소음 환경에 강한 내성을 가진 음성인식 모델 개발

분과: D (지능형융합보안)

Team: 포네틱 코드

부산대학교 정보의생명공학대학 정보컴퓨터공학부

College of Information and Biomedical Engineering

Computer Engineering Major

Pusan National University

2021년 7월 30일

지도교수: 백윤주 (인)

목 차

1. 과제 배경 및 목표

1.1 과제 배경

음성은 사람 간의 가장 자연스러운 의사소통 방식이다. 음성인식 기술은 이미 50년 전부터 지속적인 연구가 이루어지고 있는 분야이다. 특히 종래의 음성인식 기술은 낭독체 음성인식 기술이 주로 연구 대상이었으나, 딥러닝 및 잡음처리 기술의 발전으로 인해 현재는 사람 간의 자연스러운 대화 음성을 대상으로 기술 고도화가 이루어지고 있다. 특히 최근 몇 년 동안 보편화 되는 클라우드 서버 및 고성능 GPU와 같은 하드웨어의 눈부신 발전에 힘입은 딥러닝 기술에 의해, 혁신적인 성능을 보이는 음성인식 기술이 등장하고 있다. 음성인식 기술은 스마트폰, 자동차, 콜 센터 등 현재 우리 생활의 많은 부분에 녹아들어서 서비스화 되고 있다.

1.2 기존 문제점

이러한 음성인식 기술의 비약적 발전에도 불구하고 여전히 음성인식은 사람들이 웅성거리는 식당, 회의실, 버스나 지하철 등과 같은 소음이 발생하는 환경에서는 매우 낮은 정확도를 보이는 등 개선할 점이 많다. 실생활에서 음성인식 기술이 원활하게 사용되려면 실생활에서 자주 발생하는 소음에 대한 내성이 있어야 할 것이다. 이에, 우리는 딥러닝 기술 등을 활용하여, 소음이 존재하는 환경에서의 원활한 음성인식을 제공하는 기술 및 서비스를 개발하고자 한다.

1.3 과제 목표

본 졸업 과제는 소음이 학습된 음성인식 모델을 사용해 다양한 소음 환경에서도 정상적으로 작동하는 음성인식 프로그램을 만드는데 목표를 둔다.

- 소음이 있는 다양한 음성 파일을 제작
  - ▶ 음성 파일과 소음 파일을 합성시킨다.
  - ▶ 소음 파일은 실생활에서 얻을 수 있는 소음이거나 인위적으로 만든 소음으로 한다. (Google Audio Set 등의 Dataset 이용)

- 다양한 소음 환경 중 어떤 소음이 음성인식을 가장 어렵게 하는지 판단
  - ▶ Google STT나 Kakao STT 등 음성 파일을 텍스트로 변환시켜주는 API 및 서비스를 이용해 합성된 음성 파일을 얼마나 잘 인식하는지 확인한다.
  - ▶ 결과를 확인하고 인식에 가장 취약한 소음을 찾는다.

- 취약한 소음 환경에 내성을 가진 모델을 제작
  - ▶ TensorFlow를 이용해 다양한 합성 데이터 파일로 학습을 시키고 모델을 제작한다.

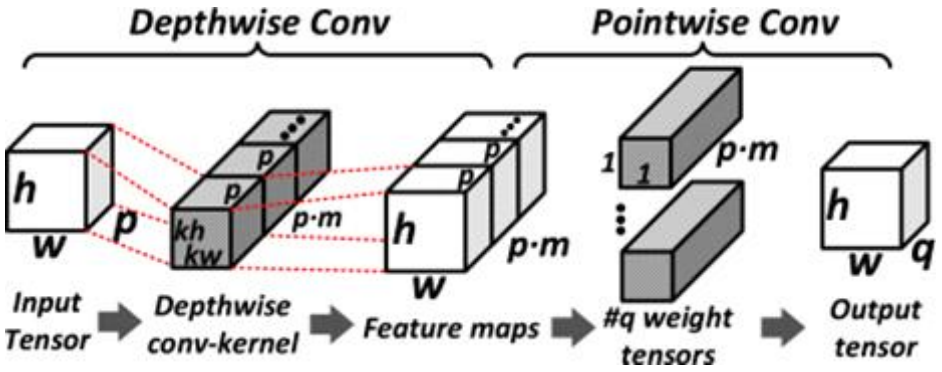
- 음성인식 프로그램을 만들어 실시간으로 사용할 수 있게끔 구현
  - ▶ 학습된 모델을 임베디드 기기에 적용하여 실시간 소음 내성 음성인식이 가능하도록 구현한다.
  - ▶ 실생활에서 인식이 잘 되는지 확인한다.

2. 설계 및 구현

2.1 TensorFlow

본 졸업 과제 프로젝트에서 사용할 머신러닝 프레임워크로 TensorFlow와 Pytorch 중에서 TensorFlow로 결정하였다. TensorFlow는 Google 브레인 팀에서 개발한 다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 End-to-End 오픈소스 소프트웨어 플랫폼이다. 심볼릭 수학 라이브러리이자, 딥러닝, 머신러닝 등의 기계학습 응용 프로그램에도 사용되는 TensorFlow는 Python, C++, Java 등 다양한 언어에 대해 API를 지원하기 때문에 사용 언어에 크게 제약을 받지 않아서 전 세계적으로 활발하게 쓰이고 있다.

즉각적인 '모델 반복' 및 쉬운 디버깅 기능을 가능하게 하는, 즉시 실행 기능이 포함된 Keras와 같은 API를 사용하여 ML 모델을 쉽게 빌드하고 학습시킬 수 있다. 데이터 플로우 그래프를 사용하여 시각화하기 편하고 다양한 추상화 라이브러리와 혼용해서 사용 가능하다는 장점이 있다. 또한, 유연한 아키텍처로 구성되어있어 코드의 수정 없이 데스크탑, 서버 혹은 모바일 디바이스에서 CPU나 GPU를 사용하여 연산을 구동할 수 있다.



[그림 1] Data-flow for depthwise separable convolution layer

본 졸업 과제 프로젝트에서 구성할 소음 내성 음식 인식 모델은 DS-CNN 모델을 통하여 학습하고

구성한다. DS-CNN(Depthwise Separable Convolution Neural Network) 모델은 합성곱신경망(CNN)의 학습 모델방식 중에서 Depthwise Convolution의 방식과 Pointwise convolution의 방식을 합성한 기법이다.

2.2 특정 소음에 강한 모델 제작

소음 내성 음성인식 모델을 구현하는 데에 있어서, 모델에 특정 소음 종류를 학습시키는 것이 좀 더 효과적인 학습을 통해 뛰어난 성능을 발휘할 수 있다. 이에 우리는 상대적으로 기존 음성인식 서비스의 성능을 저해하는 소음 종류를 선별하여 해당 종류의 소음에 대한 집중적인 학습을 통해 특정 소음에 대해 특화된 내성을 지닌 소음 내성 음성인식 모델을 개발하고자 한다. 모델은 앞서 언급한 TensorFlow 머신러닝 프레임워크를 통해 제작한다.

2.3 Dataset 수집 및 활용

모델에 이용될 Dataset의 수집 및 활용은 최대한 데이터의 객관성 및 대표성, 비편향성, 선형성, 효율성 등을 고려한다. 이에 우리는 Google Audio Set 등의 상대적으로 앞서 언급한 데이터의 객관성 등이 상당히 확보된 Dataset을 소음 데이터로 사용하고, 마찬가지로 Google Command Dataset과 같은 Dataset을 음성 데이터를 사용한다. 이를 통해 개발의 편의성 및 데이터의 객관성 모두 확보할 수 있다. 또, 학습된 모델을 바탕으로 실제 테스트를 수행 후, 결과에 따라 Dataset의 투입 수준 등을 조정하여 모델을 수정 및 개선한다.

2.4 임베디드 기기 적용

최종적으로 완성된 모델을 임베디드 기기에 적용한다. 임베디드 기기는 라즈베리파이로 한다. 실생활에서 임베디드 기기가 사용됨을 고려하여, 임베디드 기기의 스피커 및 마이크 센서의 인식률, 사용 편의성, 휴대성 등을 고려하여 설계한다. 앞서 언급하였듯이, 임베디드 환경임을 고려하여, 모델은 CNN보다 상대적으로 임베디드 환경에서 구동하기에 연산이 효율적인 DS-CNN을 이용하여 구성한다.

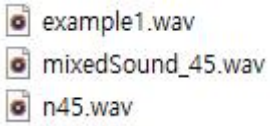
3. 보고 시점까지의 과제 수행 내용 및 중간 결과

3.1 음성과 소음 합성으로 Dataset 및 이미지 구성

```python
from pydub import AudioSegment

##import soundfile as sf
##from pydub.utils import mediainfo

SAMPLING_RATE = 16000
MONO = 1

number = 45

OUTPUT_FILE = "./mixedSound_"+str(number)+".wav"
NOISE_FILE = "./n"+str(number)+".wav"
ORIGINAL_FILE = "./example1.wav"

sound1 = AudioSegment.from_wav(NOISE_FILE)
sound2 = AudioSegment.from_wav(ORIGINAL_FILE)

combined_sounds = sound1 + sound1 + sound1 + sound1 + sound1 # 반복 횟수 조절
combined_sounds = combined_sounds - 15   # dB 조절 (+ $$$), (- $$$)
output = sound2.overlay(combined_sounds)
output = output.set_channels(MONO) # stereo to mono
output = output.set_frame_rate(SAMPLING_RATE) # frame_rate : 16000

output.export(OUTPUT_FILE, format="wav")

# check frame_rate
from scipy.io.wavfile import read as read_wav
import os
os.chdir('./') # path
sampling_rate, data = read_wav(OUTPUT_FILE)
print("Frame_Rate : " + str(sampling_rate))

# check channels
import wave
f1 = wave.open(OUTPUT_FILE, 'r')
print("Channels : " + str(f1.getnchannels()))
```

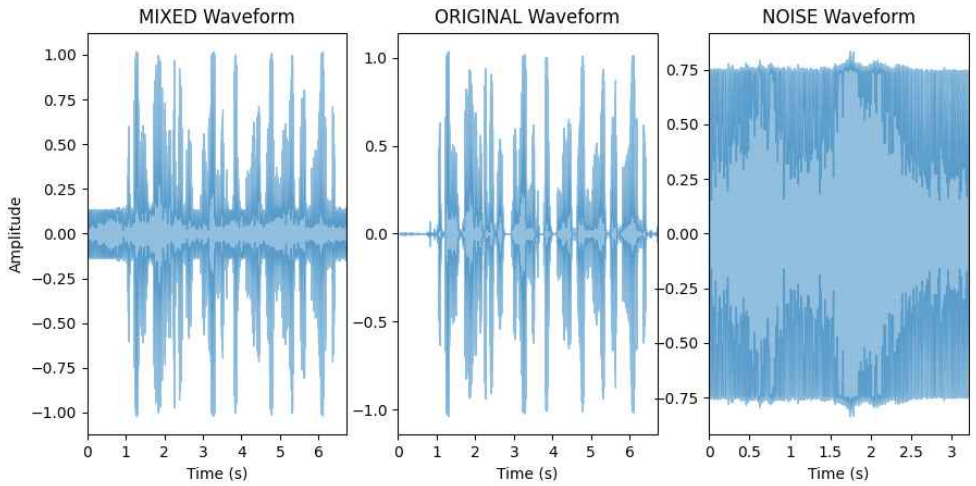
[그림 2] 음성 파일(ORIGINAL_FILE)과 소음 파일(NOISE_FILE) 합성 코드

Python 라이브러리 중 하나인 pydub을 통하여 음성 파일과 소음 파일을 합성하여 모델에 투입할 Dataset을 구성하였다. SAMPLING_RATE 변수로 샘플링 레이트를 조정할 수 있으며, 소음이나 음성의 크기를 각각 개별적으로 조정하여 합성할 수 있다. 또, 이러한 합성 결과를 wave 라이브러리를 통해 이미지로 확인할 수 있고 향후 모델에 이미지 형태로 투입되어 학습될 수 있다. 이러한 방법으로, 앞서 언급한 Google Audio Set, Google Command Dataset 등을 활용 및 합성하여 Dataset을 구성한다.



[그림 3] 합성파일(mixedSound_45.wav)이 생성된 결과

[그림 4]
Frame_Rate와 Channels를 알 수 있음



[그림 5] 원본 파일, 소음 파일, 합성 파일의 Waveform을 확인할 수 있다.

3.2 합성된 음성 파일 테스트

어떤 소음 종류가 음성인식에 상대적으로 가장 오류를 불러일으키는지 알아보기 위해, 우리는 앞서 구축된 Dataset을 활용, 여러 가지 소음 종류에 대하여 Kakao Cloud STT API 및 PNL 100 Nonspeech Sounds Dataset을 이용하여 파일을 합성 및 테스트하였다. PNL 100 Nonspeech Sounds Dataset는 대중 소음, 기계 소음, 사이렌 소음, 교통 소음 등 다양한 소음으로 구성된 Dataset이다.

- N1-N17: Crowd noise
- N18-N29: Machine noise
- N30-N43: Alarm and siren
- N44-N46: Traffic and car noise
- N47-N55: Animal sound
- N56-N69: Water sound
- N70-N78: Wind
- N79-N82: Bell
- N83-N85: Cough
- N86: Clap
- N87: Snore
- N88: Click
- N88-N90: Laugh
- N91-N92: Yawn
- N93: Cry
- N94: Shower
- N95: Tooth brushing
- N96-N97: Footsteps
- N98: Door moving
- N99-N100: Phone dialing

[그림 6] PNL 100 Nonspeech Sounds

이러한 음성과 소음의 합성 파일을 Kakao Cloud STT API에 입력한 결과에 따르면, 교통 소음에 대한 인식률이 상대적으로 현저히 떨어지는 것으로 여겨진다. 이에 향후 교통 소음 및 음성인식에 어

려움을 주는 소음 종류에 강건한 음성인식 모델을 구성하여 실생활에 활용할 수 있는 시나리오를 구성하고 해당 시나리오에 맞게 구현한다.

```
function init(fileName){
  let headers = {
    'Transfer-Encoding': 'chunked',
    'Content-Type': 'application/octet-stream',
    'Authorization': `KakaoAK ${process.env.API_KEY}`
  };

  let options = {
    url: 'https://kakaoi-newtone-openapi.kakao.com/v1/recognize',
    method: 'POST',
    headers: headers,
    body: fs.createReadStream(`./sound/${fileName}.wav`)
  };

  request(options, callback);

}
```

{"type":"finalResult","value":"국감 때 내가 하고 위험으로부터 국민을 보험 위하여 노력하여야 한다"
,"nBest":[{"value":"국감 때 내가 하고 위험으로부터 국민을 보험 위하여 노력하여야 한다","score":0}
,{"value":"국감 대리도 하고 위험으로부터 국민을 보험 위하여 노력하여야 한다","score":0},{"value":
"국감 때 내가 하고 위험으로부터 국민의 구원을 위하여 노력하여야 한다","score":0},{"value":"국감
대리도 하고 위험으로부터 국민의 구원을 위하여 노력하여야 한다","score":0},{"value":"국감 때 내가
하고 그 영으로부터 국민을 보험 위하여 노력하여야 한다","score":0},{"value":"그가 매일 이동하고 위
험으로부터 국민을 보험 위하여 노력하여야 한다","score":0},{"value":"국감 때 내가 하고 그 영으로부
터 국민의 구원을 위하여 노력하여야 한다","score":0},{"value":"그 간의 대립이 강하고 위험으로부터
국민을 보험 위하여 노력하여야 한다","score":0},{"value":"그가 매일 이동하고 위험으로부터 국민의 구원을 위하여 노력하여야 한다","score":0},{"value":"그 간의 대립이 강
하고 위험으로부터 국민의 구원을 위하여 노력하여야 한다","score":0}],"voiceProfile":{"registered":false,"authenticated":false},"durationMS":6940,"qmarkScore":0,"gende
r":0}
------newtoneV8tWItSKOBNAgreT--

[그림 7, 8] Kakao STT API 요청 및 결과

3.3 TensorFlow Audio Recognition 예제

[그림 9] Tensorflow Tutorial에서 간단한 키워드 인식을 실습

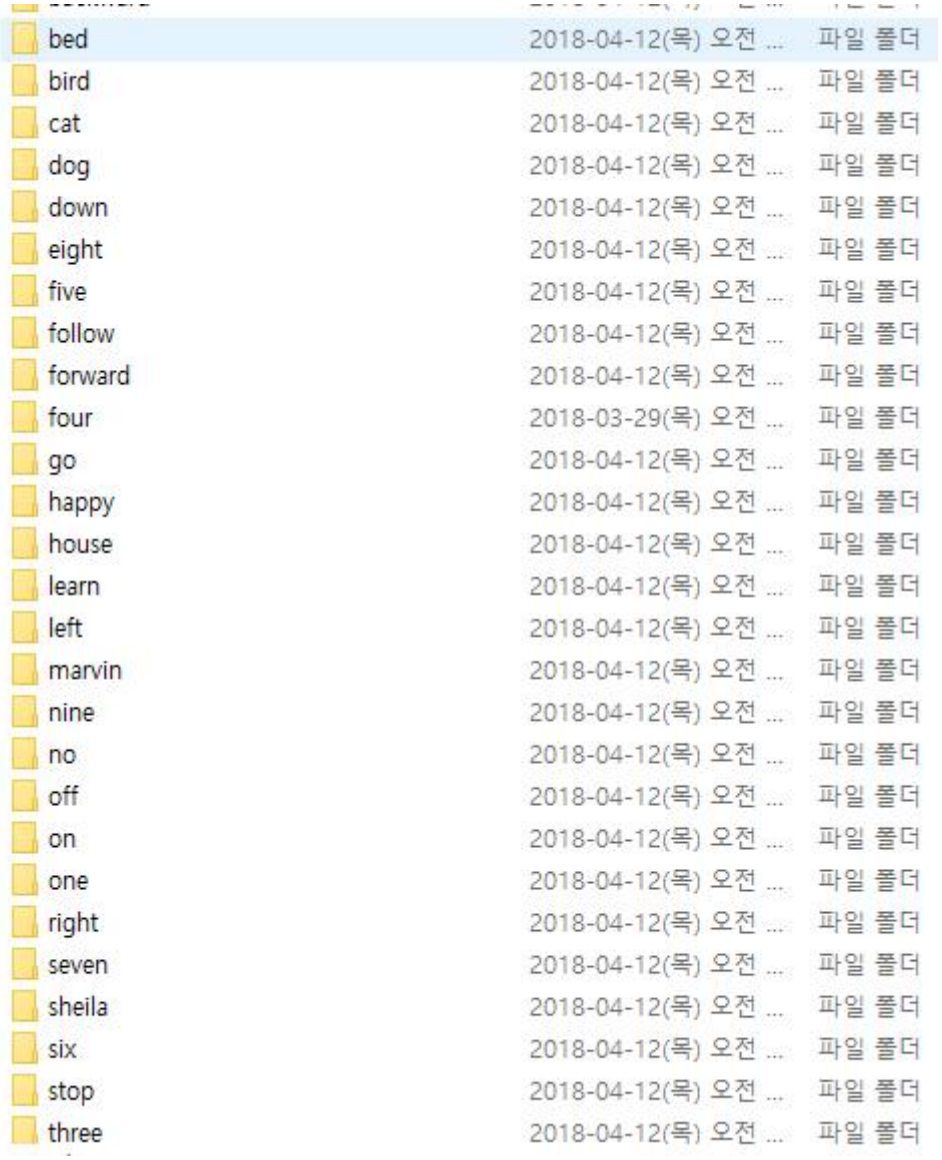
본격적인 모델 학습 및 구현에 앞서, TensorFlow의 Audio Recognition 예제를 진행하여 머신러닝에 대한 기초적인 이해와 학습을 진행하였다. 예제는 TensorFlow 공식 Document에서 제공하는 내용이며, 간단한 키워드를 인식한다. 몇 개의 후보 단어를 입력 후, 오디오 파일을 투입하면 입력된 후보 단어 중에 가장 유사한 단어를 예측하는 모델을 구성하는 예제이며, 단어로는 간단한 명령어에 해당하는 no, yes, go, stop 등으로 구성되어있다.

3.4 적합한 시나리오 구상

[그림 10] 휠체어 리프트

앞서 언급하였듯이, 특정 소음 종류에 대한 내성을 지닌 음성인식 모델을 구현하는 경우가 더욱 정교하고 효과적인 성능을 지닐 것이기 때문에 어떠한 소음 종류로 소음 내성 음성인식 모델을 구현할지 시나리오를 구상하였다. 앞서 Kakao Cloud STT API 및 PNL 100 Analysis Dataset으로 실험한 결과, 상대적으로 교통 소음에 대한 음성인식 기술의 강건성이 부족하다고 판단, 우리는 교통 소음에 대해 강건성 있는 소음 내성 음성인식 모델을 구성하고자 한다. 이에 구체적인 시나리오로서, 「지하철 역사의 소음 내성 음성인식 휠체어 리프트」를 구상하였다.

휠체어 리프트에 음성인식을 적용한다면, 손이나 발로 조작할 필요 없이 좀 더 편하게 이용이 가능할 것이다. 그러나, 지하철 역사의 입구는 대부분 대로변에 존재하여 교통 소음에 노출되어 있는데, 이러한 환경에서 원활한 음성인식을 위하여 교통 소음에 대한 내성을 지닌 모델을 휠체어 리프트의 음성인식 기기에 적용하고자 한다. 이러한 휠체어 리프트에는 forward / backward, go / stop, up / down 등의 명령어가 필요한데, 이러한 명령어는 앞서 언급한 Google Command Dataset을 활용할 수 있다.



[그림 11] Google Speech Command Dataset 0.02V

Google Command Dataset은 수 천명의 각기 다른 사람의 음성으로 구성된 명령어 Dataset으로 객관성, 대표성, 비편향성 등을 지닌 Dataset이라고 할 수 있다. 앞서 언급한 휠체어 리프트에 필요한 명령어 모두 Dataset으로 존재하며, 이에 이를 활용하면 개발 편의성 또한 확보할 수 있다.

4. 구성원별 진척도 및 개발 일정

7월				8월				9월				
1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	5주
시나리오 구상												
	중간보고서 준비											
		소음 내성을 가진 학습 모델 제작										
			임베디드 환경 적응									
			서버 환경 구축									
				설계 문서 수정								
					정확도 평가							
					테스트 및 디버깅							
								최종 발표 및 보고서 준비				

[표 1] 개발 일정표

이름	역할 분담
안준수	- 음성 파일 녹음 및 합성 완료 - Dataset 준비 완료 - 라즈베리파이 환경 적응 중
강동민	- API를 이용해 음성인식 테스트 완료 - 서버 개발 중 - Tensorflow 관련 모델 제작 중
공통	- 전반적인 지식 이해 - 시스템 테스트 - 성능 평가 - 보고서 작성 - 발표 및 시연 준비 - Git을 이용한 버전 관리

[표 2] 구성원 역할 분담표

[Reference 및 출처]

<Cloud STT API>
Google Cloud STT: https://cloud.google.com/speech-to-text/pricing?hl=ko
KAKAO NEWTON STT: https://developers.kakao.com/docs/latest/ko/voice/rest-api
Microsoft Azure STT: https://docs.microsoft.com/ko-kr/azure/cognitive-services/speech-service/rest-speech-to-text

<Data Samples>
PNL 100 Nonspeech Sounds: http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html
AudioSet: https://research.google.com/audioset/
FSD50K: https://zenodo.org/record/4060432#.YJTrJrUzaUI
Google Speech Commands Dataset: Pete Warden, Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition, 2018. 04.

<Python Library>
librosa: https://librosa.org/

[그림 1] 출처 : Li Yang, Binarized Depthwise Separable Neural Network for Object Tracking in FPGA, 2019. 05.
[그림 10] 출처 : http://www.imedialife.co.kr/