

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики



**Розробка системи цифрової обробки звукового сигналу для
подальшого використання в системах автоматизованого
розпізнавання голосу**

Курсова робота за спеціальністю „Прикладна математика ” 113

Керівник курсової роботи
доктор техн. наук, доц. Глибовець А.М.

_____ (підпис)

“ ____ ” _____ 2020 р.

Виконав
студент 4 курсу
факультету інформатики
Картавий М. О.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мережних технологій,
доктор техн. наук, доц. Глибовець А.М.

_____ (підпис)

„_____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Картавому Миколі Олексійовичу факультету інформатики 4 курсу
ТЕМА Розробка системи цифрової обробки звукового сигналу для подальшого використання в системах автоматизованого розпізнавання голосу

Вихідні дані:

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Анотація

Вступ

РОЗДІЛ 1: Актуальність проблеми розпізнавання природньої мови

РОЗДІЛ 2: Виділення основних ознак з аудіо сигналу

РОЗДІЛ 3: Цифрова обробка числових рядів

РОЗДІЛ 4: Тестування розробленої системи

Висновки

Список використаної літератури

Додатки

Дата видачі „_____” _____ 2020 р. Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Тема: Розробка системи цифрової обробки звукового сигналу для подальшого використання в системах автоматизованого розпізнавання голосу

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи.	07.11.2019	
2.	Пошук тематичної літератури	14.11.2019	
3.	Ознайомлення з тематичними матеріалами та побудова структури практичної та теоретичної частин курсової роботи	29.11.2019	
4.	Написання вступу та змісту роботи	04.01.2020	
5.	Ознайомлення з проблемою розпізнавання природньої мови та написання першого розділу	08.01.2020	
6.	Ознайомлення зі способами виділення ознак з аудіо сигналу та написання другого розділу	12.01.2020	
7.	Аналіз, виділення ознак з аудіо сигналу та тестування різних способів обробки аудіо сигналу	15.02.2020	
7.	Створення системи цифрової обробки аудіо сигналу та написання третього розділу	22.02.2020	
8.	Тестування створеної системи та написання четвертого розділу	18.03.2020	
9.	Оформлення роботи відповідно до вимог написання курсової роботи.	29.03.2020	
10.	Створення презентації та написання доповіді для захисту курсової роботи.	2020	
11.	Узгодження попередньої версії роботи з керівником	2020	
12.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника	2020	
13.	Захист курсової роботи	2020	

Студент Картавий М.О.

Керівник Глибовець А.М.

“ _____ ”

Зміст

<i>Використані терміни в роботі</i>	5
<i>Анотація до курсової роботи</i>	5
<i>Вступ</i>	6
<i>Загальні відомості про звук</i>	6
<i>Загальні відомості про шум</i>	7
<i>Розділ 1</i>	8
<i>Актуальність проблеми розпізнавання природньої мови</i>	8
1.1 Проблема розпізнавання природньої мови	8
1.2. Відомості про цифрову обробку звукових сигналів	9
1.3. Основна мета цифрової обробки чисельних рядів	9
<i>Розділ 2</i>	9
<i>Виділення основних ознак з аудіо сигналу</i>	9
2.1 Спектральний центроїд.....	10
2.2 Спектральний спад.....	10
2.3 Швидкість пересічення нуля	10
2.4 Мел-частотні кепстральні коефіцієнти	10
<i>Розділ 3</i>	11
<i>Цифрова обробка числових рядів</i>	11
3.1 Аналіз вхідного звукового сигналу	11
3.2 Аналіз звукового сигналу після «очищення» від тиші	15
3.3 Порівняємо детально фонему до та після очищення від тиші.....	19
3.4 Очищення звукового сигналу від шуму	23
3.5 Огляд фільтра Калмана	24
3.6 Програмна реалізація розробленої системи	26
3.7 Складність розробленої системи	27
<i>Розділ 4</i>	27
<i>Тестування розробленої системи</i>	27
4.1 Тестування.....	27
4.2 Лінійна регресія	28
4.3 Нейронна мережа	28
<i>Висновок</i>	30
<i>Список використаних джерел</i>	31

Використані терміни в роботі

Розпізнавання природної мови(РПМ(speech recognition з англ.)) – процес перетворення природної мови, шляхом переведення звукового сигналу в цифровий, наприклад, текст.

Фонема – умовна одиниця звуку.

Нейронна мережа(НМ(Neural network з англ.)) - сукупність станів з входами, проміжними(схованими) станами та виходами, в стані – певна математична функція, котра повертає значення на певному проміжку, а переходи мають ваги(коефіцієнти), котрі домножуються на вхідні дані.

Частоти центральних коефіцієнтів за шкалою Mel(ЧЦК за Mel(Mel frequency cepstral coefficients з англ.)) - спосіб представлення енергії спектру сигналу.

Частота кадрів(ЧК(Framerate з англ.)) – кількість одиниць частоти на 1 сек в звуковому файлі.

Коваріація – математичне сподівання добутку відхилень від математичних сподівань двох випадкових змінних.

Дисперсія випадкової величини – середньоквадратичне відхилення від середнього значення.

Матриця коваріації – це квадратна матриця, що складається з попарних коваріацій та дисперсій двох або більше випадкових величин.

Анотація до курсової роботи

Мета дослідження: побудувати систему цифрової обробки звукових сигналів для подальшого використання у системах переведення природної мови в текст, протестувати її.

Джерела для дослідження: використані іноземні джерела – наукова література та статті. Більш детально можна ознайомитися в розділі «Список використаних джерел».

Що було зроблено: ми розглянули основні проблеми цифрового представлення звуку. Створили комплексне рішення для подальшого використання системами розпізнавання природньої мови.

Дані для тестування системи: Дані, котрі використовуються для тестування системи – набір фонем української мови в розмірі 26 унікальних фонем (А,Б,В,Г,Д,Е,Й,Ж,З,І,К,Л,М,Н,О,П,Р,С,Т,У,Ф,Х,Ц,Ч,Ш,И), по 2000 прикладів кожної.

Вступ

Спроби перетворити природню мову в текст почалося ще з 1970-х років компанією DARPA, тому РПМ(speech to text) – не нове поняття в світі технологій. [1]

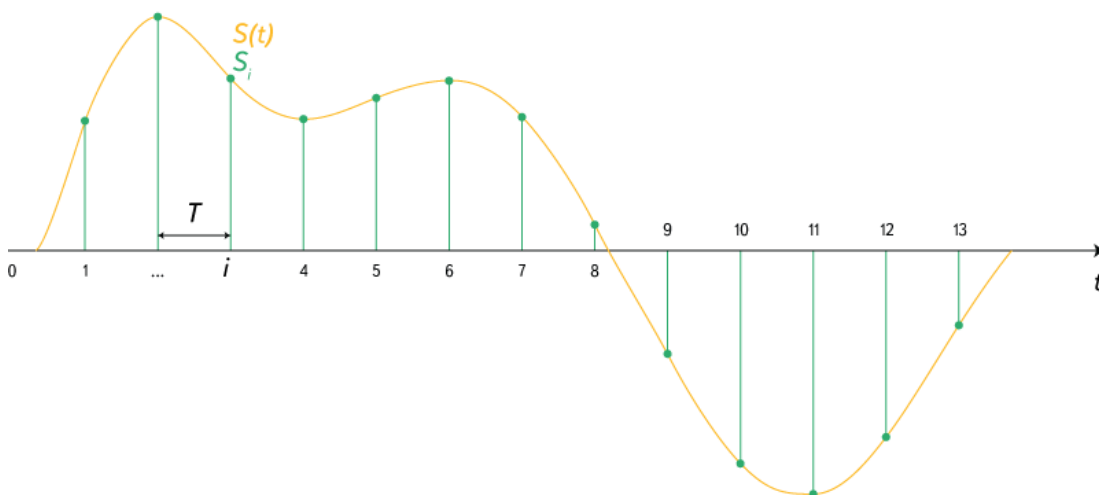
Де використовується? РПМ використовується для автоматичного переведення природньої мови в текст. Також переведення природньої мови в текст використовується в IoT(Internet of Things) для голосового керування будинком, апаратурою. Наразі активно РПМ використовується в системах для керування розумним будинком: Apple HomeKit, розумна станція Yandex – колонка Alisa, розумна станція Amazon – колонка Alexa, Google Home. Shazam – як приклад використання розпізнавання музики. Apple Siri, Google Assistant, Microsoft Cortana, Yandex Alisa – голосові помічники, котрі знаходяться на мобільних пристроях. Це найбільш поширене використання РПМ, інші приклади не можуть бути масовими, оскільки використовуються «ентузіастами» для побудови своєї системи розумного будинку з використанням переносних комп'ютерів: Rassyberry PI, Orange PI, Banana PI, інші.

Загальні відомості про звук

Процес перетворення звуку в цифрове представлення називається дискретизація. Цифровий сигнал – набір значень рівню сигналу, записаних через певний проміжок часу.

Як відомо, існує дві характеристики цифрового сигналу: частота дискретизації та глибина дискретизації.

На рисунку нижчу(цифрове представлення сигналу) можна бачити, що точки(зелений колір) – це міра сигналу в проміжок часу. Проміжок часу між точками позначений як T .



Рисуюнок цифрового представлення сигналу

Після проведених досліджень, було встановлено, що кожна фонема триває різну кількість часу в слові. Таким чином фонема «А» може тривати до 400 мсек, фонема «К» до 50 мсек.

Частота вказує на те, з якими інтервалами в часі надходять дані про рівень сигналу.

Глибина дискретизації вказує на точність запису сигналу. Стандартна глибина дискретизації на звичайних диктофонах 16 біт. Чим більша глибина – точність звуку вище, але це використовують в студіях звукозапису.

Загальні відомості про шум

В цифровому представленні звуку виділяють три види шуму:

- Джитер – випадкові відхилення сигналу, котрі зумовлені нестабільністю генератора частоти або різною швидкістю розповсюдження частотних характеристик одного сигналу.

- Шум дроблення – це шум, котрий пов’язаний з глибиною дискретизації. При цифровому представленні звуку деякі значення сигналу округлюються з деякою точністю – виникають слабкі шуми із-за втрати цих даних, що округлюються.
- Аліасинг – після цифрового представлення звуку в оригінальному представленні можуть виникнути частоти, котрих спочатку не було. Це пов’язано з частотою дискретизації. Частота впливає на проміжок часу між записами даних, якщо даних більше, а частота менша – частину даних ми втрачаємо.

Основною темою роботи є розробка системи для цифрової обробки звукового сигналу для подальшого опрацювання системами розпізнавання. Після розробки системи, визначити недоліки та способи їх вирішення. Пояснити чому саме ця система є корисною.

Розділ 1

Актуальність проблеми розпізнавання природньої мови

1.1 Проблема розпізнавання природньої мови

Оскільки представлення звуку залежить від багатьох факторів, то створити універсальну систему розпізнавання мови не є можливим на даний момент. На даний момент не існує універсальної системи розпізнавання природньої мови – дикторонезалежна, стійка до шумів, самонавчальна, щоб розпізнавала злитну мову, працювала зі словниками великих розмірів та при цьому мала б низький рівень помилки. Тому зараз РПМ має різні реалізації для конкретних задач та умов.

Наразі нам треба розробити систему для цифрової обробки звукового сигналу для подальшого використання у системах переводу природньої мови в текст. Ця система має прибирати неважливу інформацію з вхідного сигналу, та виділяти основні ознаки з обробленого сигналу.

1.2. Відомості про цифрову обробку звукових сигналів

Оскільки, звуковий сигнал у комп'ютерному представленні – це набір числових характеристик, то надалі можна використовувати поняття «цифрова обробка чисельних рядів».

Існує теорема Котельникова, котра стверджує що для того, щоб максимально точно відновити цифрове представлення сигналу з аналогового, то треба, щоб частота дискретизації була мінімум вдвічі більша за максимальну частоту вхідного сигналу. Тому, якщо брати діапазон частот, котрі може почути людина – від 20 Гц до 20 КГц, то оптимальна частота дискретизації має бути 40 КГц – стандарт частоти дискретизації дорівнює 44.1 КГц.

1.3. Основна мета цифрової обробки чисельних рядів

Серед чисельного ряду (а оскільки це цифрове представлення звуку) треба:

- визначити найбільш важливі для сприйняття людини частоти.
- згладити числові характеристики в чисельному ряді – прибрати викиди із звукового сигналу
- вивести «корисну» інформацію з кожного фрейма, та відкинути «неважливу» інформацію з кожного фрейма

Розділ 2

Виділення основних ознак з аудіо сигналу

Для опрацювання звуку я буду використовувати мову програмування Python, та бібліотеку Librosa. Отримати частотні характеристики можна за допомогою перетворення Фур'є – за нас це зробить бібліотека librosa, котра поверне нам частоту кадрів та набір самих частот.

До характеристик сигналу відносяться частота основного тону, частотні компоненти, спектральний центроїд, спектральний потік, спектральна площа, спектральний спад, тощо.

2.1 Спектральний центроїд

Спектральний центроїд вказує на якій частоті зосереджена енергія спектру – центр мас для звуку. Та обчислюється за формулою:

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)}$$

За формулою:

$S(k)$ – спектральна величина елемента k

$f(k)$ – частота елемента k

2.2 Спектральний спад

Спектральний спад – це міра форми сигналу, що представляє з себе частоту, де високі частоти дорівнюють нулю. Щоб отримати її необхідно розрахувати долю елементів в спектрі потужності, де 85% її потужності знаходиться на більш низьких частотах.

2.3 Швидкість пересічення нуля

Простий спосіб вирахувати гладкості сигналу – порахувати кількість пересічень нуля в межах цього сигналу. Голосовий сигнал повільно пересікає нуль. Наприклад сигнал з частотою 100Гц буде пересікати нуль 100 разів на секунду, тоді як тишина буде мати близько 3000 пересічень нуля.

2.4 Мел-частотні спекральні коефіцієнти

Це коефіцієнти, котрі представляють з себе набір ознак(зазвичай 10-20), котрі описують енергію сигналу. За допомогою мел-шкали вираховують важливі для сприйняття людини частоти. Обраховуються за формулою:

$$m = 1125 \ln\left(1 + \frac{f}{700}\right)$$

За формулою:

f - частотна міра вхідного сигналу

m – значення певної частоти за шкалою mel – рисунок 1.

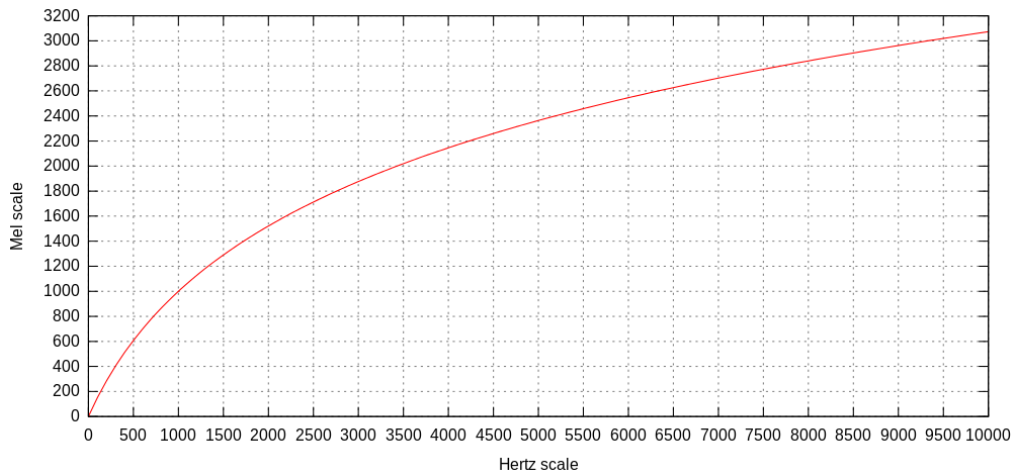


Рисунок 1

Розділ 3

Цифрова обробка числових рядів

3.1 Аналіз вхідного звукового сигналу

Розберемо фонему літери «А» в розмірі 10 примірників. Розглянемо рисунок 2 - графік частот необробленої фонему літери «А». По осі x зображено порядковий номер частоти, а по осі y міру частоти.

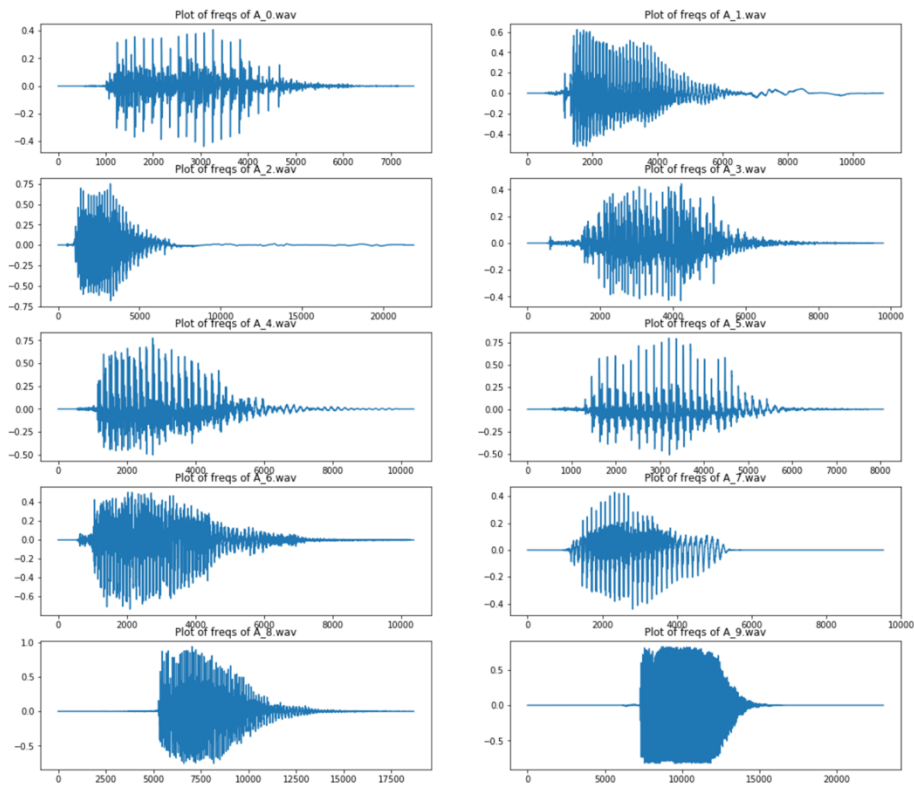


Рисунок 2

Досить важко сказати, що саме в них подібного, тому дослідимо далі основні ознаки аудіо сигналів на прикладі цих 10 примірників фонем.

Спектральний центроїд – на рисунку 3 зображено графіки частоти та графік кривої центроїдів фонем. Загалом, де графік кривої більш рівний відносно осі x там і зосереджена основна енергія спектру. Зверніть увагу, на значення пересічення нуля, зазначених в підписах графіків, щоб порівняти їх після обробки.

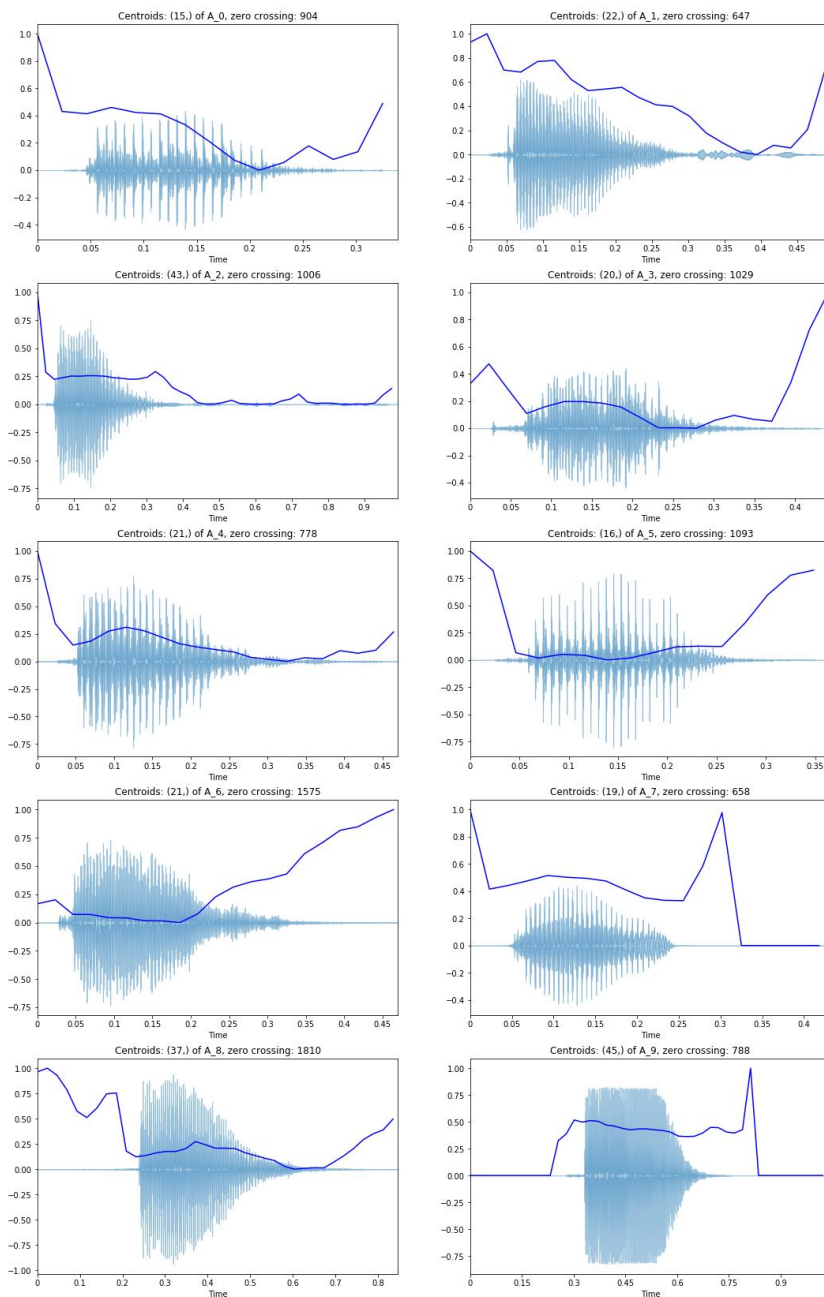


Рисунок 3

Спектральний спад - на рисунку 4 зображено графіки частоти та графік кривої спектральних спадів фонем. Можна бачити, що на даному етапі графіки центроїдів та спадів однакові, але це різні графіки, після проведення через деякі функції це можна буде побачити – оскільки зараз вхідні дані не є обробленими, то шум та інше «сміття» впливають графік центроїдів та спектрального спаду.

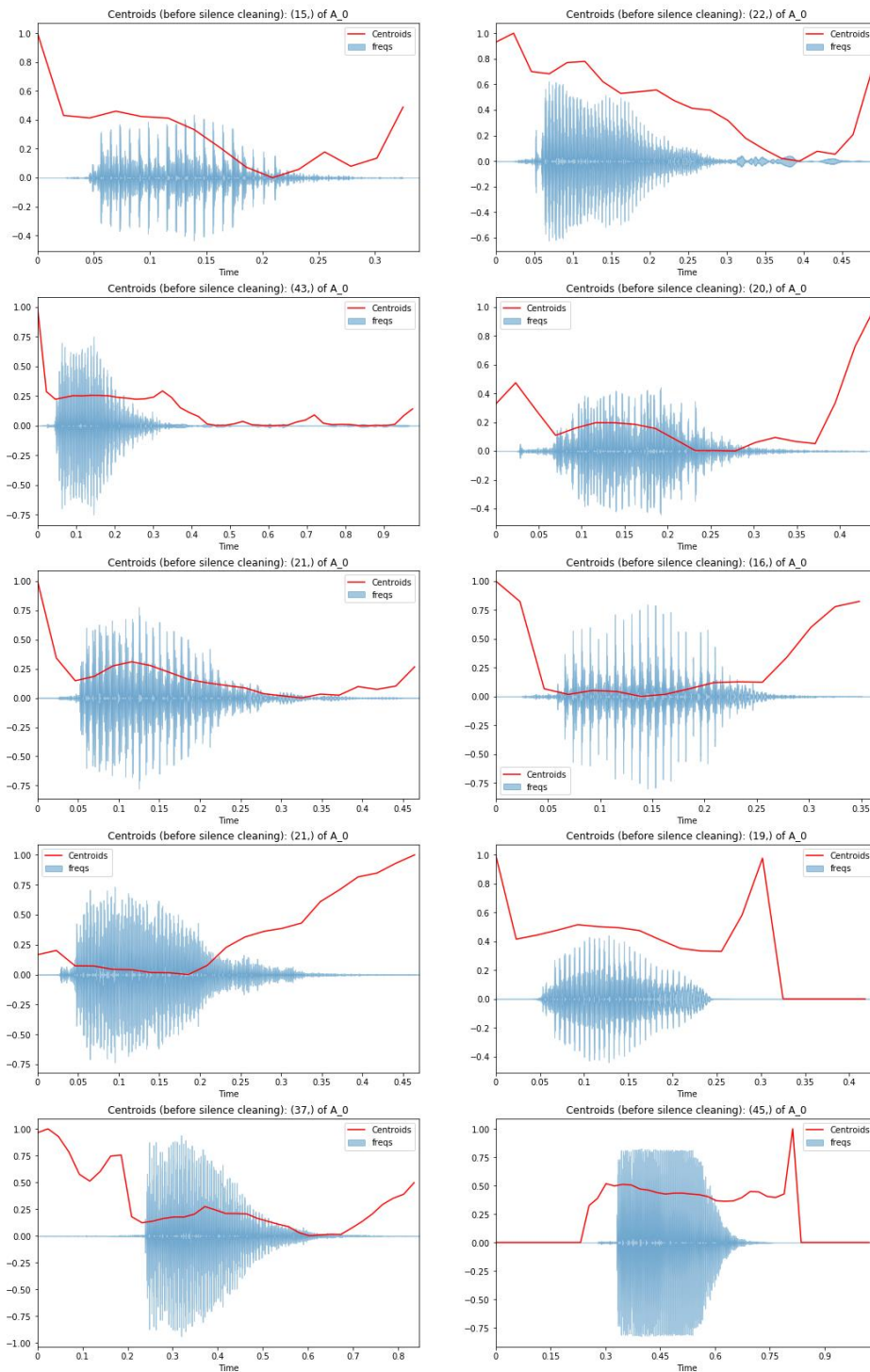


Рисунок 4

Розглянемо спектрограму мел-частотних спектральних коефіцієнтів на прикладі цих самих 10 фонем – рисунок 5.

По осі у зображено міру частоти в Гц, а колір вказує на потужність гучності, де синій колір – тихо, червоний колір - гучно.

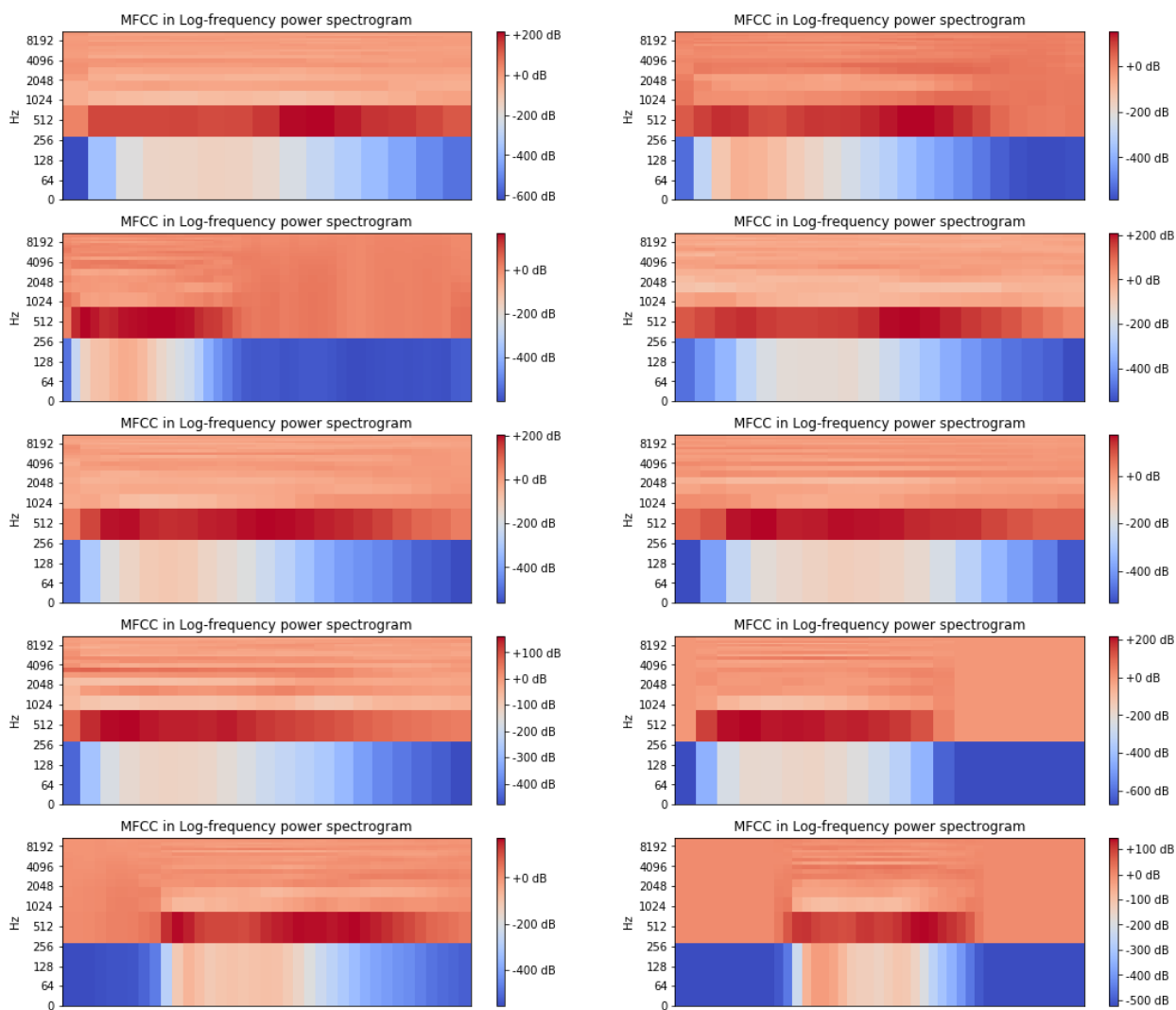


Рисунок 5

3.2 Аналіз звукового сигналу після «очищення» від тиші

Щоб працювати далі з потрібними даними необхідно прибрати «порожній» шум – тишу, тоді кількість даних, котрі треба буде обробляти зменшиться. Для цього скористаємося вбудованою функцією в бібліотеку Librosa. Викличемо функцію `librosa.effects.trim()`, куди як параметр передамо частоти отримані з аудіо файлу – на виході отримаємо сигнал без «тиші».

Графік частот фонемі «А» після очищення тиші – рисунок 6

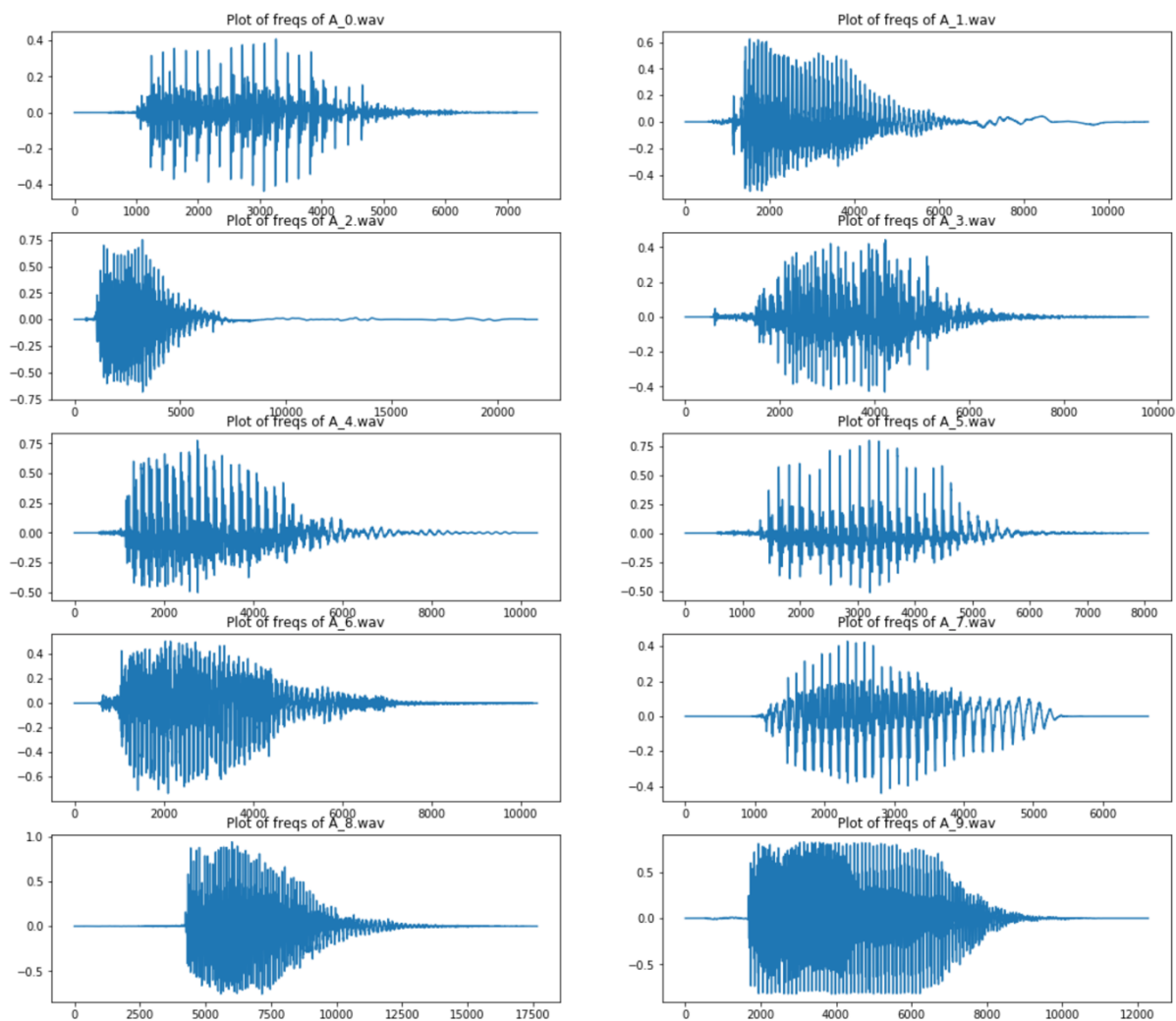


Рисунок 6

Спектральний центроїд – на рисунку 7 зображено графіки частоти та графік кривої центроїдів фонемі після очищення від шуму. Звернімо увагу на значення пересічення нуля, зазначених в підписах графіків – в основному значення не змінилось – можемо зробити висновок, що тиші в обраних фреймах було небагато.

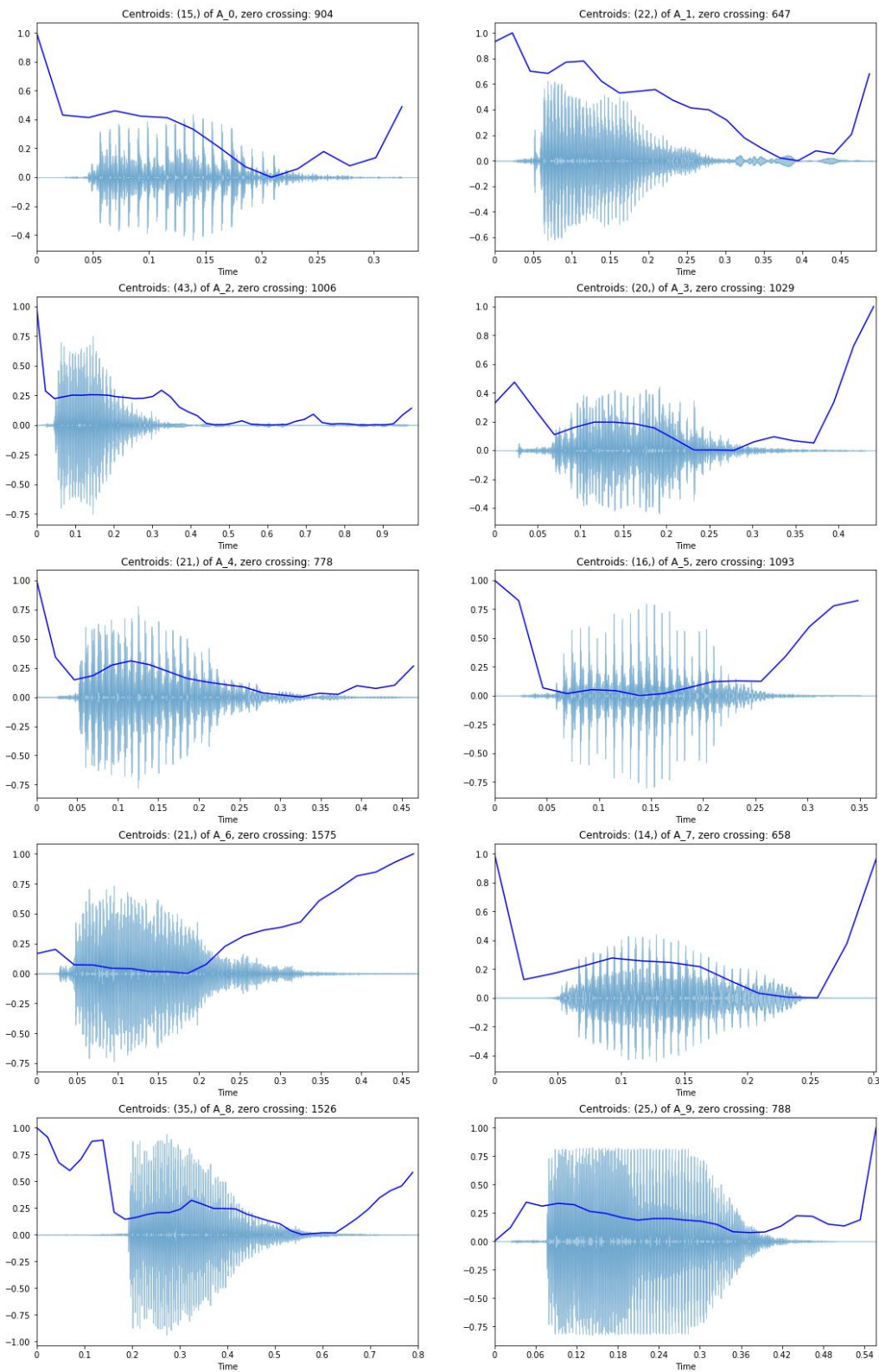


Рисунок 7

Спектральний спад - на рисунку 8 зображено графіки частоти та графік кривої спектральних спадів фонемі після очищення від шуму.

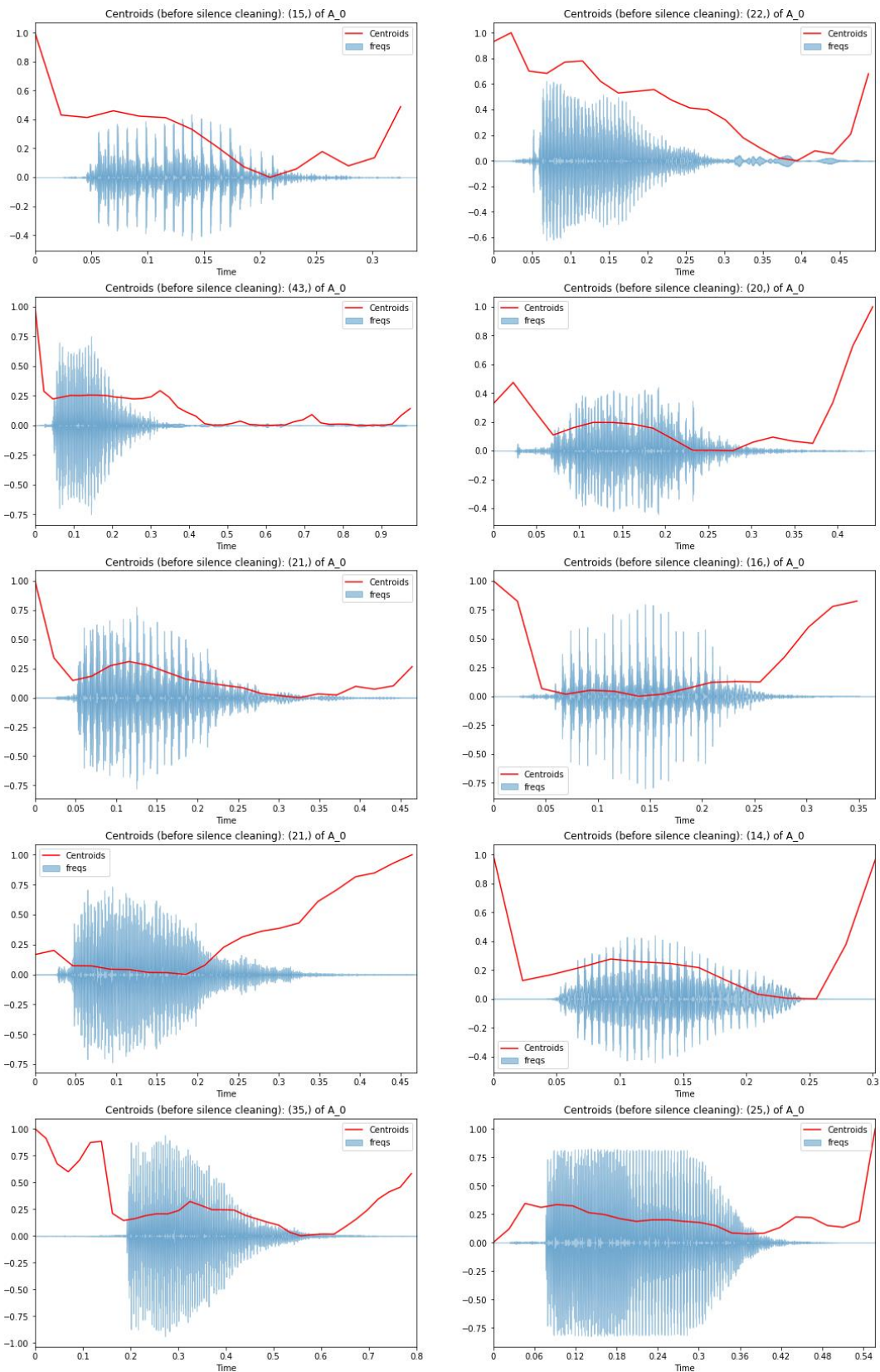


Рисунок 8

Розглянемо спектрограму мел-частотних спектральних коефіцієнтів на прикладі цих самих 10 фонем – рисунок 9.

Опис спектрограми вказаний до рисунку 5.

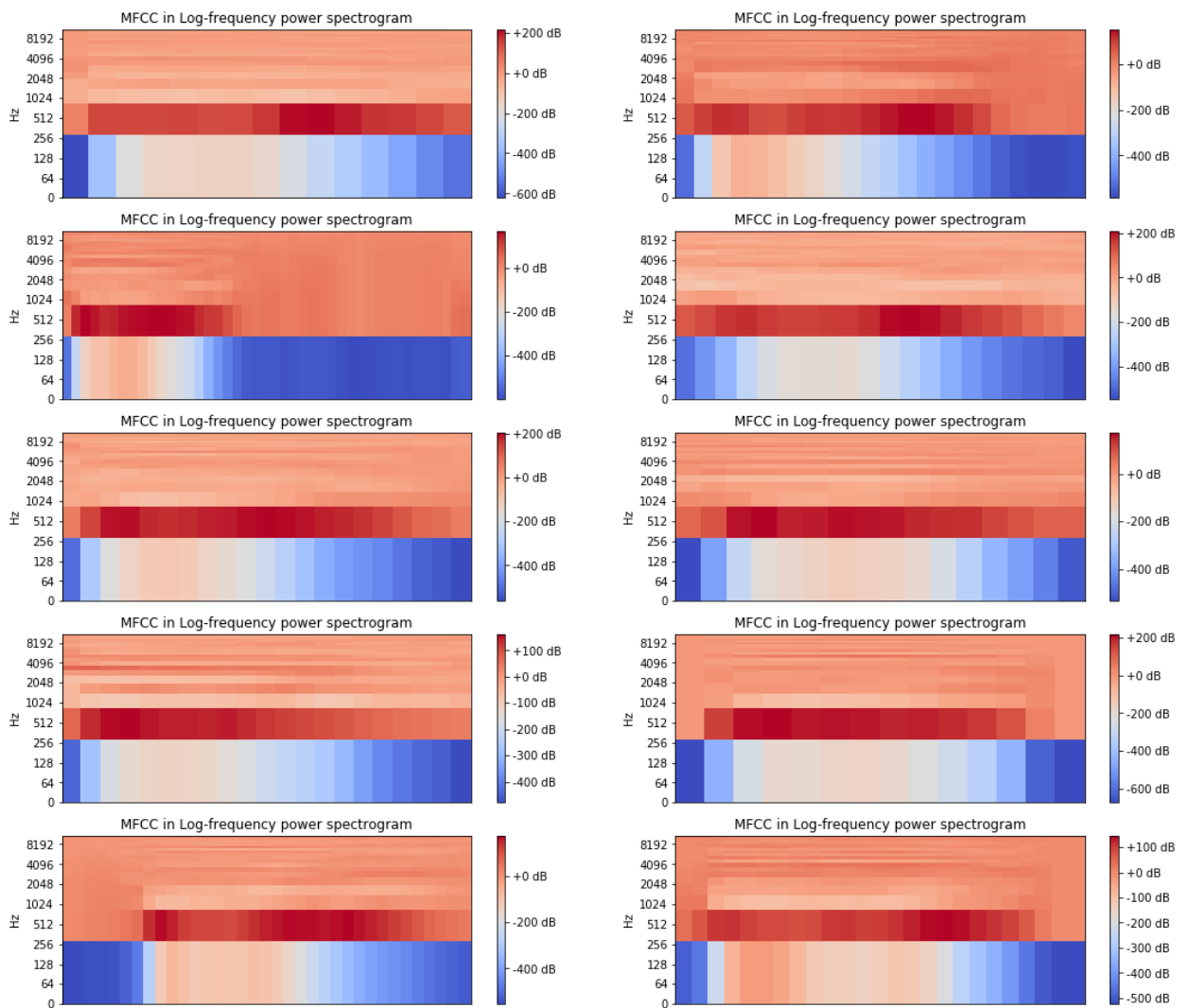


Рисунок 9

3.3 Порівняємо детально фонему до та після очищення від тиші

Графік частот фонему «А» до(зліва) та після(справа) очищення від тиші – рисунок 10. На рисунку 10 можна побачити різницю у розмірі фрейму до та після очищення, якщо подивитись на шкалу x координат, – зліва графік частот до очищення з кількістю частот 23038 та справа – після очищення з кількістю частот 12288.

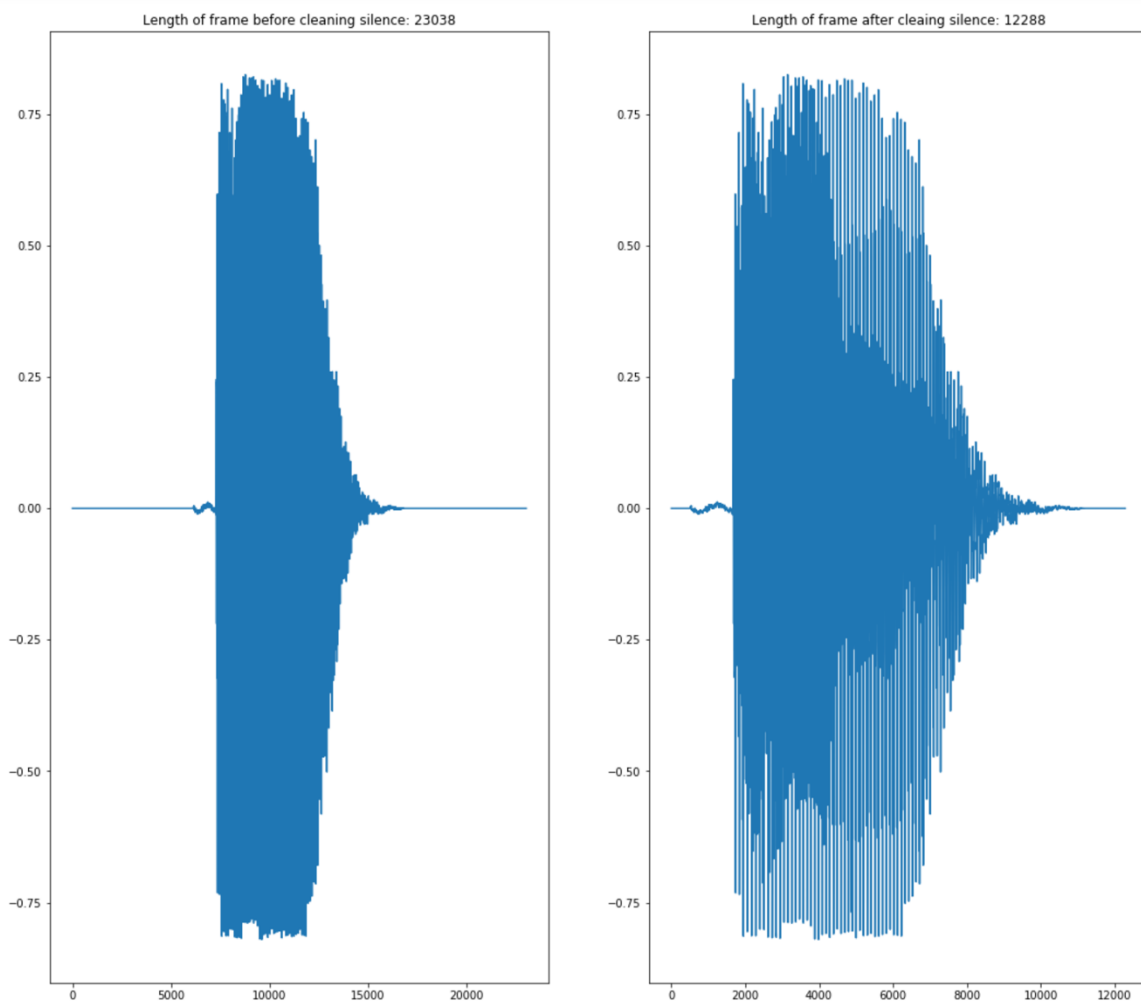


Рисунок 10

Графіки кривої центроїдів до(зліва) та після(справа) очищення від шуму фонемі «А» - рисунок 11. Можна побачити, що кількість пересічень нуля не змінилась – не було «тиші», частота котрої пересікала нуль.

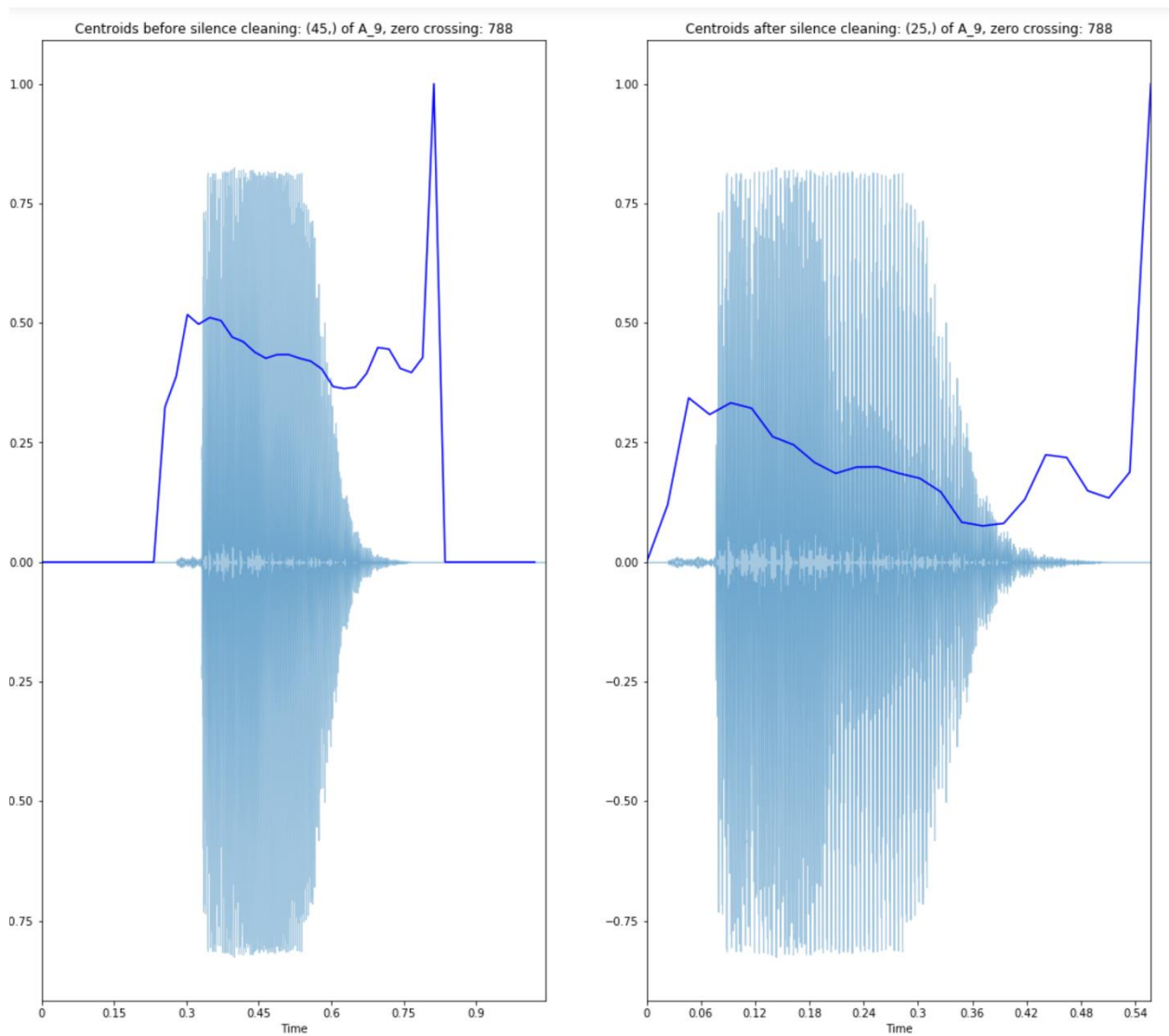


Рисунок 11

Спектральний спад - на рисунку 12 зображено графіки частоти та графік кривої спектральних спадів фонем до(зверху) та після(знизу) очищення від шуму. Загалом, крива начебто не змінилась. Також можна бачити, що графік спектрального спаду після очищення від тиші відрізняється від графіку центроїдів – деяка частина тиші була прибрана з «корисних» даних, тому графік центроїдів змінився і відрізняється від графіку спаду.

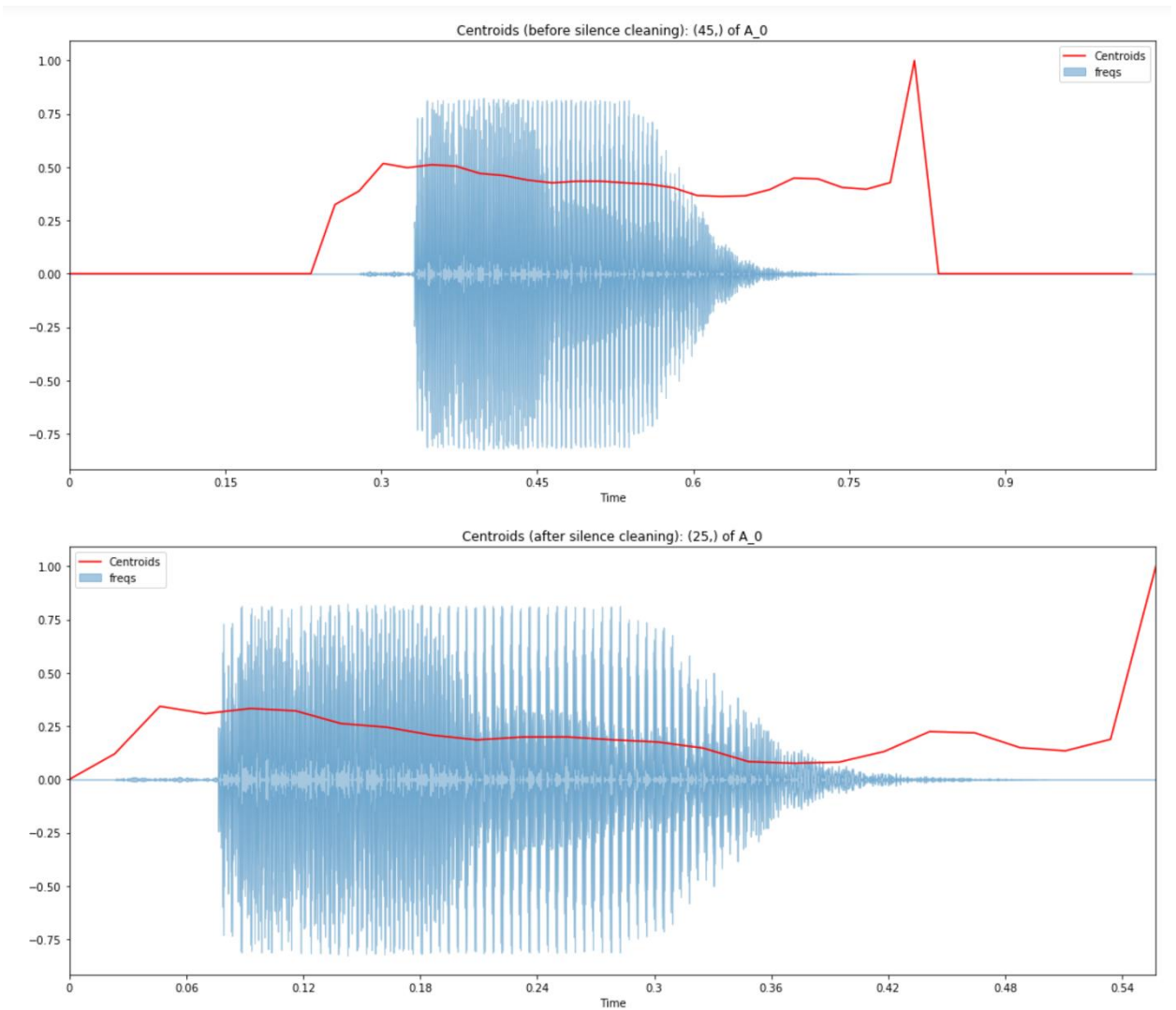


Рисунок 12

Розглянемо спектрограму мел-частотних спектральних коефіцієнтів до(зліва) та після(справа) очищення від тиші – рисунок 13. Можна бачити, що в фонемі після очищення від тиші більше потужність звуку, а отже, тиші немає.

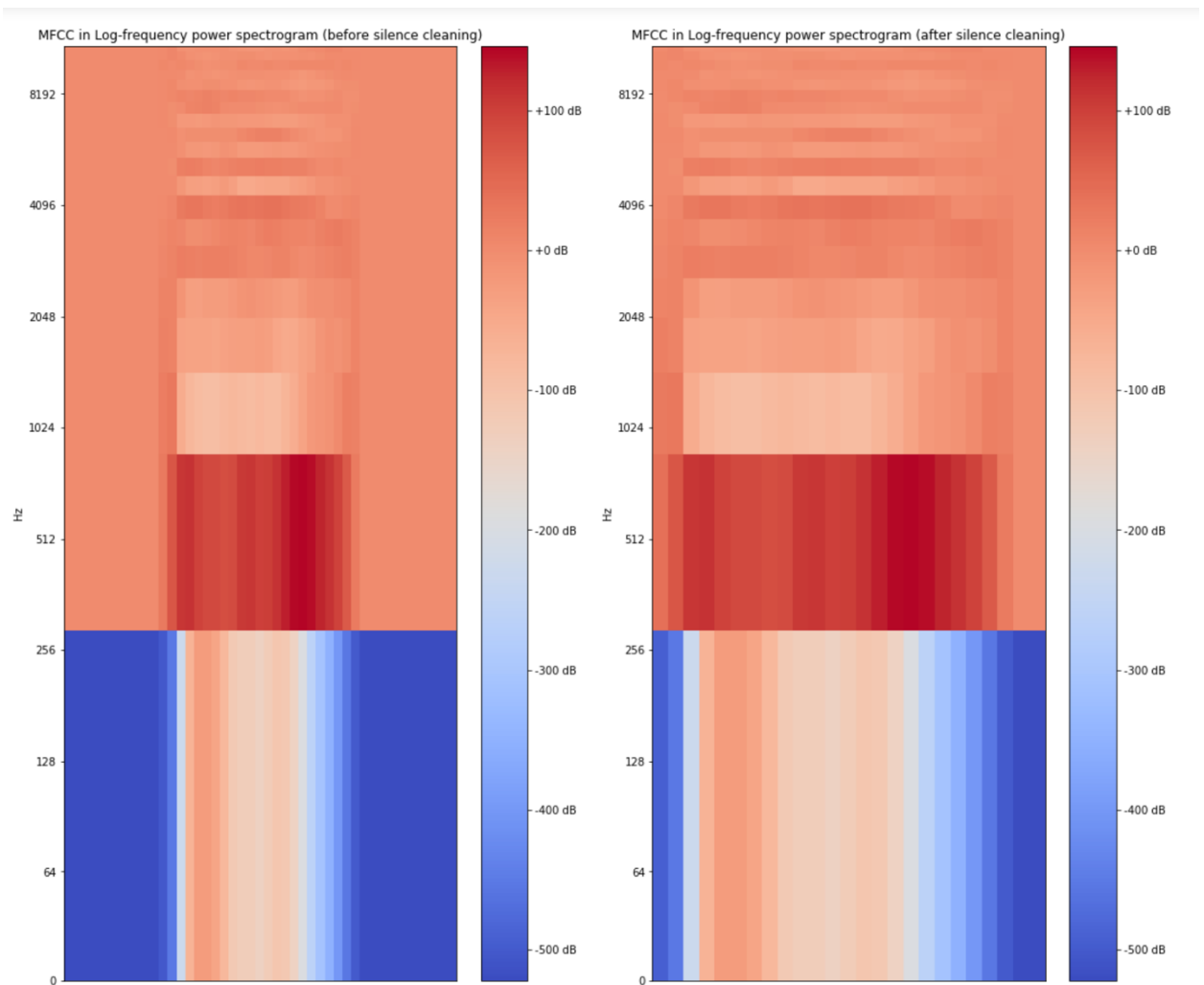


Рисунок 13

3.4 Очищення звукового сигналу від шуму

Для очищення звукового сигналу, а як ми раніше називали числового ряду, було застосовано фільтр Калмана. Фільтр Калмана - це алгоритм, що використовує послідовності вимірювань протягом часу, які містять шум (випадкові відхилення) та інші неточності, й видає оцінки невідомих змінних, що є потенційно точнішими за базовані на самих лише вимірюваннях. Формальніше, фільтр Калмана працює рекурсивно на потоках зашумлених вхідних даних, і видає статистично оптимальну оцінку базового стану системи.

На рисунку 3 можемо бачити синім кольором вхідний сигнал, та червоним сигналом після обробки фільтром Калмана – прибрані високочастотні шуми. Як

приклад можна бачити на рисунку 14, де синій колір – вхідні дані, а червоний колір – прибрані високочастотні шуми.

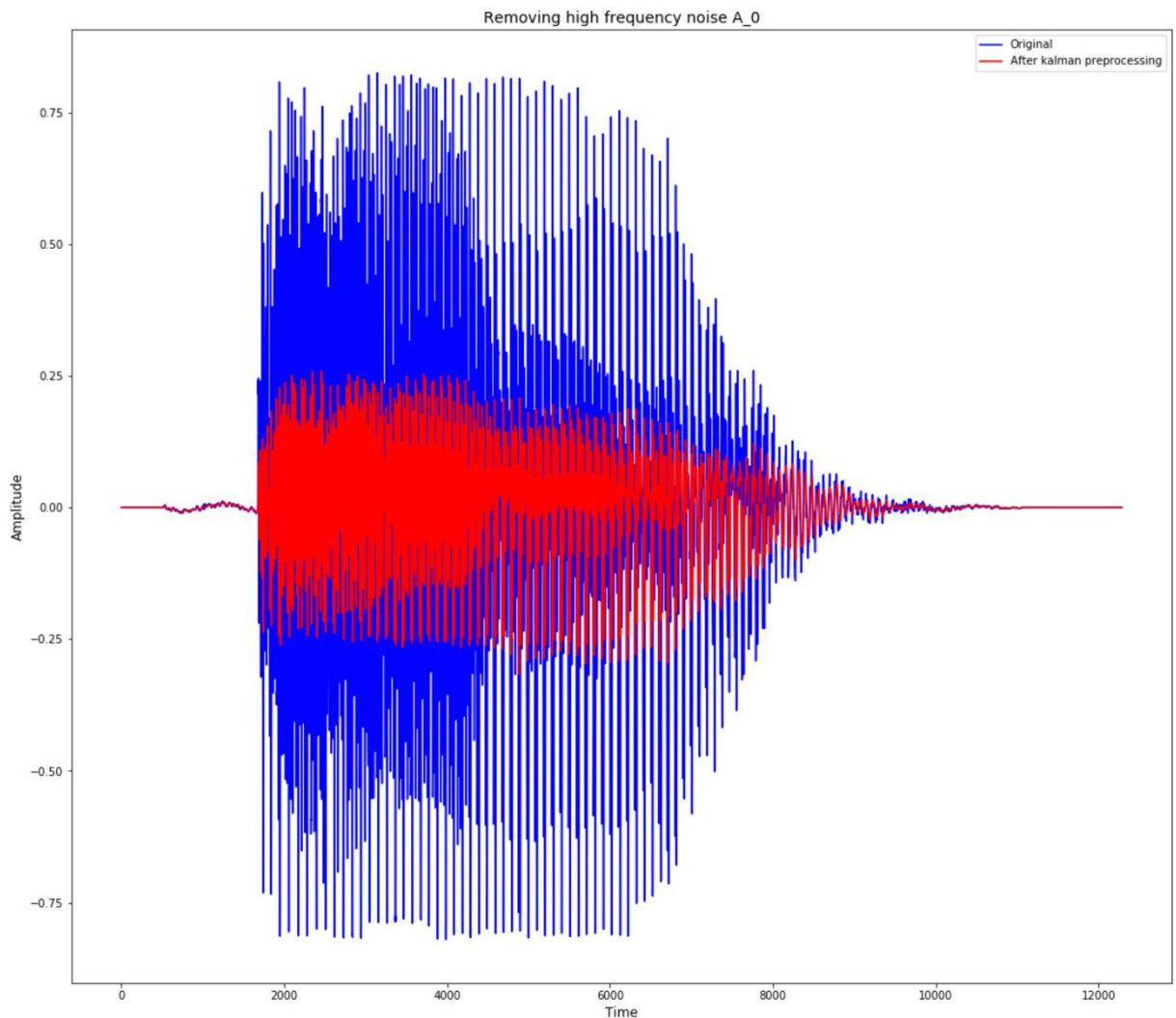


Рисунок 14

3.5 Огляд фільтра Калмана

Фільтр Калмана використовує динамічну модель системи (наприклад, фізичний закон руху), відомі керуючі взаємодії та багаточисельні послідовні вимірювання для формування оптимальної оцінки стану. Алгоритм складається з двох етапів, що повторюються: передбачення та корегування.

Передбачення

Передбачення стану системи:

$$\hat{x}_k^- = F * \hat{x}_{k-1} + B * u_{k-1}$$

За формулою:

k – індекс, що позначає даний момент часу,

«-» в верхньому індексі позначає, що це передбачення – проміжне значення,

\hat{x}_k^- - передбачення системи в даний момент часу,

F – матриця переходу між станами (динамічна модель системи),

\hat{x}_{k-1} - передбачення системи в минулий момент часу,

B – матриця застосування керуючої взаємодії

\hat{u}_{k-1} – керуюча взаємодія в минулий момент часу.

Передбачення помилки коваріації:

$$P_k^- = F * P_{k-1} * F^T + Q$$

За формулою:

P_k^- - передбачення помилки,

P_{k-1} - помилка в минулий момент часу,

F^T – транспонована F,

Q – коваріація шуму процесу.

Корегування

Обчислення підсилення Калмана:

$$K_k = P_k^- * H^T * (H * P_k^- * H^T + R)^{-1}$$

За формулою:

K_k - підсилення Калмана,

H – матриця змін, котра відображає відношення вимірювань зі станами,

H^T – транспонована H,

R – коваріація шуму вимірювань.

Оновлення оцінки з урахуванням вимірювання z_k :

$$\hat{x}_k = \hat{x}_k^- + K_k * (z_k - H * \hat{x}_k^-)$$

За формулою:

z_k – вимірювання в даний момент часу.

Оновлення помилки коваріації:

$$P_k = (I - K_k * H) * P_k^-$$

За формулою:

I – матриця ідентичності

Тепер застосуємо все вище наведене, проведемо тести, та підіб'ємо підсумки. А саме очистимо дані від тиші, очистимо від високочастотних шумів – отримаємо «корисні» дані, котрі ще обробимо функцією мел-кепстральних коефіцієнтів, котра визначить енергію кожного сигналу.

3.6 Програмна реалізація розробленої системи

З урахуванням того, що дані на вхід отримуємо як набір частот та частоту дискретизації

```
def audioPreprocessing(signal,sampleRate):
#Очищення сигналу від тиші
signal= librosa.effects.trim(signal)[0]
F = eye(2)
Q = eye(2)*3e-3
H = eye(2)
R = eye(2)*3e-1
x = z[0,:].transpose()
P = Q
kalmanFiltering = []
for i in range(1,len(signal)):
    x = F.dot( x )
    P = F.dot(P).dot( F.transpose() ) + Q
    y = z[i] - H.dot(x)
    K = P.dot(H.transpose()).dot(linalg.inv(H.dot(P) .dot(H.transpose()) + R))
    x = x + K.dot(y)
    P = (eye(z.shape[1]) - K.dot(H)).dot(P)
    kalmanFiltering.append(x.transpose())
res = librosa.feature.mfcc(kalmanFiltering[:,1],sr= sampleRate)
return res
```

На виході отримуємо функцію дані оброблені.

3.7 Складність розробленої системи

Алгоритм системи досить простий:

- Обробити вхідний сигнал функцією `librosa.effects.trim()` для очищення від тиші
- Очистити сигнал фільтром Калмана для очищення від шуму та випадкових значень
- Накласти функцію мел-кепстральних коефіцієнтів для виділення енергії сигналу

Складність `librosa.effects.trim()` з використанням `lowpass filter` – фільтр для прибирання наднизьких частот $O(5N)$

Складність фільтра Калмана $O(N^3)$.

Складність підрахування мел-кепстральних (реалізація була взята з бібліотеки `librosa`) коефіцієнтів: $O(1)$

Загальна складність системи: $O(N^3)$.

Розділ 4

Тестування розробленої системи

4.1 Тестування

Тестування розробленої системи буде виконуватися шляхом побудови системи для розпізнавання фонем, а саме лінійної регресії та простої нейронної мережі. Будуть зафіксовані константи для випадкових значень, щоб для кожного тесту умови були однакові. Будемо тестувати:

- звичайні дані, без обробки
- дані після очищення від тиші та з використанням фільтра Калмана
- дані, без обробки з виділенням мел-кепстарльних коефіцієнтів
- дані після очищення від тиші та з використанням фільтра Калмана з виділенням мел-кепстральних коефіцієнтів

Для тесту було відібрано 26 фонем по 1400 примірників кожної для тренування та 600 для тестування.

Для розбиття на дані для тренування та дані для тестування використовувалася функція `train_test_split` з бібліотеки `sklearn`.

Приклад використання:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y_processed, test_size=0.3,
random_state=42)
```

Де `X` – набір даних, а `Y_processed` – індекси фонем у словнику

4.2 Лінійна регресія

Було використано реалізацію лінійної регресії з бібліотеки `sklearn`.

Використання лінійної регресії:

```
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(X_train, Y_train)
reg.score(X_test, Y_test)
```

Дані та результати регресії на даних описано нижче.

Обрахування відсотків правильно передбачених фонем проводилося шляхом виклику функції `score` реалізації лінійної регресії `sklearn`.

Ознайомитись з результатами можна в таблиці №1.

4.3 Нейронна мережа

Було побудовано нейронну мережу з використанням LSTM рівня, Dropout рівня, та використанням бібліотеки `Keras`.

Для спрощення навчання нейронної мережі – дані без використання мелкепстральних коефіцієнтів були відкинуті, також кількість епох дорівнювала 20.

Реалізація нейронної мережі:

```
import keras
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout
categorical_labels = to_categorical(Y_train)
categorical_labels_val = to_categorical(Y_val)
```

```

model = Sequential()
model.add(Dense(len(np.unique(Y_train)),activation = "relu"))
model.add(Dropout(0.2))
model.add(Dense(50, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(50, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(40,activation = "sigmoid"))
model.add(Dropout(0.2))
model.add(Dense(30,activation = "relu"))
model.add(Dropout(0.2))
model.add(Dense(30,activation = "sigmoid"))
model.add(Dropout(0.2))
model.add(LSTM(26, return_sequences=False, dropout=0.3,
recurrent_dropout=0.2,activation="sigmoid"))
model.add(Dropout(0.2))
model.add(Dense(52, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(len(np.unique(Y_train)), activation="softmax"))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model.fit(X_train, categorical_labels, batch_size=50, epochs=20, verbose=1,
validation_data=(X_val, categorical_labels_val))

```

Ознайомитись з результатами можна в таблиці №1.

Таблиця №1

Спосіб тестування	Дані без обробки з мел(вибірка для навчання/тестова %)	Дані після очищення від тиші + Калман з мел(вибірка для навчання/тестова %)	Дані без обробки та без мел(вибірка тестова %)	Дані після очищення від тиші + Калман без мел(вибірка тестова %)

Лінійна регресія	-/17	-/26	12	18
Нейронна мережа	21/34	25/39	-	-

Висновок

В цій роботі ми створили систему цифрової обробки звуку, протестували, та виявили, що система покращує результати розпізнавання природньої мови в існуючих системах ніж якщо не використовувати її. Оскільки, відсоток розпізнаних даних нейронною мережею дорівнює 39, то цей результат не можна вважати задовільним.

Такий результат зумовлений:

- Якістю вибірки для тестування
- Структурою нейронної мережі
- Обмеження в потужності для обрахування більших даних

Результати можна покращити:

- Покращити якість вибірки
- Зробити вибірку з використанням більшої кількості даних
- Змінити побудову нейронної мережі
- Використовувати більш потужні машини для навчання нейронної мережі

Отже, розроблена система не є рішенням проблеми розпізнавання природньої мови, проте може покращити вже існуючі системи розпізнавання мови. Я вважаю, що тема розпізнавання природньої мови є актуальною й потребує більшої уваги до себе для покращення вже існуючих систем розпізнавання або створення нових систем розпізнавання.

Список використаних джерел

- 1) <https://habr.com/ru/post/131945/>
- 3) «Классификация распознавания речи» Федосин С.А., Еремин А. Ю.
- 4) Research Article “An Overview of Speech Recognition Using HMM” Ms. Rupali S Chavan , Dr. Ganesh. S Sable
- 6) Research Article “An Overview of Speech Recognition Using HMM” Ms. Rupali S Chavan , Dr. Ganesh. S Sable
- 7) https://en.wikipedia.org/wiki/Dynamic_time_warping
- 8) <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>
- 9) <https://studopedia.info/6-90603.html>
- 10) <https://habr.com/ru/post/140828/>
- 11) https://en.wikipedia.org/wiki/Mel_scale#/media/File:Mel-Hz_plot.svg
- 12) <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- 13) <https://medium.com/nuances-of-programming/анализ-аудиоданных-с-помощью-глубокого-обучения-и-python-часть-1-2056fef8525e>
- 14) <https://habr.com/ru/company/yandex/blog/270765/>