지역별 음성 데이터 학습에 따른 방언 구별 프로그램

[기계학습]

양성호 / 201720721
김용찬 / 201820456
허준형 / 201820664
이상훈 / 201820774
강준형 / 202126906

1

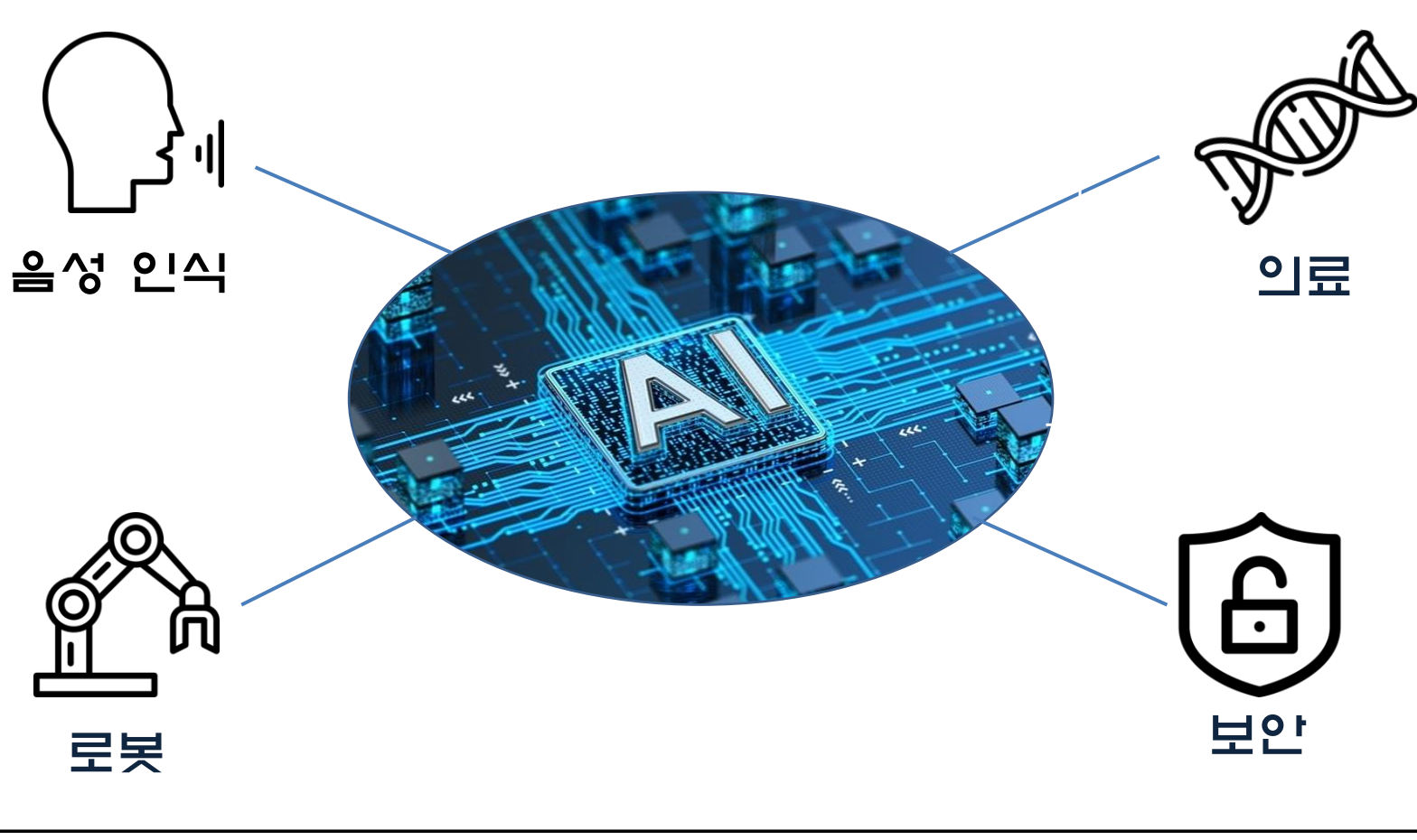
목차

2

/01 프로젝트 개요

3

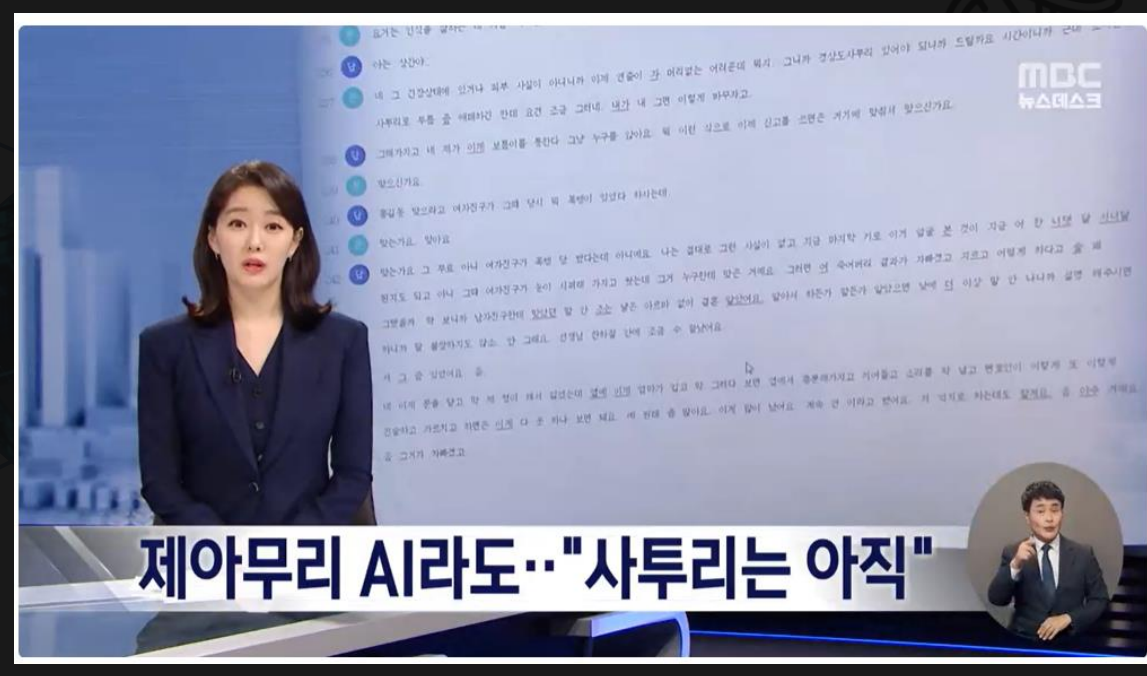
/01. 프로젝트 개요



4

/01. 프로젝트 개요

음성 인식



로봇

보안

https://www.youtube.com/watch?v=5ERdj7B-S5c

5

/01. 프로젝트 개요

음성 인식

문제

방언 사용자의
낮은 음성 인식률

어휘, 음운, 빠르기

일상의 불편함

음성 인식 서비스 장애

해결책

학습

“ 표준어 ”

+

“ 방언 ”

로봇

의료

보안

6

/01. 프로젝트 개요

01 다양한 음성 데이터 수집 및 분석

02 적절한 전처리 수행

03 기존 선행 연구결과와 비교

7

/02 데이터 구축

8

/02. 데이터 구축

AI Hub

- ☑ 한국어 방언 발화 음성
- ☑ 한국인 대화 음성


```
"id": "DJES20000098",
"metadata": {
    "title": "DJES20000098",
    "creator": "솔트룩스",
    "distributor": "솔트룩스",
    "year": "2020",
    "category": "제주방언 > 사적 대화 > 일상 대화",
    "annotation_level": [
        "원시"
    ],
    "sampling": "본문 전체",
    "author": "개인 발화자",
    "publisher": "개인 발화 녹음",
    "date": "20201208",
    "topic": "자동차/오토바이"
},
"speaker": [
    {
        "id": "1",
        "name": "JJ10598",
        "age": "20대",
```

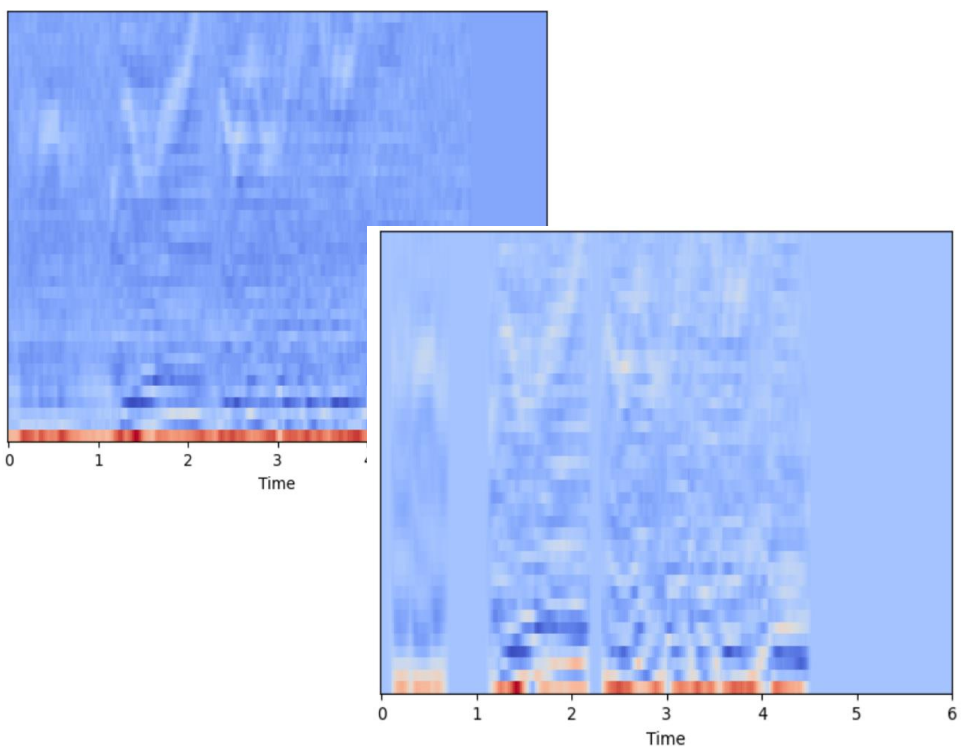
```
"speaker": [
    {
        "id": "1",
        "name": "JJ10598",
        "age": "20대",
        "sex": "여성",
        "occupation": "학생",
        "birthplace": "경기",
        "principal_residence": "제주",
        "current_residence": "제주",
        "education": "대재"
    },
    {
        "id": "2",
        "name": "JJ10599",
        "age": "20대",
        "sex": "남성",
        "occupation": "무직/취업준비생",
        "birthplace": "제주",
        "principal_residence": "제주",
        "current_residence": "제주",
        "education": "대졸"
```


과제명	데이터 구축량
한국어 방언 발화 데이터 (경상도)	- 조용한 환경에서 2,000명 이상의 화자가 발화한 성별, 연령별 적정 길이의 3,000시간 이상의 음성 데이터셋 - 원본 표준어 텍스트 및 방언 특성을 고려하여 그대로 전사한 텍스트 50만건
한국어 방언 발화 데이터 (강원도)	- 조용한 환경에서 2,000명 이상의 화자가 발화한 성별, 연령별 적정 길이의 3,000시간 이상의 음성 데이터셋 - 원본 표준어 텍스트 및 방언 특성을 고려하여 그대로 전사한 텍스트 50만건
한국어 방언 발화 데이터 (제주도)	- 조용한 환경에서 2,000명 이상의 화자가 발화한 성별, 연령별 적정 길이의 3,000시간 이상의 음성 데이터셋 - 원본 표준어 텍스트 및 방언 특성을 고려하여 그대로 전사한 텍스트 50만건
한국어 방언 발화 데이터 (전라도)	- 조용한 환경에서 2,000명 이상의 화자가 발화한 성별, 연령별 적정 길이의 3,000시간 이상의 음성 데이터셋 - 원본 표준어 텍스트 및 방언 특성을 고려하여 그대로 전사한 텍스트 50만건
한국어 방언 발화 데이터 (충청도)	- 조용한 환경에서 2,000명 이상의 화자가 발화한 성별, 연령별 적정 길이의 3,000시간 이상의 음성 데이터셋 - 원본 표준어 텍스트 및 방언 특성을 고려하여 그대로 전사한 텍스트 50만건

9

/02. 데이터 전처리

- **노이즈 제거**

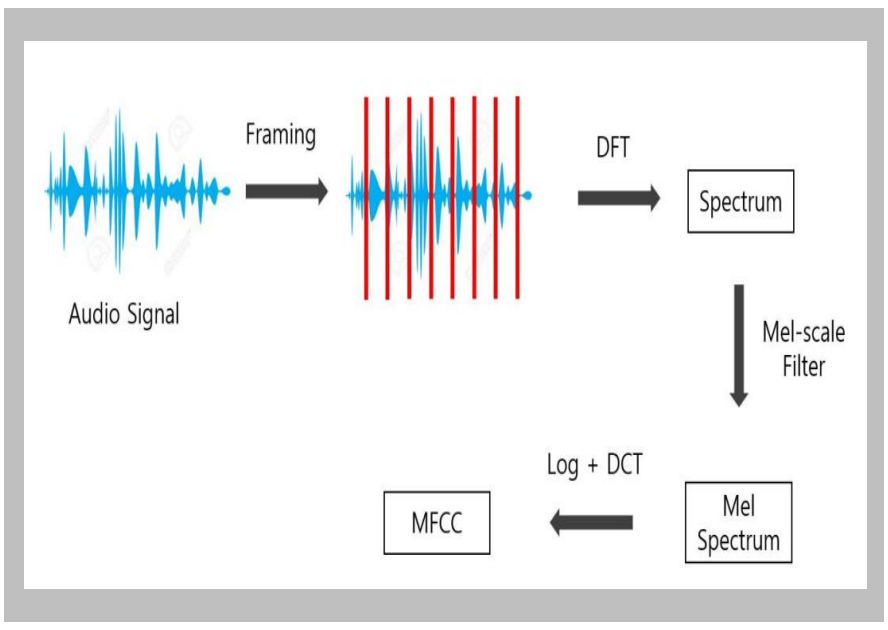



```
############## 노이즈 제거 부분 ##########
wavfileLocation = wavDir + wavFile
rate, wdata = wavfile.read(wavfileLocation);
reduced_noise = nr.reduce_noise(y=wdata, sr=25500);
# wavfile.write("노이즈 제거.wav", rate, reduced_noise);
#####################################

############ 정규화 과정 ###############
# 정규화 수식
def mu_law(x, mu=255):
  return np.sign(x) * np.log(1+mu*np.abs(x))/np.log(1+mu)

# 정규화
x = np.linspace(-1,1,1000)
x_mu = mu_law(x)

# 정규화
norm_wav = reduced_noise / np.max(np.abs(reduced_noise))

# 정규화된 데이터를 정수형으로 변환
norm_wav_int = np.int16(norm_wav * 32767)

# 정규화된 WAV 파일 저장(파괴적 저장/기존 파일 백업 필요)
wavdnfileLocation = wavDenoiseDir + wavFile
wavfile.write(wavdnfileLocation, rate, norm_wav_int)
#####################################
```

10

/02. 데이터 전처리

- MFCC

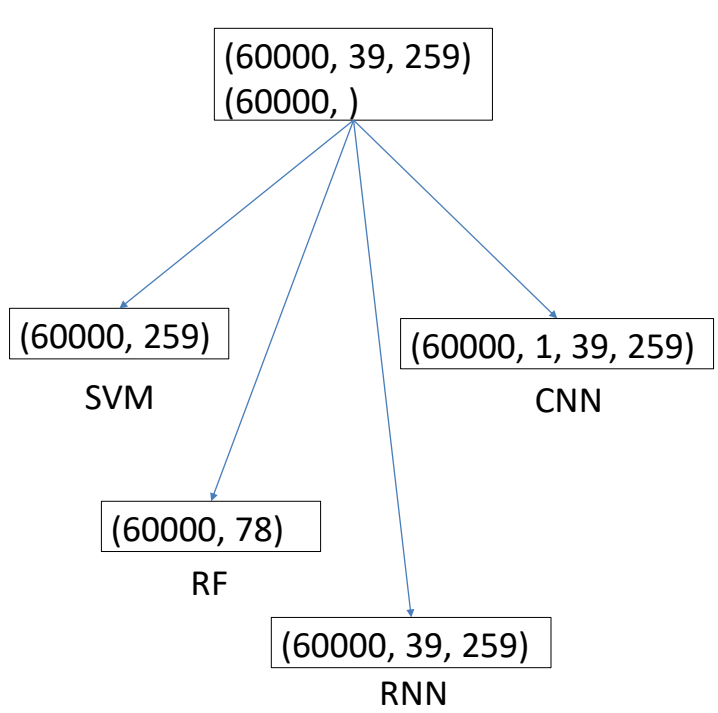



```
# MFCC 추출을 위한 파일 로드
wavLocation = wavDenoiseDir + wavFile
y, sr = librosa.load(wavLocation, offset=start_utter, duration=end_utter-start_utter)

# MFCC 추출
mfcc = librosa.feature.mfcc(y=y, sr=22050, n_mfcc=40)
mfcc = mfcc[1:]

# MFCC 길이 패딩(SpeakUP)
padding = lambda a, i: a[:, 0:i] if a.shape[1] > i else np.hstack((a, np.zeros((a.shape[0], i-a.shape[1]))))
mfcc_pad = padding(mfcc, 259)

# 서브데이터와 라벨데이터 관리
sub_data = [id_utter, start_utter, end_utter, time_utter, speaker_sex, speaker_age, speaker_resi, numDialect, numEojeol]
label_data = dialect_region

# Dataset에 처리 결과 추가
X_mfcc.append(mfcc_pad)
X_subdata.append(sub_data)
Y_label.append(label_data)
```

11

/02. 데이터 전처리

- 차원 변환



[SVM]

```
[7]  # 데이터 변환(3차원 -> 2차원)
     X_mfcc_ALL = np.mean(X_mfcc_ALL, axis=1)

[8]  print(X_mfcc_ALL.shape)

     (60000, 259)
```

[RF]

```
# 3차원 -> 2차원
X_mfcc_conv = []
for item in X_mfcc_ALL:
    series = pd.Series(np.hstack((np.mean(item, axis=1), np.std(item, axis=1))))
    X_mfcc_conv.append(series)

X_mfcc_conv = np.array(X_mfcc_conv)

print(X_mfcc_ALL.shape)
print(X_mfcc_conv.shape)

(60000, 39, 259)
(60000, 78)
```

[CNN]

```
# 채널 차원 추가
X_mfcc_ALL = np.expand_dims(X_mfcc_ALL, axis=1)

print(X_mfcc_ALL.shape)

(60000, 1, 39, 259)
```

12

/03 모델 설계

13

/03. 모델 설계

SVM

RF

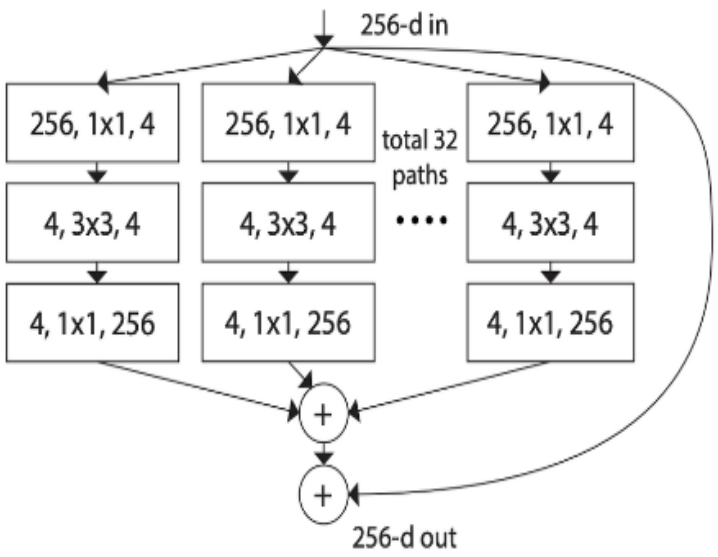
CNN (resNeXt50)

RNN

14

/03. 모델 설계

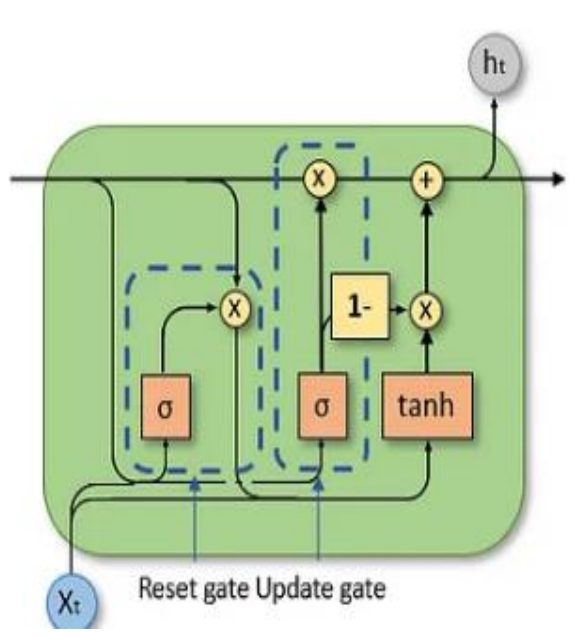
- CNN (resNeXt50)




```
ResNet(
  (conv1): Conv2d(1, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32, bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32, bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32, bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
```


/03. 모델 설계

- RNN (GRU)



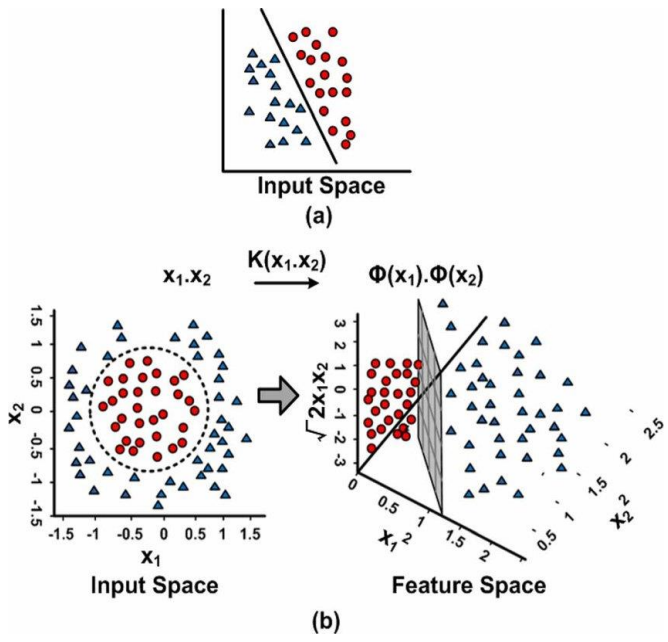
Model: "sequential_2"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 39, 100)	108300
batch_normalization (BatchNormalization)	(None, 39, 100)	400
leaky_re_lu (LeakyReLU)	(None, 39, 100)	0
gru_1 (GRU)	(None, 39, 200)	181200
batch_normalization_1 (BatchNormalization)	(None, 39, 200)	800
leaky_re_lu_1 (LeakyReLU)	(None, 39, 200)	0
gru_2 (GRU)	(None, 200)	241200
leaky_re_lu_2 (LeakyReLU)	(None, 200)	0
flatten (Flatten)	(None, 200)	0
dense (Dense)	(None, 100)	20100
dropout (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 20)	2020
dropout_1 (Dropout)	(None, 20)	0
dense_2 (Dense)	(None, 6)	126

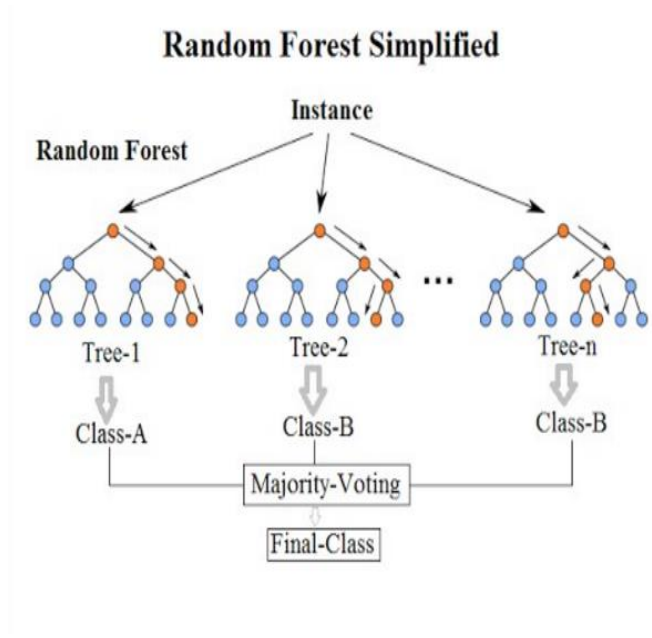
Total params: 554,146
Trainable params: 553,546
Non-trainable params: 600

/03. 모델 설계

- SVM



- RF



17

/04 결과 분석

18

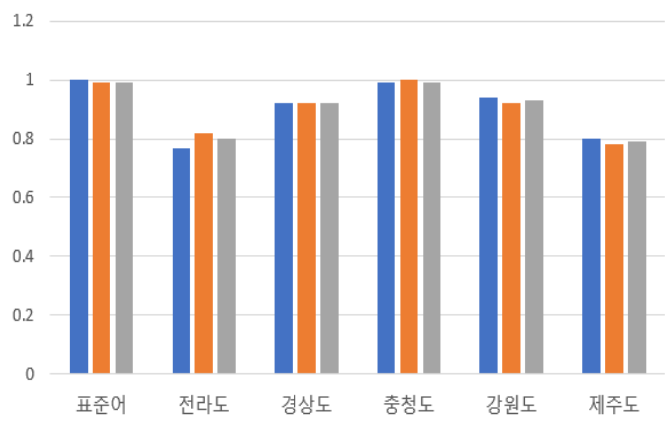
/04. 결과 분석

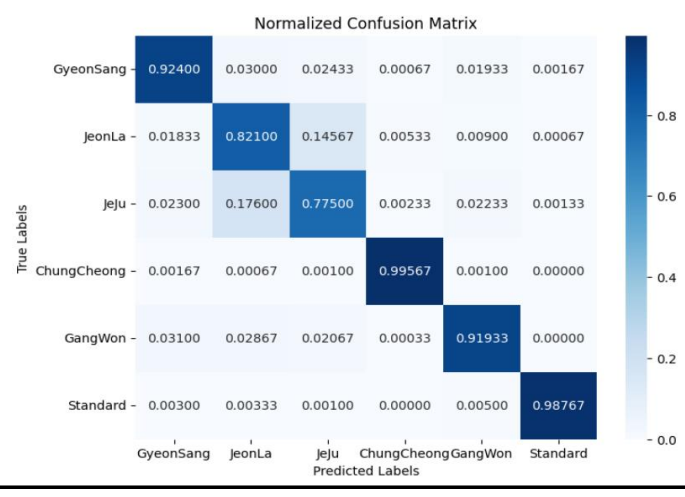
- **모델 별 성능지표**

	Accuracy	Precision	Recall	F1-Score
SVM(RBF)	0.49	0.59	0.60	0.59
RF	0.36	0.91	0.36	0.48
ResNeXT	0.90	0.90	0.90	0.90
GRU	0.71	0.72	0.71	0.72

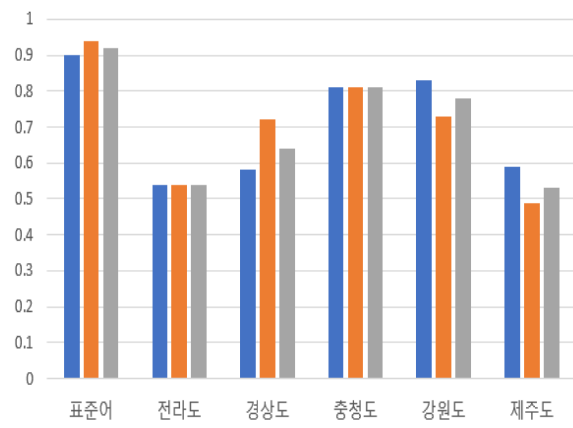
19

CNN





RNN



20

/05 결론

21

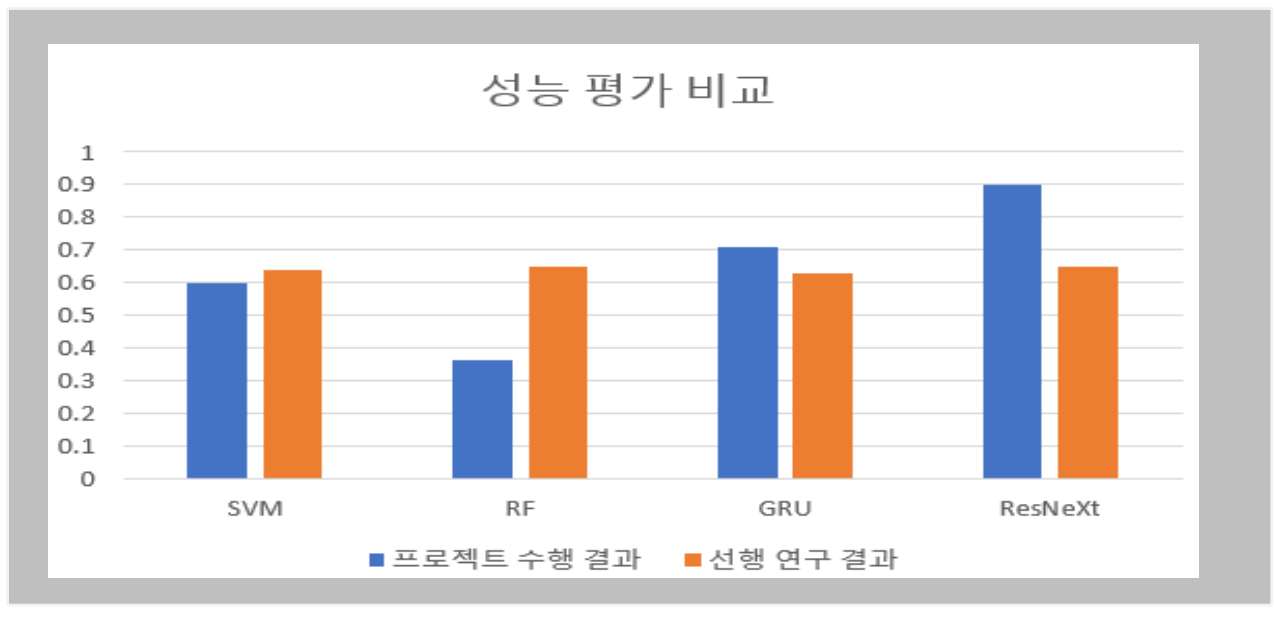
/05. 결론

	Accuracy	Precision	Recall	F1-score
SVM	64.02	62.69	64.02	61.73
RF	64.50	63.97	64.50	62.02
DNN	**65.00**	**64.49**	**65.00**	**63.43**
GRU	62.62	56.99	62.62	59.28
1D-CNN	64.23	58.80	64.23	60.84

[선행 연구 결과]

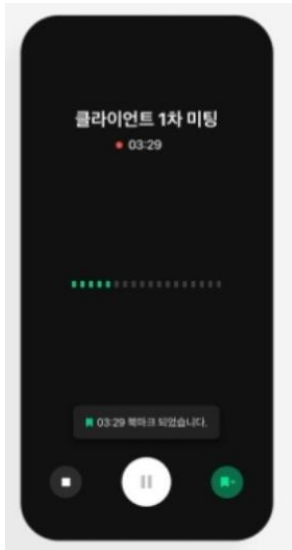
	Accuracy	Precision	Recall	F1-Score
SVM(RBF)	0.49	0.59	0.60	0.59
RF	0.36	0.91	0.36	0.48
ResNeXT	0.90	0.90	0.90	0.90
GRU	0.71	0.72	0.71	0.72

[프로젝트 수행 결과]



22

/05. 결론



충청도 : 어서 오세유

제주도
혼자 옵서예

전라도
언능 오랑께

경상도
퍼뜩 오이소

/06 참고문헌

/06. 참고문헌

[1] https://scholarworks.bwise.kr/ssu/handle/2018.sw.ssu/42106

[2] https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE02105288&googleIPSandBox=false&mark=0&ipRange=false&accessgl=Y&language=ko_KR&hasTopBanner=true

[3] https://www.earticle.net/Article/A408508

[4] http://ki-it.com/_PR/view/?aidx=17301&bidx=1359

[5] https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE07218067&googleIPSandBox=false&mark=0&ipRange=false&accessgl=Y&language=ko_KR&hasTopBanner=true

[6] https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE00215894&googleIPSandBox=false&mark=0&ipRange=false&accessgl=Y&language=ko_KR&hasTopBanner=true

[7] https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NODE01086202&googleIPSandBox=false&mark=0&ipRange=false&accessgl=Y&language=ko_KR&hasTopBanner=true

25

감사합니다

26