1st Implementation Demo

열 려 라 , 쿠 (: K U) 굴

201411303 이준호
201511243 김동언
201511262 박우진
201714160 권혜주

INDEX

열려라, 쿠(:KU)굴

프로젝트 소개 : 설명

열려라, 쿠(:KU)굴

'알리바바와 40인의 도둑'에서 "**열려라 참깨**" 라는 명령어에 대한
처리 결과로 동굴이 열리는 기능이 수행되었다.

도둑의 우두머리라는 **화자 인식을 하지 않고 명령어만 처리한 결과**
도둑들은 자신의 보물을 알리바바에게 도둑질 당했다.

이 동굴의 시스템의 문제는 **화자의 대한 인식을 하지 못한 것**.
이러한 동굴의 문제점을 보완하고자 화자 인식에 대한 기능을 추가하고자 한다.

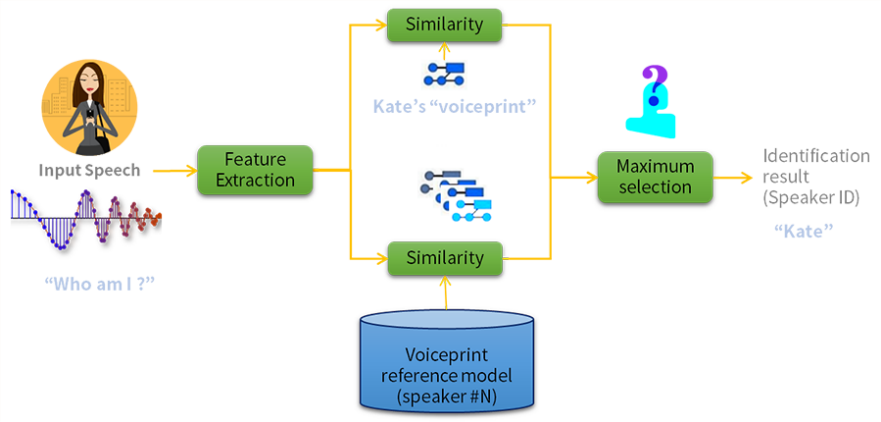
화자를 인식하고 인가된 화자에 대한 **음성 명령을 텍스트로 변환하여 등록된 기능을 수행**하는 것이 목표.

프로젝트 소개 : 설명

음성으로 입력된 신호에서 노이즈와 배경 소리로부터 실제 유효한 소리의 특징을 추출하기위해 ***MFCC***, *STFT* 등의 Feature Extraction을 이용한다. 이후, 데이터를 ***DNN***, *CNN, RNN* 과 같은 Classification(분류기) 모델로 학습을 시킨다.

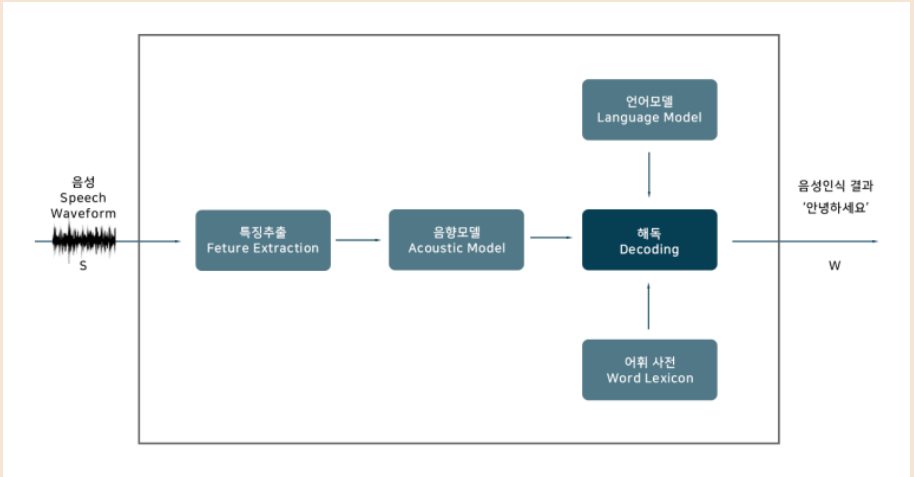
새로운 음성으로 입력된 신호를 마찬가지로 소리의 특징을 추출하고
학습된 모델을 재사용하여 원하는 화자 인식 기능을 수행한다.

프로젝트 소개 : 설명



화자 인식(Speaker Recognition)

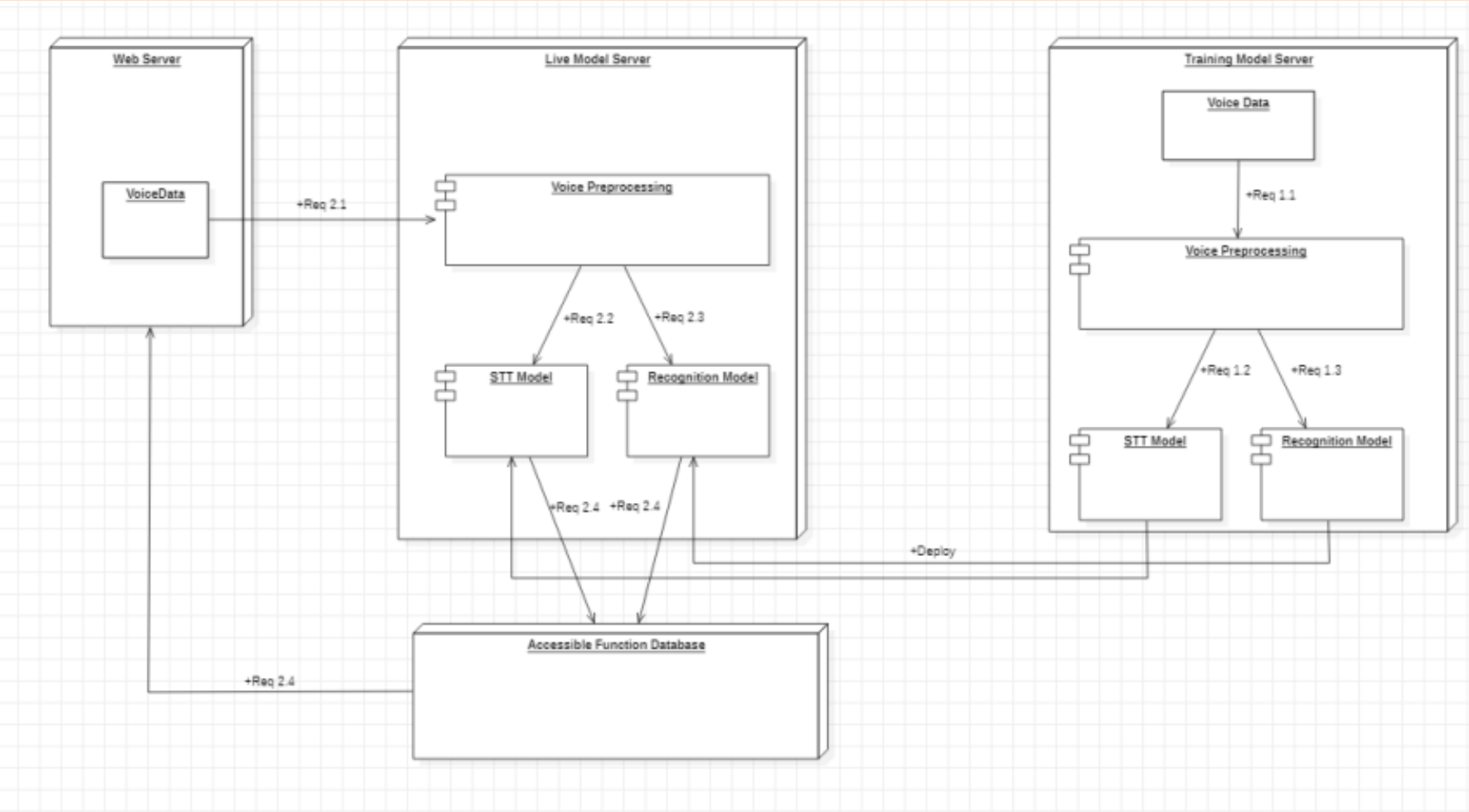
주어진 화자의 음성을 이용하여 음성의 특징을 추출하고 학습된 모델에 반영하여, 등록된 화자를 인식한다.



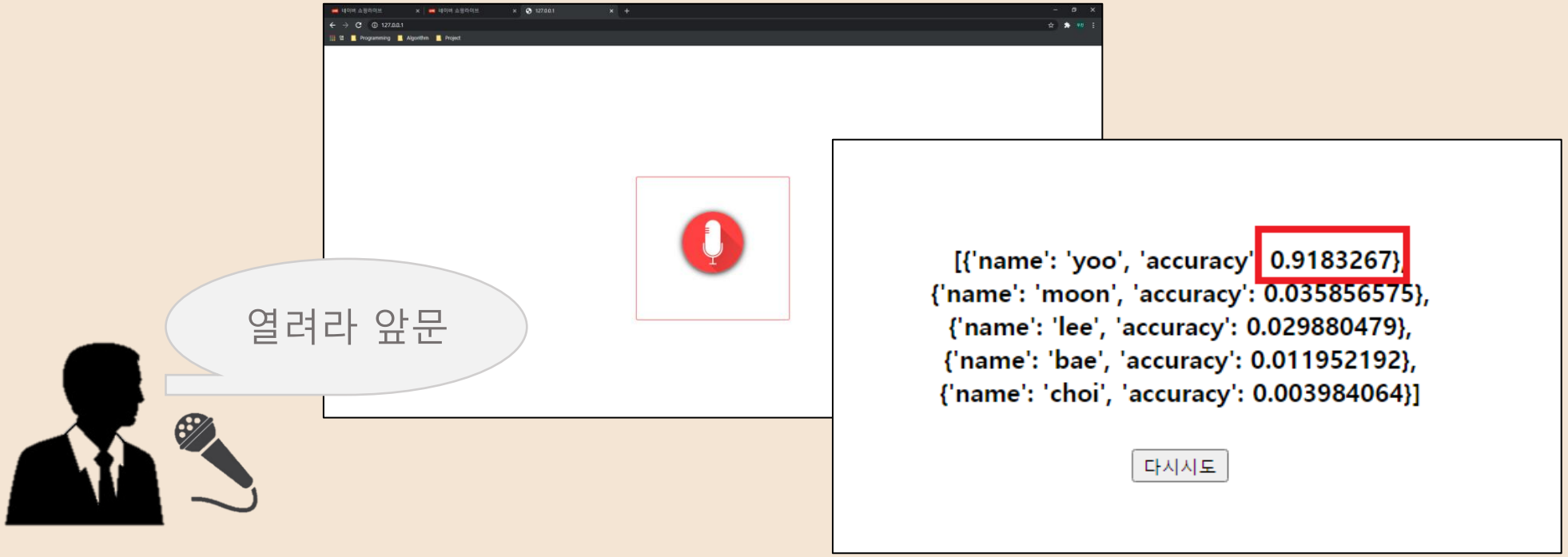
STT(Speech To Text)

주어진 화자의 음성을 이용하여 학습된 모델에 반영하여 음성을 텍스트 데이터로 변환한다.

Component Diagram

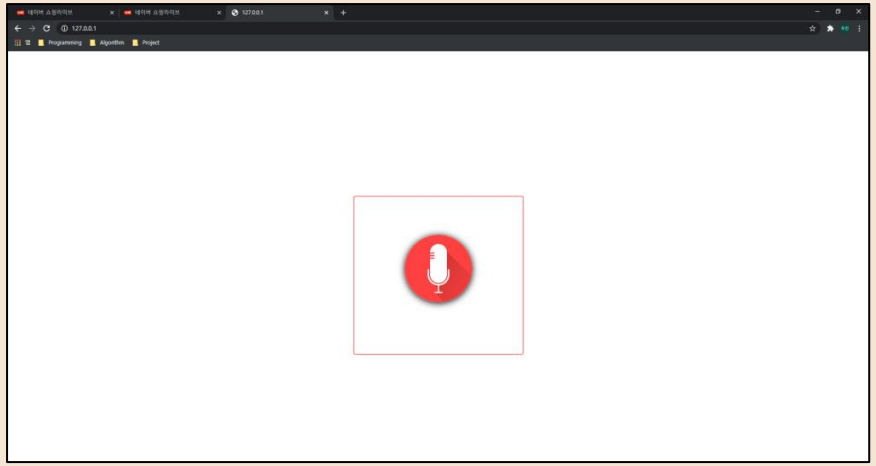


1차 구현물(Web Server)



1. 사용자의 음성 파일을 업로드 한다.
2. 음성 파일 업로드가 성공하면 서버에서 MFCC 특징을 추출하여 npy 파일로 저장한다.
3. 학습된 SR 모델과, STT 모델을 각각 Restore하여 변환 된 npy 파일의 Test를 수행하고,
각각의 결과를 조건에 따라 성공 또는 실패로 분기하여 결과를 반환한다.

1차 구현물 – File Upload

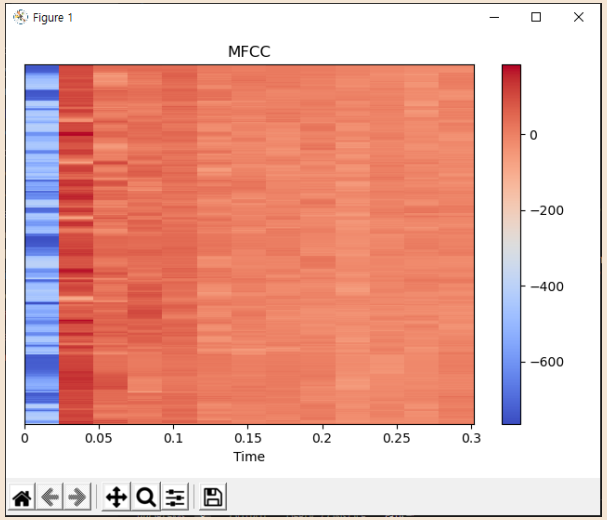


1. 사용자의 **음성을 녹음한다.**
2. 음성의 원본 파일 형식 **WAV 파일**로 저장한다.
3. WAV파일을 서버에 업로드 한다.

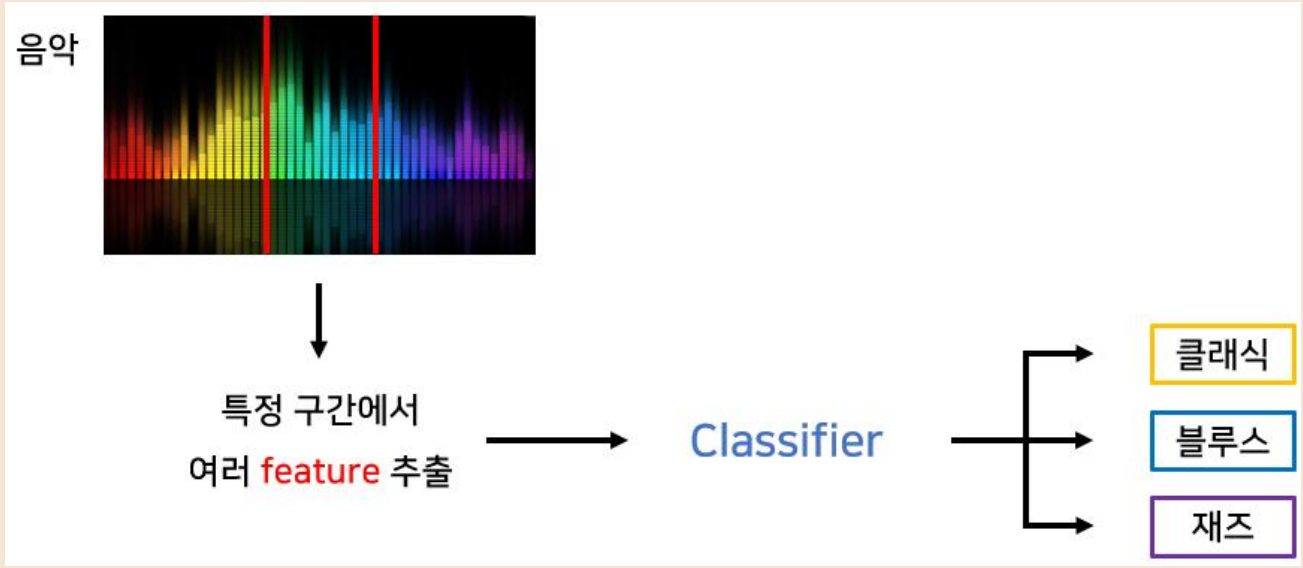
1차 구현물 – Feature Extraction

1. 서버에 업로드 된 음성파일의 **MFCC 특징**을 추출한다.
2. 재사용을 위해 **파일**로 저장한다.

```
2020-09-07 21:43:12.161745: I tensorflow/stream_executor/platform/default/ds
[[-394.75876    127.19304     30.741901  ...  -16.293608    -5.9108253
   12.609705 ]
 [-596.4947     132.00122     90.59825   ...  -14.441874   -30.406517
   -3.1708603]
 [-578.3387     117.72168     91.55144   ...  -19.162102   -32.431072
    0.8641346]
 ...
 [-737.3877     102.25579     43.449986  ...  -17.748533   -19.267479
  -18.709053 ]
 [-738.3183     101.90717     45.06222   ...  -14.985735   -21.655264
  -22.662407 ]
 [-709.28284    117.03282     32.86099   ...   -6.969739   -13.427218
   -6.989418 ]]
```



Feature Extraction (MFCC)



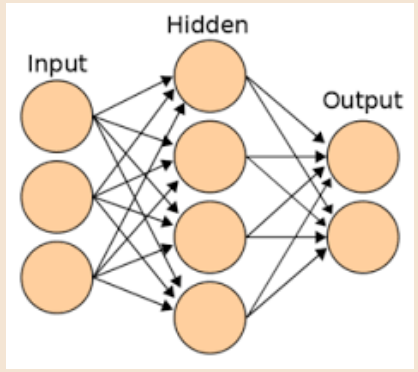
MFCC(Mel-Frequency Cepstral Coefficient)

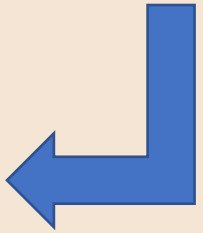
MFCC는 **오디오 신호**에서 추출할 수 있는 feature로, 소리의 고유한 **특징**을 나타내는 수치이다.

오디오 신호를 프레임별(20ms - 40ms 단위)로 나누어 FFT(고속 푸리에 변환)를 적용해 Spectrum을 구하고
Mel Filter Bank를 적용해 Mel Spectrum을 구한 뒤, Cepstral 분석을 적용해 MFCC를 구한다.

1차 구현물 – Predict (Test)

```
2020-09-07 21:43:12.161745: I tensorflow/stream_executor/platform/default/ds
[[-394.75876    127.19304     30.741901  ...  -16.293608    -5.9108253
   12.609705 ]
 [-596.4947     132.00122     90.59825   ...  -14.441874   -30.406517
   -3.1708603]
 [-578.3387     117.72168     91.55144   ...  -19.162102   -32.431072
    0.8641346]
 ...
 [-737.3877     102.25579     43.449986  ...  -17.748533   -19.267479
  -18.709053 ]
 [-738.3183     101.90717     45.06222   ...  -14.985735   -21.655264
  -22.662407 ]
 [-709.28284    117.03282     32.86099   ...   -6.969739   -13.427218
   -6.989418 ]]
```



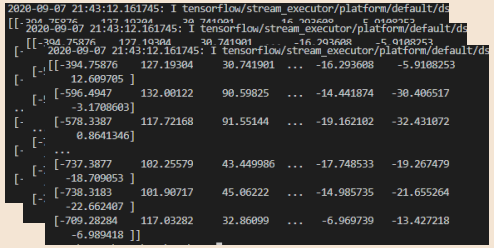


[{'name': 'yoo', 'accuracy': 0.9183267},
{'name': 'moon', 'accuracy': 0.035856575},
{'name': 'lee', 'accuracy': 0.029880479},
{'name': 'bae', 'accuracy': 0.011952192},
{'name': 'choi', 'accuracy': 0.003984064}]

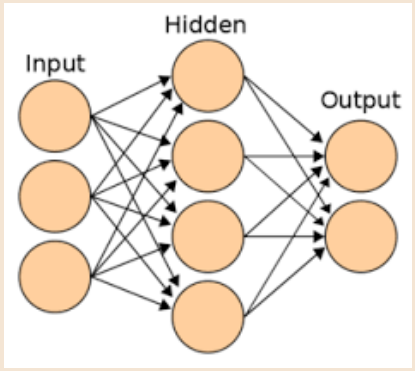
다시시도

1. 학습된 SR 모델과, STT 모델을 각각 Restore한다.
2. MFCC의 결과를 Predict(Test)하여, 각각의 결과를 반환한다.

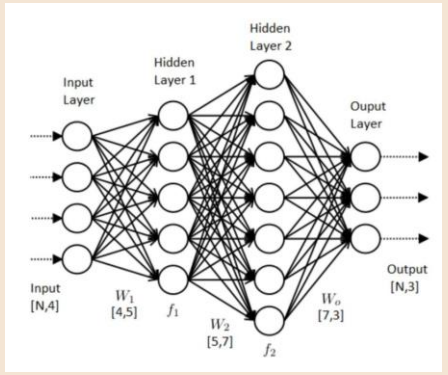
1차 구현물 - Train



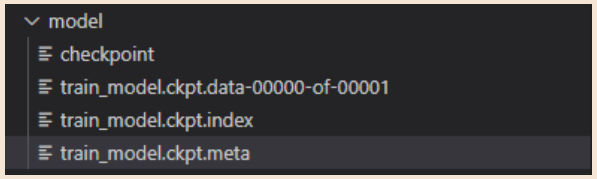
1. 가지고 있는 Dataset을 MFCC 변환한다.
2. 변환된 결과를 각각 SR Model, STT Model 에 Train 하여 그 결과를 저장한다.

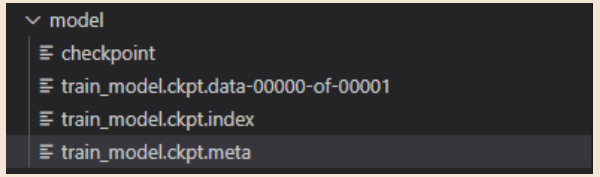


SR Model

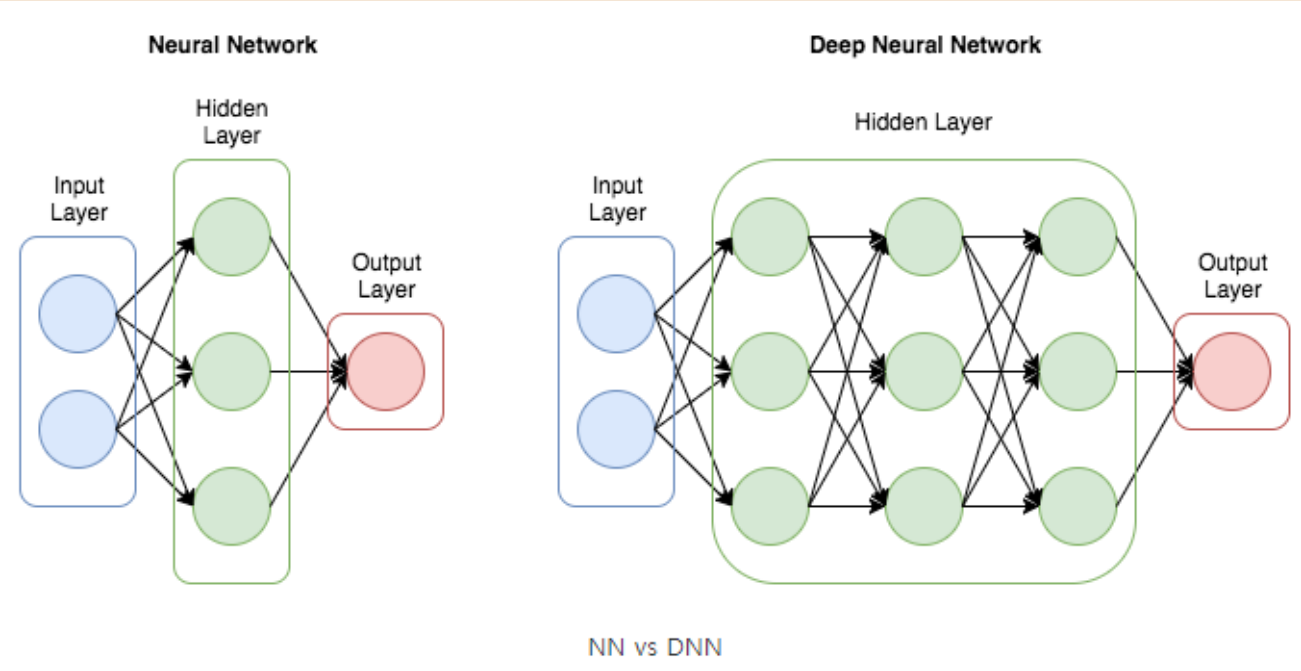


STT Model(예정)





DNN



DNN(Deep Neural Network)이란 **은닉층을 2개 이상** 지닌 학습 방법이다.

DNN은 모델 내 입력층과 출력층 사이에 있는 **은닉층을** 늘려서 **학습의 결과를 향상** 시킨다.

컴퓨터가 스스로 **분류 레이블**을 만들어 내 **공간을 왜곡**하고
데이터를 구분 짓는 과정을 반복하여 **최적의 구분선**을 도출 한다.

모델

SR Model

1. SR에서는 단순히 은닉층을 늘린 DNN을 사용하고 있다. (8개의 Layer)
2. 활성화 함수는 **Relu**를 사용하고 있다.
3. 역전파가 적용된 CNN이나 LSTM, GRU등의 모델을 적용해볼 예정이다.

STT Model (진행 예정)

1. STT에서는 역전파가 적용된 CNN이나 LSTM, GRU등의 모델을 적용할 예정이다.
2. CTC 기법으로 STT의 정확성을 늘릴 예정이다.
3. 문장을 음소단위로 분리하여 학습시킬 예정이다

1차 구현물 한계와 계획

1. Accuracy에 대해 적당한 **Threshold (임계값 – Pass or Fail)** 를 찾지 못했다.

1.1) 현재에는 음성에 대한 정확도의 수치가 결과로 나온다. 하지만 이 수치를 판단하기 위한 값을 지정하지 못하여 많은 테스트를 진행해보고 결정할 예정이다.

1.2) 분류 절차 이후에 추가적인 검증이 필요하다고 판단되어 현재 시스템에 추가할 것을 고려하는 중이다.

2. **SR** 의 경우 **DNN** 으로 구현이 되어있는데 **CNN, LSTM, GRU** 등의 다양한 모델들을 적용해볼 예정이다.

3. **STT** 의 경우 전 페이지에서 언급한 대로 진행할 예정이다.

앞으로의 진행 계획

	9월 2주	9월 3주	9월 4주	9월 5주	10월 1주	10월 2주
SR 다른 모델 적용	Framework 변경 DNN -> CNN, GRU					
STT 학습						
Documents	**SRS-2nd**	**STP**	**SW Design Specification**			**2nd Implementation**

최종 예상 산출물

열려라 앞문

등록되지 않은 유저 A

열려라 앞문

등록된 유저 B