn_cell, x)
    return tf.matmul(outputs[-1], weights['out']) +
biases['out']
```

У цій роботі запропоновано реалізувати архітектуру encoder-decoder для LSTM, оскільки існує процес перетворення з послідовності на послідовність. На вхід decoder поступають віземи з попереднього блока розпізнавання, а результатом encoder буде послідовність символів.

Технічно вхідні дані encoder-decoder LSTM можуть використовувати лише дійсні числа. Спосіб перетворення фонеми або віземи у число полягає у тому, щоб позначити кожному елементу унікальне ціле число в залежності від частоти його появи, тому для підготовки даних для навчання потрібно створити два словники, де кожний ключ – це одна з 11 візем і значення має бути числом, а також навпаки для декодування результату LSTM. Процес декодування полягає у тому, що потрібно подати вектори стану і послідовність символів у декодер, щоб зробити прогнози для наступного символу і повторювати процес, доки не з’явиться символ кінця послідовності або досягне останнього символу у послідовності.

3.4 Завантаження файлів з набору GRID

При побудові передбачуваних моделей вихідні дані зазвичай розбиваються на навчальну та контрольну вибірки. Корпус GRID складається із відео та аудіоданих, записаних 18 чоловіками та 16 жінками у контрольованому середовищі з роздільною здатністю відео 720 × 576 пікселів і 25 кадрами в секунду. Також кожна пара відео та аудіо супроводжується відповідною транскрипцією, приклади яких наведено у таблиці 3.2.

Таблиця 3.2 – Набір даних GRID

Команда	Колір	Прийменник	Літера	Число	Дієслово
bin	blue	at	a-z (без w)	0-9	again
lay	green	by			now
place	red	in			place
set	white	with			soon

Лістинг 3.4 – Фрагмент функції завантаження кадрів відео

```
reader = cv2.VideoCapture(path)
ok, frame = reader.read()
if not ok:
     break
else:
     frames.append(frame[...,::-1])
return np.array(frames)
```

На початку цього проєкту потрібно встановити шлях до набору даних, щоб система могла завантажити файл. У наборі даних GRID існує 35 директорій, які потрібно завантажити, але файли під номером 21 пропускається, оскільки відео було пошкоджено через технічні причини. Після завантаження набору даних використовуються функції з бібліотеки OpenCV для перетворення відео на кадр, приклад наведено у лістингу 3.4. Також можна застосувати аналогічну функцію для формування результатів.

3.5 Використання CuDNN

Для прискорення можна застосувати ядра графічної карти з підтримкою CUDA, що значно зменшить час навчання нейронної мережі до обробки у реальному часі. Оскільки обрано мову програмування Python, то можна застосувати бібліотеки з підтримкою GPGPU CUDA – Tensorflow і Keras для збирання високопродуктивних шарів і для побудови моделей.

Бібліотека глибоких нейронних мереж CuDNN створена для прискорення глибоких нейронних мереж, оскільки розпаралелює за допомогою Tensor Core для популярних згорткових мереж, наприклад 2D, 3D, згрупованих та інших. Також цю бібліотеку можна застосувати для CNN і LSTM, оскільки прискорює операції об’єднання, копіювання, поелементного обчислення гіперболічного тангенса, матричного множення, додавання або інших математичних функцій.

Модель з підтримкою CuDNN за замовчуванням не використовується у Tensorflow, тому потрібно перевірити tf.device, який вказує на пристрій, що використовується та перемикнути з CPU на GPU.

Також відомо, що у TensorFlow 2.0 вбудовані шари LSTM і GRU придатні для використання ядер CuDNN за замовченням, якщо доступний графічний процесор.

У Keras існує три пакети для роботи з RNN:

– SimpleRNN – повноз’єднана RNN;

– GRU - керовані рекурентні блоки;

– LSTM - довга короткострокова пам’ять.

Для LSTM можна застосувати вкладені структури, оскільки вони зберігають більше інформації за одиницю часу. Наприклад, відеофрейм може містити аудіо та відео одночасно або складатись з декількох векторів для кожного кольору. Такі структури застосовують для модифікації комірки RNN або створення власної архітектури з різних комірок, а приклад ініціалізації наведено у лістингу 3.5.

Лістинг 3.5 – Ініціалізація модифікованої комірки RNN

```
NestedInput = collections.namedtuple('NestedInput', ['feature1',
'feature2'])
NestedState = collections.namedtuple('NestedState', ['state1',
'state2'])

class NestedCell(tf.keras.layers.Layer):
  def __init__(self, unit1, unit2, unit3, **kwargs):
    self.unit1 = unit1
    self.unit2 = unit2
    self.unit3 = unit3

    self.state_size = NestedState(state1=unit1,
state2=tf.TensorShape([unit2, unit3]))
    self.output_size = (unit1, tf.TensorShape([unit2, unit3]))

    super(NestedCell, self).__init__(**kwargs)
```

Бібліотека для вбудованих шарів RNN надає API на рівні комірки. На відміну від шарів RNN, які обробляють цілі пакети вхідних векторів, комірка RNN обробляє лише одну за ітерацію. Оскільки комірка розташована всередині циклу, то обертання її шаром tf.keras.layers.RNN надає можливість обробляти пакети послідовностей, наприклад RNN(LSTMCell(10)), яка відповідає LSTM(10). Приклад застосування LSTM у Keras наведено у лістингу 3.6. Абстракція комірок разом із загальним класом tf.keras.layers.RNN дозволяє редагувати стандартну архітектуру RNN.

Лістинг 3.6 – Приклад застосування LSTM у Keras

```
model = Sequential()

model.add(Embedding(max_features, 256, input_length=maxlen))
model.add(LSTM(output_dim=128, activation='sigmoid',
inner_activation='hard_sigmoid'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
```

Зазвичай, внутрішній стан шару RNN скидається при кожному новому пакеті даних (дані не залежать від минулого стану). Шар підтримуватиме поточний стан лише на час обробки цього елемента. Однак, якщо існують довгі послідовності, наприклад у декілька сотень елементів, краще їх розбити на коротші та поступово передавати їх до RNN шару без скидання стану шару. Також у LSTM використана функція TimeDistributed для кодування вмісту даних. Таким чином, шар може зберігати інформацію про всю послідовність, хоча він обробляє лише одну підпослідовність за раз.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Метрики

У цій роботі застосовуються стандартні метрики оцінки якості розпізнавання мови [37] для оцінки створеної системи, такі як послівна точність розпізнавання (WRR) та послівна помилка розпізнавання (WER). Формула для WRR виглядає наступним чином:

$$WRR = \frac{W_N - W_s - W_D - W_I}{W_N}, \quad (4.1)$$

де W_s – кількість слів або речень, замінених на інше;

W_D – кількість слів або речень, що випали;

W_I – кількість вставлених слів або речень;

W_N – загальна кількість слів у фразі або речень, що розпізнається.

Також для визначення помилок потрібно розрахувати WER, який часто застосовується у задачах розпізнаванні мови, машинного перекладу та обчислюється за формулами 4.2 або 4.3.

$$WER = 1 - WRR \quad (4.2)$$

$$WER = \frac{W_s + W_D + W_I}{W_N} \quad (4.3)$$

Також для порівняння з іншими проєктами можна застосувати частоту помилок символів (CER), що обчислюється за формулою 4.4 і відповідає попередній, але замість слів застосовуються символи:

$$CER = \frac{C_s + C_D + C_I}{C_N}, \quad (4.4)$$

4.2 Характеристики апаратного та програмного забезпечення

Для тестування систем розпізнавання голосових команд на основі аудіо або відео використовується ноутбук, характеристики якого:

- процесор: Intel Core i7 6700HQ, 8 потоків, частота 3.1 ГГц;
- оперативна пам’ять: 24 Гб, DDR4, частота 2133 МГц;
- відеокарта: NVIDIA GeForce GT 960m, Compute Capability – 5.0, кількість мультипроцесорів – 5, об’єм відеопам’яті – 2 Гб, версія драйверів - 440.33, версія CUDA SDK – 10.2.89;
- операційна система: Linux 5.15, дистрибутив Ubuntu 22.04.1 LTS.

Пристрій застосовується як для отримання даних, так і для подальшої обробки, але на початковому етапі реалізації цього проєкту використано як пристрій введення та первинної обробки відео одноплатний комп’ютер Raspberry Pi 4, а у ролі сервера з формуванням результату – ноутбук. Причиною відмови від цієї реалізації – це недостатня потужність для обробки відео та роботи з нейронними мережами (CNN, RNN), що було виявлено на етапі виявлення візем та в аналогічній системі [18].

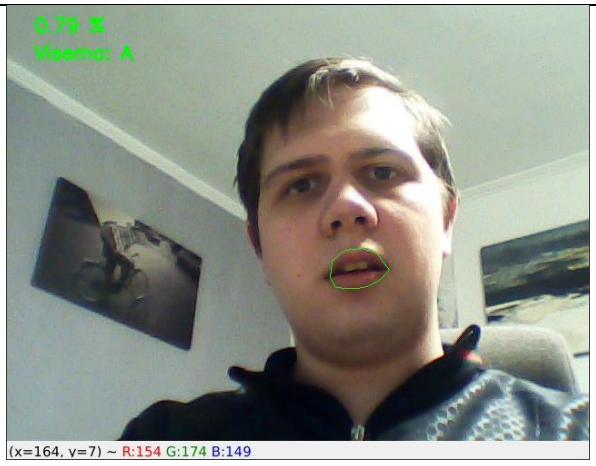
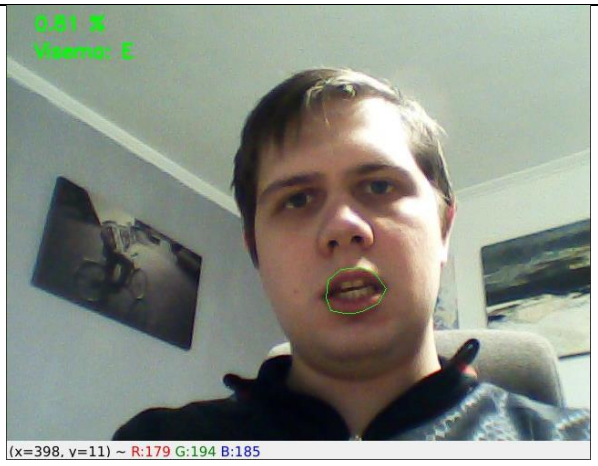
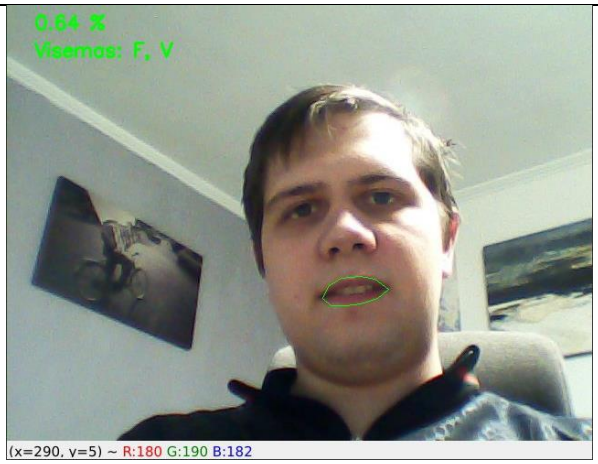
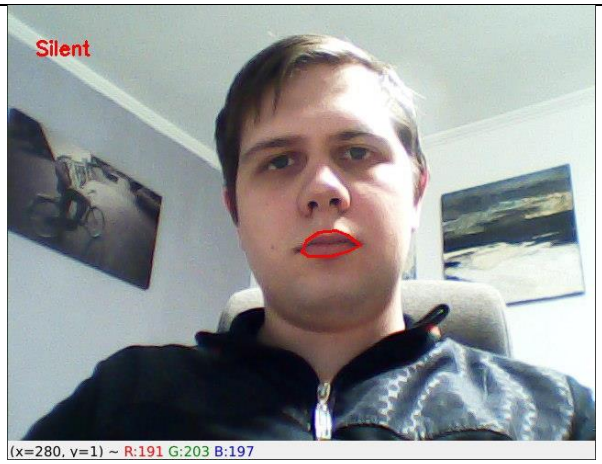
4.3 Тестування власної системи на основі ALR

Перед початком тестування потрібно завантажити колекцію відео, що має звук та текст у вигляді окремого файлу субтитрів. Також набір файлів має бути поширеним для порівняння з іншими проєктами без SSI-підходу.

Навчальна вибірка складається з перших 30 архівів корпусу GRID без 22 за порядком (немає відео через технічні причини), а тестова вибірка містила 3 останніх, диктори на яких не говорили у навчальній вибірці. Також було використано генератор псевдовипадкових чисел для обрання файлу з тестової вибірки. Для перевірки працездатності системи було протестовано визначення рота на відео (таблиця 4.1) та обрізання його на послідовність кадрів розміром 150 × 100.

Наступним етапом потрібно використати навчальну вибірку для тренування нейронних мереж, для чого у циклі послідовно обробляються кілька сотень відео з кожної директорії. Крім цього, використовувалася відеокарта для прискорення процесу навчання нейронних мереж, що значно скоротило час обробки у кілька десятків разів. До цього було протестовано на основі CPU першу директорію, що зайняло близько однієї години. Також був варіант використовувати відео в низькій роздільній здатності та відповідно обрізати кадр з ротом до розміру 75x50, але в аналогічних проєктах це призводило до погіршення точності розпізнавання [36]. У результаті створюється текстовий файл команд для кожного відео.

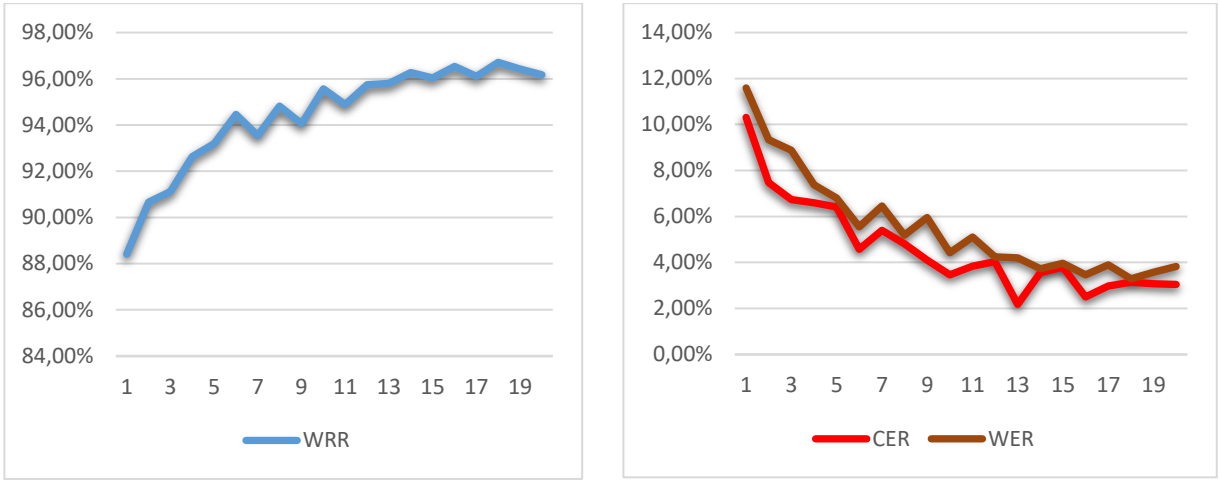
Таблиця 4.1 – Результати розпізнавання візем

	
Візема V_A	Візема V_E
	
Візема $V_{F,V}$	Мовчання

Для тестування системи було обрано 20 епох, оскільки краще підлаштовується під дані, переходячи з погано навченого стану до оптимального. Розраховано показники точності розпізнавання слів, помилок у словах та символах і середнє арифметичне. Загальна кількість для навчання в 1 епосі відповідає кількості всіх відео з диктором, наприклад 489 або 492.

Таблиця 4.2 – Показники CER, WER та WRR для 20 епох

Епоха	CER	WER	WRR
1	10,31%	11,59%	88,41%
2	7,48%	9,34%	90,66%
3	6,73%	8,88%	91,12%
4	6,59%	7,37%	92,63%
5	6,41%	6,8%	93,2%
6	4,58%	5,55%	94,45%
7	5,4%	6,46%	93,54%
8	4,82%	5,19%	94,81%
9	4,1%	5,95%	94,05%
10	3,47%	4,43%	95,57%
11	3,84%	5,11%	94,89%
12	4,03%	4,25%	95,75%
13	2,18%	4,2%	95,8%
14	3,56%	3,73%	96,27%
15	3,8%	3,96%	96,04%
16	2,51%	3,47%	96,53%
17	2,98%	3,9%	96,1%
18	3,15%	3,29%	96,71%
19	3,07%	3,58%	96,42%
20	3,04%	3,82%	96,18%
Середнє	4,6%	5,54%	94,46%



а) б)

Рисунок 4.1 – Залежність показників від 20 епох: а) послівна точність розпізнавання; б) помилка розпізнавання у символах та у словах

Велика кількість епох негативно впливає на показники точності системи та приводить до перенавчання мережі, що помітно в останніх двох результатах таблиці 4.1, тому збільшення кількості епох погіршує результати. При використанні рекурентної нейронної мережі відбувається послідовна подача вхідних даних з кожного кадру, тому система не навчиться правильно визначати слово за першими кадрами та точність у перших епохах гірше, чим у наступних.

Також було перевірено кількість помилок у символах та словах, на основі чого було виявлено значний вплив неправильного розпізнавання слів на точність системи (рисунок 4.1). Можливо, це пов’язано з проблемою розпізнавання слів на основі візем у модулі, який використовує рекурентну нейронну мережу та потребує змін для підвищення точності, наприклад заміна LSTM на двонаправлені GRU або інші нейронні мережі. Велика кількість помилок була в омофонах, наприклад літеру v замінено f або b на p.

Середнє значення точності розпізнавання у словах дорівнює 94,46%, що є гарним показником для сучасних систем розпізнавання на основі інтерфейсів безмовного доступу та можна порівняти з аналогічними проєктами або ПЗ, яке розпізнає слова на тільки основі аудіо.

4.4 Тестування систем з шумом на основі ALR, AV-ASR та ASR

Реалізація на основі ALR вміє розпізнавати команди тільки з відео, тому для порівняння у різних звукових умовах застосовуються інші підходи: класичний ASR та поєднання ASR та ALR – AV-ASR. Тестова вибірка використовується з минулих експериментів, а також завантажується окремий набір даних корпусу GRID без відео для навчання LipNet на основі AV-ASR і тестування CMU Sphinx, який не може працювати з відеофайлами.

Також для тестування систем було створено декілька умов, які відрізняються перешкодами (додавання звуку гучністю 10 дБ) для розпізнавання голосу (таблиця 4.3):

- білий шум;
- сірий шум;
- шум натовпу А – одночасна розмова україномовних дикторів;
- шум натовпу Б – одночасна розмова англомовних дикторів;
- відсутність звуку.

Таблиця 4.3 – Показники систем у різних умовах

Умови	ALR			AV-ASR			ASR		
	Власна система			LipNet			CMU Sphinx		
	CER	WER	WRR	CER	WER	WRR	CER	WER	WRR
Білий шум	4,34	5,79	94,21	13,04	27,4	72,6	24,9	45,59	54,41
Сірий шум	5,12	6,06	93,94	11,38	21,46	78,54	22,17	37,42	62,58
Шум натовпу А	4,52	5,52	94,48	13,74	28,18	71,82	23,6	41,11	58,89
Шум натовпу Б	4,43	5,65	94,35	22,2	39,52	60,48	28,04	49,86	50,14
Без звуку	4,58	5,73	94,27	8,34	15,8	84,2	-	-	-
Без відео	-	-	-	7,35	13,1	86,9	5,1	10,51	89,49
Оригінал	4,6	5,54	94,46	3,1	5,8	94,2	9,67	11,14	88,86

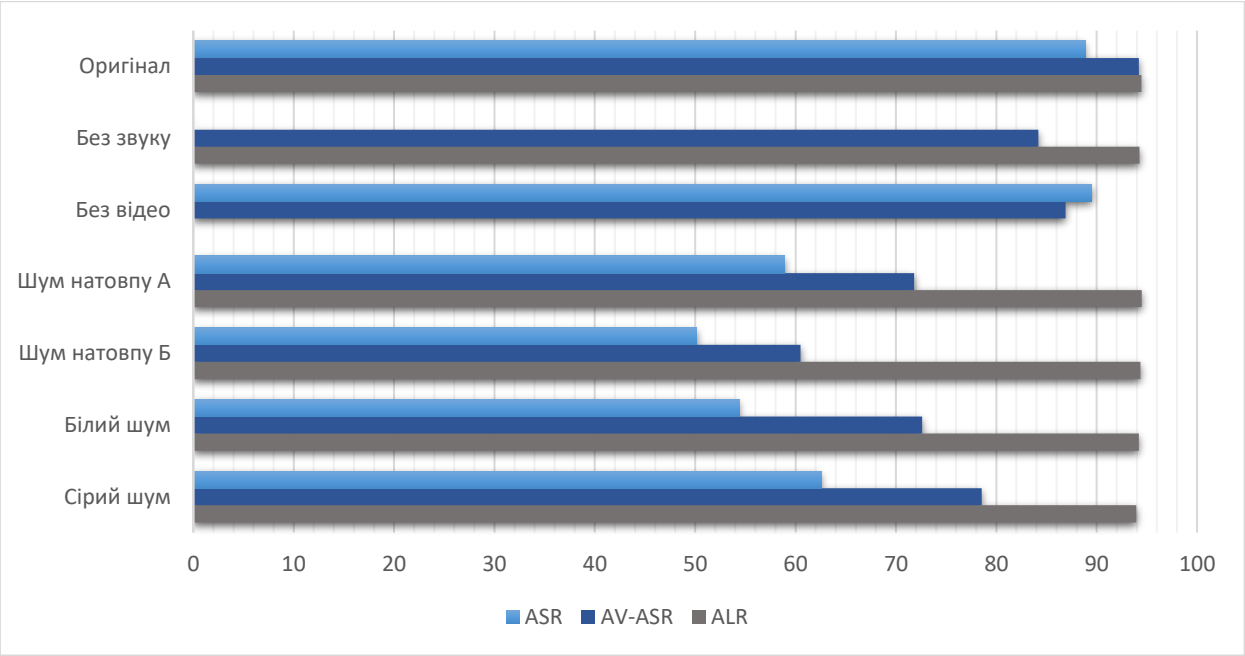


Рисунок 4.2 – Результати WRR у різних умовах

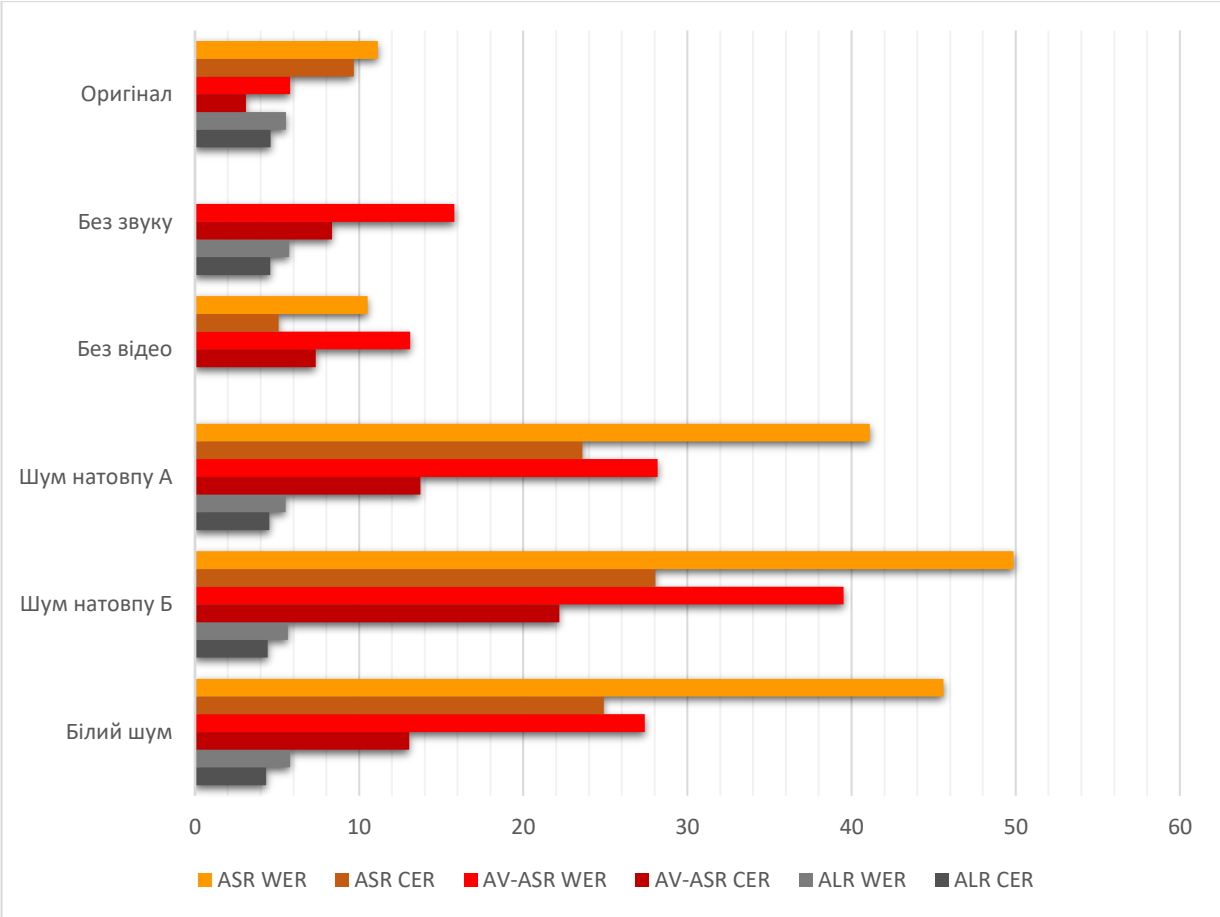


Рисунок 4.3 – Результати CER і WER у різних умовах

За результатами у таблиці 4.3 всі системи розпізнавання команд гарно працюють для оригіналу на основі останніх трьох колекцій корпусу GRID, але для CMU Sphinx (ASR) показники точності та помилок гірше, ніж для інших. Похибка розпізнавання у різних умовах становить 0,54% для власної ALR системи у 7 експериментах та майже не відрізняється від проєкту LipNet, який додатково обробляє дані з аудіо для оригіналу без шумів. Також не було виміряно показники для ALR в умовах без відео та для ASR без аудіо, оскільки у таких проєктах неможливе використання цих даних.

Показники помилок для ASR значно збільшуються в умовах додавання будь-якого шуму, особливо англомовних дикторів (шум натовпу Б), а також для LipNet, який застосовує дані з аудіо та поєднує розпізнавання фонем та візем. Причиною погіршення точності є виявлення системою звуків та слів, які раніше були розпізнані у навчальній вибірці, наприклад at, zero, але обробка візем у LipNet покращує результат, ніж CMU Sphinx у цих умовах. Також ASR має кращі показники помилок у символах та словах, ніж LipNet в умовах, коли немає відео. Можливо це пов’язано з тим, що система навчена на відео і чекає кадрів з людьми, які говорять.

Найкращий результат кількості помилок у символах у LipNet, який якісно розпізнав фонеми (рисунок 4.3), які вплинули на підсумковий результат, але додавання шуму погіршило точність в інших звукових оточеннях, але ПЗ вміє працювати в умовах, коли немає відео, що є корисним для розпізнавання у випадках, коли губи спікера не видно у темряві або камера перестала працювати.

Для реалізованої системи на основі ALR будь-які умови не впливають, оскільки змінюється звуковий сигнал, який не є вхідним у розпізнаванні фонем або цілих слів, на відміну від зміни обличчя на кадрах. Це може бути перевагою над іншими системами, оскільки у проведених експериментах сигнал з шумом погіршує точність розпізнавання для ASR та AV-ASR, але існують роботи, в яких покращили архітектуру LipNet для таких випадків [36], а для ASR застосовуються алгоритми зниження шуму.

ВИСНОВКИ

У даній роботі було створено систему розпізнавання команд з SSI-підходом та проведено ряд експериментів над сучасними рішеннями на основі інтерфейсів безмовного доступу (ALR та AV-ASR) та без них (ASR) у різних звукових оточеннях. Головною метою SSI-підходу є покращення точності розпізнавання мови у таких випадках, коли аудіо немає, наприклад на великій відстані від того, хто говорить, або у шумному оточенні.

За результатами експериментів, система розпізнавання на основі ALR або AV-ASR працює краще за ПЗ, яке використовує дані з аудіо в умовах додавання шуму. Наприклад точність розпізнавання слів на 18,19% більше для AV-ASR, ніж для системи, яка розпізнає лише звуковий сигнал з додаванням білого шуму гучністю 10 дБ.

Розробка системи розпізнавання команд з SSI-підходом – це не проста задача, оскільки вимагає застосування великої кількості алгоритмів для обробки звуку або визначення візем на кадрах відеозапису, прогнозування слів, покращення продуктивності обробки із застосуванням технологій паралелізму тощо. Архітектура рекурентних та згорткових нейронних мереж для визначення слів у контексті з відео та продуктивність сучасних комп’ютерних систем спрощують розв’язання задачі та можуть замінити системи розпізнавання на основі аудіо.

Для подальшого розвитку роботи, пропоную реалізувати її на основі інших підходів, наприклад камери глибини, системи із кількох камер навколо людини або застосуванням відео з більшою частотою кадрів. Також не розв’язано задачу в умовах недостатньої яскравості або положення обличчя під іншим кутом. Окрім цього, не були протестовані інші бібліотеки (Openface, OpenVINO Toolkit тощо) та інші нейронні мережі (GRU замість LSTM, MTCNN для виявлення області рота), а також інші корпуси (MIRACL-VC1, OuluVS2).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Терещенко О. В., Барковська О.Ю. Аналіз впливу SSI-підходу на продуктивність розпізнавання голосових команд. Матеріали десятої міжнародної науково-технічної конференції «Проблеми інформатизації» (24 –25 листопада 2022 року).

2. Wang H. H. Speech Recorder and Translator using Google Cloud Speech-to-Text and Translation. Journal of IT in Asia. 2021. Vol. 9, no. 1. P. 11–28. URL: https://doi.org/10.33736/jita.2815.2021

3. Evaluation of a Silent Speech Interface Based on Magnetic Sensing and Deep Learning for a Phonetically Rich Vocabulary / J. A. Gonzalez et al. Interspeech 2017. ISCA, 2017. URL: i

4. Galatas G., Potamianos G., Makedon F. Audio-visual speech recognition using depth information from the Kinect in noisy video conditions. the 5th International Conference, Heraklion, Crete, Greece, 6–8 June 2012. New York, New York, USA, 2012. URL: https://doi.org/10.1145/2413097.2413100

5. Modelling and Verification of Context-Aware Intelligent Assistive Formalism / S. Yousaf et al. Computers, Materials & Continua. 2022. Vol. 71, no. 2. P. 3355–3373. URL: https://doi.org/10.32604/cmc.2022.023019

6. Van Deemter K. Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology, and Product Evaluation. Journal of Pragmatics. 2003. Vol. 35, no. 8. P. 1281–1285. URL: https://doi.org/10.1016/s0378-2166(03)00072-9

7. Hoboken, NJ. Brief History of Automatic Speech Recognition. Speech and Audio Signal Processing, USA, 2011. P. 40–58. URL: https://doi.org/10.1002/9781118142882.ch4

8. Najkar N., Razzazi F., Sameti H. A novel approach to HMM-based speech recognition system using particle swarm optimization. 2009 Fourth International Conference on Bio-Inspired Computing (BIC-TA), Beijing, China,

16–19 October 2009. 2009. URL: https://doi.org/10.1109/bicta.2009.5338098

9. Furui S. History and Development of Speech Recognition. Speech Technology. New York, NY, 2010. P. 1–18. URL: https://doi.org/10.1007/978-0-387-73819-2_1

10. Introduction. Recurrent Neural Networks. 1999. P. 10–21. URL: https://doi.org/10.1201/9781420049176-1

11. An overview of the SPHINX-II speech recognition system / X. Huang et al. the workshop, Princeton, New Jersey, 21–24 March 1993. Morristown, NJ, USA, 1993. URL: https://doi.org/10.3115/1075671.1075690

12. Бронніков А. І., Онишко В. О. Обробка інформації при голосовому керуванні у робототехніці. Системи обробки інформації. 2017. № 3(149). С. 85–87. URL: https://doi.org/10.30748/soi.2017.149.17

13. Speech Research at Google to Enable Universal Speech Interfaces / M. Bacchiani et al. New Era for Robust Speech Recognition. Cham, 2017. P. 385–399. URL: https://doi.org/10.1007/978-3-319-64680-0_18

14. Lee A., Kawahara T., Shikano K. Julius. An open source real-time large vocabulary recognition engine. 7th European Conference on Speech Communication and Technology. ISCA, 2001. URL: https://doi.org/10.21437/eurospeech.2001-396

15. Silent Speech Interfaces for Speech Restoration: A Review / J. A. Gonzalez-Lopez et al. IEEE Access. 2020. Vol. 8. P. 177995–178021. URL: https://doi.org/10.1109/access.2020.3026579

16. Chung J. S., Zisserman A. Learning to lip read words by watching videos. Computer Vision and Image Understanding. 2018. Vol. 173. P. 76–85. URL: https://doi.org/10.1016/j.cviu.2018.02.001

17. Kapur A., Kapur S., Maes P. AlterEgo. IUI'18: 23rd International Conference on Intelligent User Interfaces, Tokyo Japan. New York, NY, USA, 2018. URL: https://doi.org/10.1145/3172944.3172977

18. Deep learning-based classification using Cumulants and Bispectrum of EMG signals / E. C. Orosco et al. IEEE Latin America Transactions. 2019. Vol.

17, no. 12. P. 1946–1953. URL: https://doi.org/10.1109/tla.2019.9011538

19. Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips / T. Hueber et al. Speech Communication. 2010. Vol. 52, no. 4. P. 288–300. URL: https://doi.org/10.1016/j.specom.2009.11.004

20. Vocal tract area function extraction using ultrasound for articulatory speech synthesis / D. R. Mohapatra et al. 11th ISCA Speech Synthesis Workshop (SSW 11). ISCA, 2021. URL: https://doi.org/10.21437/ssw.2021-16

21. Wand Michael, Koutník Jan, Schmidhuber Jürgen. Lipreading with Long Short-Term Memory // CoRR.— 2016.— Vol. abs/1601.08188.— URL: http://arxiv.org/abs/1601.08188.

22. Ялковський А. Є. Проблеми розпізнавання мови людини. Problems of Informatization and Management. 2009. Т. 3, № 27. URL: https://doi.org/10.18372/2073-4751.3.570

23. Loy C. C., Luo P., Huang C. Deep Learning Face Attributes for Detection and Alignment. Visual Attributes. Cham, 2017. P. 181–214. URL: https://doi.org/10.1007/978-3-319-50077-5_8

24. McManus S. Raspberry Pi. 3rd ed. 2017. 482 p.

25. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров: пер. с англ. Слинкина А. А., научный редактор Боресков А. В. — М.: ДМК Пресс, 2018. — 232 с.

26. Brahmbhatt S. Introduction to Computer Vision and OpenCV. Practical OpenCV. Berkeley, CA, 2013. P. 3–5. URL: https://doi.org/10.1007/978-1-4302-6080-6_1

27. Sasmita G. M. A., Dharmaadi I. P. A., Ferdian H. A. The Verification of Voice Recognition Using Cmusphinx and DTW. International Journal of Computer Applications Technology and Research. 2020. Vol. 9, no. 2. P. 076–082. URL: https://doi.org/10.7753/ijcatr0902.1007

28. Kumar S., Kiran S., Mishra N. Face mask detection using OpenCV. International journal of health sciences. 2022. P. 5282–5288. URL:

https://doi.org/10.53730/ijhs.v6ns2.6331

29. Lazebna N. ENGLISH-LANGUAGE BASIS OF PYTHON PROGRAMMING LANGUAGE. Research Bulletin Series Philological Sciences. 2021. Vol. 1, no. 193. P. 371–376. URL: https://doi.org/10.36550/2522-4077-2021-1-193-371-376

30. GPU Scripting and Code Generation with PyCUDA / A. Klöckner et al. 2012. P. 373–385. URL: https://doi.org/10.1016/b978-0-12-385963-1.00027-7

31. Visual Speech Recognition Using Optical Flow and Hidden Markov Model / U. Sharma et al. Wireless Personal Communications. 2018. Vol. 106, no. 4. P. 2129–2147. URL: https://doi.org/10.1007/s11277-018-5930-z

32. Christopher Tralie & Elizabeth Dempsey. Exact, parallelizable dynamic time warping alignment with linear memory. In Proceedings of the International Society for Music Information Retrieval Conference, 2020 https://doi.org/10.48550/arXiv.2008.02734

33. Improved algorithm of DTW in speech recognition / H. Zhi-Qiang et al. IOP Conference Series: Materials Science and Engineering. 2019. Vol. 563. P. 052072. URL: https://doi.org/10.1088/1757-899x/563/5/052072

34. Alsobhani A., ALabboodi H. M. A., Mahdi H. Speech Recognition using Convolution Deep Neural Networks. Journal of Physics: Conference Series. 2021. Vol. 1973, no. 1. P. 012166. URL: https://doi.org/10.1088/1742-6596/1973/1/012166

35. Biswas, A., Sahu, P.K., Chandra, M. Multiple cameras audio visual speech recognition using active appearance model visual features in car environment. Int. J. Speech Technol, 2016. – pp. 159-171.

36. Efficient End-to-End Sentence-Level Lipreading with Temporal Convolutional Networks / T. Zhang et al. Applied Sciences. 2021. Vol. 11, no. 15. P. 6975. URL: https://doi.org/10.3390/app11156975

37. Ali A., Renals S. Word Error Rate Estimation for Speech Recognition: e-WER. Stroudsburg, PA, USA, 2018. URL: https://doi.org/10.18653/v1/p18-2004