Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**«Разработка нейросетевых систем»**

**Домашнее задание №2**

**ИСПОЛНИТЕЛЬ:**

Демирев Н.К.

Группа ИУ5-21М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2023 г.

Москва 2023

# Задание

Необходимо создать telegram-бота для шумоподавления звуковых сэмплов. Боту отправляется зашумленная аудиозапись, в ответ бот отправляет аудиозапись с удаленным шумом.

# Выполнение

Изначальные файлы для работы:

denoise.pt – модель автоэнкодера с сохраненной архитектурой и весами

Код бота:

import telebot

import numpy as np

from scipy import signal

from scipy.io import wavfile

import numpy as np

from IPython.display import Audio

import torch

from tqdm.auto import tqdm

import ffmpeg

import os

from scipy import signal

from scipy.io import wavfile

bot = telebot.TeleBot('6387696681:AAFwb3KJ\_Ol3re6\_rhikFKsly\_pWS6HhP8I')

@bot.message\_handler(content\_types=['voice'])

def get\_message(message):

    file\_info = bot.get\_file(message.voice.file\_id)

    downloaded\_file = bot.download\_file(file\_info.file\_path)

    with open('new\_file.ogg', 'wb') as new\_file:

        new\_file.write(downloaded\_file)

    os.system('C:/Users/hae19/ffmpeg.exe -y -i new\_file.ogg -loglevel quiet -ac 1 -ar 16000 audio.wav')

    fs, data\_noised = wavfile.read('audio.wav')

    data\_noised = data\_noised / (2\*\*16-1)

    device = torch.device('cpu')

    model = torch.load('denoise.pt', map\_location=device)

    model.eval()

*# noise = model(data)*

    \_, \_, Zxx = signal.stft(data\_noised, fs=fs, nperseg=512)

    \_, xrec = signal.istft(Zxx, fs)

    X = np.concatenate([np.real(Zxx).T[:,:,None],

                    np.imag(Zxx).T[:,:,None]], axis=-1)

    normalization = X.reshape(-1, 2).std()

    X /= normalization

    tensor\_x = torch.Tensor(np.transpose(X, [0, 2, 1])) *# channels first*

    pred = []

    batch\_size = 128

    for i in tqdm(range(tensor\_x.shape[0]//batch\_size+1)):

        preds = model(tensor\_x[i\*batch\_size:(i+1)\*batch\_size].to(device))

        pred.append(np.transpose(preds.detach().cpu().numpy(),

                                [0, 2, 1])\*normalization)

    pred = np.concatenate(pred, axis=0)

    \_, xrec = signal.istft((pred[:,:,0]+pred[:,:,1]\*1j).T, fs)

    xrec = data\_noised - xrec[:data\_noised.size]

*# preds = model(tensor\_x.to(device)).detach().cpu().numpy()*

*# \_, xrec = signal.istft((preds[:,:,0]+preds[:,:,1]\*1j).T, fs)*

*# xrec = data\_noised - xrec[:data\_noised.size]*

*# print(preds)*

    audio = Audio(xrec, rate=fs)

    with open('output.wav', 'wb') as f:

        f.write(audio.data)

    bot.send\_audio(message.chat.id, audio=open('output.wav', 'rb'))

bot.polling(none\_stop=True, interval=0)

Ввод/Вывод

