**Лабораторна робота**

**з предмету Машинне навчання**

**на тему «Аудіокласифікація собак та кішок»**

**студента групи ММАІ-1**

**Ліпський Даниїл**

Зміст

[Набір даних 3](#_Toc88413555)

[Мел-кепстральні коефіцієнти (MFCC) 5](#_Toc88413556)

[**Основні поняття** 5](#_Toc88413557)

[**Покроковий розбір** 6](#_Toc88413558)

[Часове представлення аудіо файлу 6](#_Toc88413559)

[Cпектр вихідного сигналу (отримуємо за допомогою перетворення Фур'є) 6](#_Toc88413560)

[Аудіо кадрування 6](#_Toc88413561)

[Перетворення на частотну область 6](#_Toc88413562)

[Отриманий спектр розташовуємо на мел-шкалі 8](#_Toc88413563)

[Обчислюємо точки фільтрації 8](#_Toc88413564)

[Генерація кепстральних коефіцієнтів 9](#_Toc88413565)

[Вилучення ознак аудіофайлів за допомогою MFCC 9](#_Toc88413566)

[Компонування тренувального набору даних 9](#_Toc88413567)

[Компонування випробувального набору даних 9](#_Toc88413568)

[Побудова моделі класифікації звуку 10](#_Toc88413569)

[Перевірка моделі 11](#_Toc88413570)

[Висновок 11](#_Toc88413571)

## **Набір даних**

Компонуємо тренувальний та випробувальний набори даних у вигляді таблиць за допомогою наступного алгоритму

for folder in os.listdir(directory):

for filename in os.listdir(directory+str(folder)):

f = os.path.join(directory+str(folder), filename)

if os.path.isfile(f):

ID.append(f.split('/')[-1])

label.append(f.split('/')[-2])

Кожна таблиця складається з двох колонок: ID – назва файлу та вид тварини.

Також наведені графіки кількісного розподілу даних

## **Мел-кепстральні коефіцієнти (MFCC)**

### **Основні поняття**

MFCC (Mел-кепстральні коефіцієнти) - це спосіб вилучення функцій зі звуку. Mел - одиниця висоти звуку, заснована на сприйнятті цього звуку нашими органами слуху. MFCC використовує шкалу MEL для поділу смуги частот на піддіапазони, а потім витягує кепстральні коефіцієнти за допомогою дискретного косинусного перетворення (DCT). Шкала MEL ґрунтується на способі, яким люди розрізняють частоти, що робить обробку звуків дуже зручною.

Мовний апарат людини можна у вигляді сукупності генераторів тонових сигналів і шумів, а також фільтрів. Схематично це можна так:

1. Генератор імпульсної послідовності (тонів)

2. Генератор випадкових чисел (шумів)

3. Коефіцієнти цифрового фільтра (параметри голосового тракту)

4. Нестаціонарний цифровий фільтр

Сигнал на виході фільтра (4) можна подати у вигляді згортки

де s(t) – початковий вид акустичної хвилі, а h(t) – характеристика фільтра (залежить від параметрів голосового тракту)  
  
У частотній області це виглядає так

Добуток можна прологарифмувати, щоб отримати замість нього суму

Тепер нам потрібно перетворити цю суму так, щоб отримати набори характеристик вихідного сигналу і фільтра, що не перетинаються. Для цього застосовуємо зворотне перетворення Фур'є, яке дасть нам наступне

Це і є Кепстр - один із видів гомоморфної обробки сигналів, функція зворотного перетворення Фур'є від логарифму спектра потужності сигналу.

### **Покроковий розбір**

### Часове представлення аудіо файлу

### Cпектр вихідного сигналу (отримуємо за допомогою перетворення Фур'є)

### Аудіо кадрування

Оскільки звук – це нестаціонарний процес, FFT викликає спотворення. Щоб подолати це, ми можемо припустити, що звук є стаціонарним процесом упродовж коротких періодів часу. Через це ми розбиваємо сигнал на короткі кадри. Кожен аудіо-кадр буде того самого розміру, що і FFT. Також ми хочемо, щоби кадри перекривалися. Тому що ми втрачаємо інформацію по краях кожного кадру після застосування віконної функції.

### Перетворення на частотну область

Тепер ми перетворимо звук, який в даний час знаходиться в часовій області, частотну область. FFT передбачає, що звук є періодичним і тривалим. Кадруючи сигнал, ми гарантували, що звук буде періодичним. Щоб звук був тривалим, ми використовуємо віконну функцію кожного кадру. Якщо ми цього не зробимо, отримаємо високочастотні спотворення. Щоб подолати це, нам потрібно спочатку застосувати віконну функцію до кадрованого звуку, а потім виконати FFT. Вікно гарантує, що обидва кінці сигналу будуть близькі до нуля.

На графіці вище ви можете бачити, що обидва кінці кадру закінчуються в різних місцях по осі Y. Ви можете бачити, що у вікні краю кожного кадру наблизилися до нуля.

Тепер давайте виконаємо FFT. Після виконання FFT ми беремо лише позитивну частину спектра (перша половина +1).

### Отриманий спектр розташовуємо на мел-шкалі

Тут ми обчислюємо набір фільтрів, що рознесені на MEL, а потім пропускаємо через них кадрований звук. Це дасть нам інформацію про потужність у кожній смузі частот. Фільтри можуть бути створені для будь-якої лінії частот, але в нашому прикладі ми будемо розглядати всю смугу дискретизації.

Що характерно для набору фільтрів з рознесенням MEL, так це відстань між фільтрами, що експоненційно зростає із частотою. Набір фільтрів можна зробити для будь-якої лінії частот. Тут ми обчислимо набір фільтрів для смуги частот.

### Обчислюємо точки фільтрації

Спочатку ми створюємо точки фільтра, які визначають початок та кінець фільтрів. Для цього ми спочатку перетворимо два краї filterbank на простір MEL. Після цього ми будуємо лінійний масив між двома частотами MEL. Потім ми перетворимо масив на частотний простір і, нарешті, нормалізуємо масив до розміру FFT і вибираємо пов'язані значення FFT.

Потім ділимо трикутні ваги MEL на ширину смуги MEL (нормалізація площі). Якщо ми не будемо нормалізувати фільтри, ми побачимо, що шум збільшується з частотою через ширину фільтра.

### Генерація кепстральних коефіцієнтів

Останнім кроком у генерації MFCC є використання дискретного косинусного перетворення (DCT). Ми будемо використовувати DCT-III. Цей тип DCT витягує високочастотні та низькочастотні зміни в сигналі

## **Вилучення ознак аудіофайлів за допомогою MFCC**

Реалізовано метод для вилучення ознак з об’єктів дата сету з відповідними ідентифікатора. Також виконується перевірка - чи збігаються ідентифікатори, отримані за допомогою функції extract\_features, з фактичними ідентифікаторами в train\_data, які були створені раніше, щоб не було невідповідності між рядками даних.

## **Компонування тренувального набору даних**

X\_train=np.array(features\_train)

Y\_train=train\_data.species

le=LabelEncoder()

temp=le.fit\_transform(Y\_train)

Y\_train=temp.reshape(-1,1)

## **Компонування випробувального набору даних**

X\_test=np.array(features\_test)

Y\_test=test\_data.species

le=LabelEncoder()

temp=le.fit\_transform(Y\_test)

Y\_test=temp.reshape(-1,1)

## **Побудова моделі класифікації звуку**

Для моделювання ми будемо використовувати згорткову нейронну мережу з 4 шарами. Щоб модель була простою, ми почнемо зі створення послідовної послідовної моделі. Додаємо 4 щільних шари. Перші 3 з розміром вихідного простору 200, функція активації = 'relu'. Останні 4 з розміром вихідного простору як 1, функція активації = 'sigmoid ' Функція випрямленої лінійної активації або скорочено ReLU є кусково-лінійною функцією, яка виводить вхідний сигнал безпосередньо, якщо він позитивний, інакше він виводить нуль.

## **Перевірка моделі**

Для перевірки точності моделі генеруємо матрицю помилок

## Висновок