



입 모양 인식을 통한 구어 텍스트화 인터페이스

휴먼지능정보공학과 201710784 신은화
휴먼지능정보공학과 201710876 박희지
휴먼지능정보공학과 201710805 최영윤
휴먼지능정보공학과 201710809 최희수
휴먼지능정보공학과 201710812 한우정





1. 프로젝트 소개

1) 주제 선정 배경

2. 영어 Data를 이용한 입 모양 인식

1) 사용한 DataSet

2) DataSet 전처리

3) 모델 설명

4) 실험 결과

3. 한국어 Data를 이용한 입 모양 인식

1) DataSet 수집 방법

2) DataSet 전처리

3) 모델 설명

4) 실험 결과

4. 결론



1. 프로젝트 소개



프로젝트 소개

주제 선정 배경



Launch of the World Report on Hearing

3 March 2021 11:00 – 12:00 CET | Virtual

BACKGROUND

Globally over 430 million people experience disabling hearing loss, a number that could grow to nearly 700 million by 2050. When unaddressed, hearing loss poses a significant challenge for all age groups, hindering language development, communication, cognition,

Related

[World Report on Hearing](#)

[World Hearing Day 2021](#)

2020년 주요 지표별 등록장애인 현황

(단위 : 천 명, %)

성별			연령별			장애유형별			장애정도별			시도별		
구분	인원	비율	구분	인원	비율	구분	인원	비율	구분	인원	비율	구분	인원	비율
합계			2,633(100)											
남성	1,521	57.8	0~9세	32	1.2	지체	1,207	45.8	심한 장애	985	37.4	서울	394	15.0
			10~19세	59	2.2	시각	252	9.6				부산	176	6.7
			20~29세	98	3.7	청각	396	15.0				대구	126	4.8
			30~39세	122	4.6	언어	22	0.8				인천	146	5.5
			40~49세	243	9.2	지적	217	8.2				광주	70	2.7
			50~59세	452	17.2	뇌병변	250	9.5				대전	73	2.8
여성	1,112	42.2	60~64세	314	11.9	자폐성	31	1.2	심하지 않은 장애	1,648	62.6	울산	51	1.9
			65~69세	289	11.0	정신	104	3.9				세종	12	0.5
			70~79세	585	22.2	신장	98	3.7				경기	570	21.6
			80세~	440	16.7	심장	5	0.2				강원	102	3.9
						호흡기	12	0.5				충북	98	3.7
						간	14	0.5				충남	134	5.1
			안면	3	0.1	전북	132	5.0						
			장루요루	15	0.6	전남	141	5.4						
			뇌전증	7	0.3	경북	181	6.9						
								경남	189	7.2				
								제주	37	1.4				

장애대학생 온라인 줌 강의 '산 넘어 산'

수강신청 난관...편의 부족 시험·과제 '끔끔'

"학습권 보장" 비대면 교육운영 매뉴얼 필요

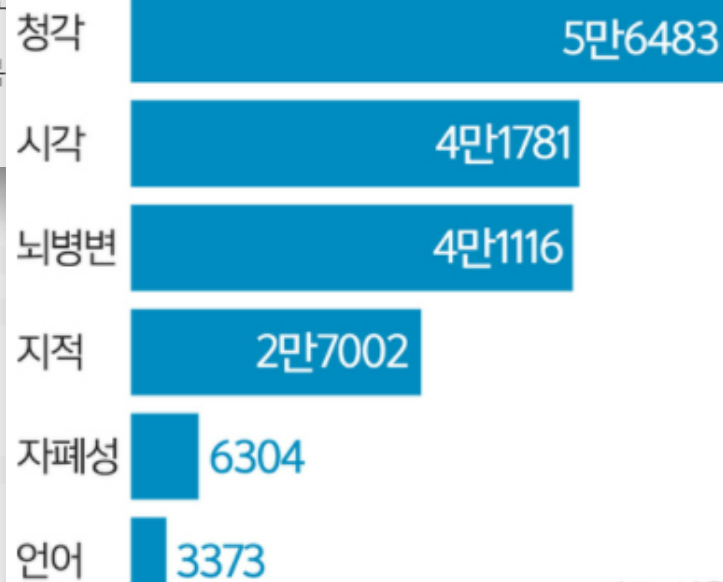
HOME > 통계뉴스 > 통계뉴스

2019년 신규 등록장애인 현황

김세진 기자 |

의사소통에 어려움을 겪는 유형의 장애인 (단위: 명)

*2020년 7월 서울시 등록 기준



자료: 서울시



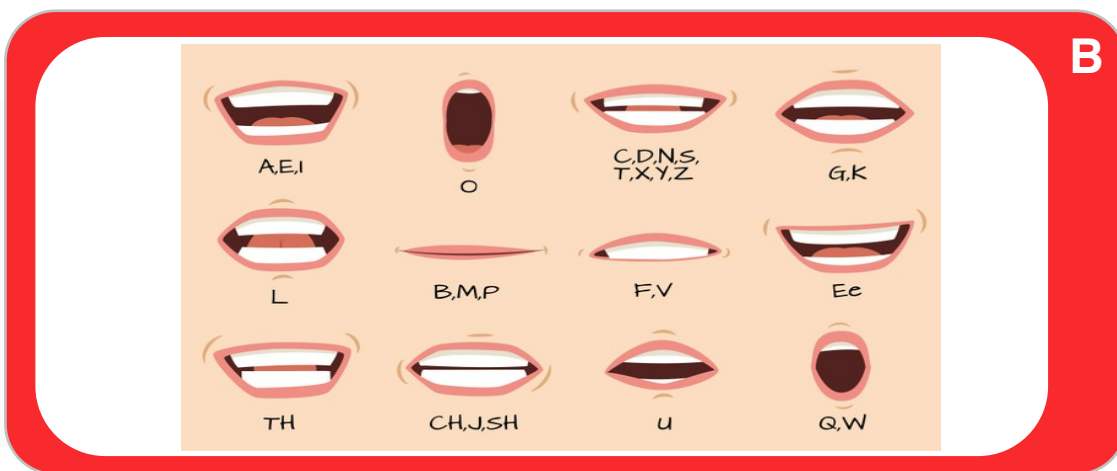
A. 음성인식 기반

소음문제에 취약

부정확한 발음은 제대로 인식되기 어려움



문제점 보완 가능



B. 입술 모양 인식 기반

소음문제 없음

부정확한 발음에도 강함



2. 영어 *Data*를 이용한 입 모양 인식



1) 사용한 DataSet

Achraf Ben-Hamadou

 Search this site

Navigation

[::Homepage](#)

[::Publications](#)

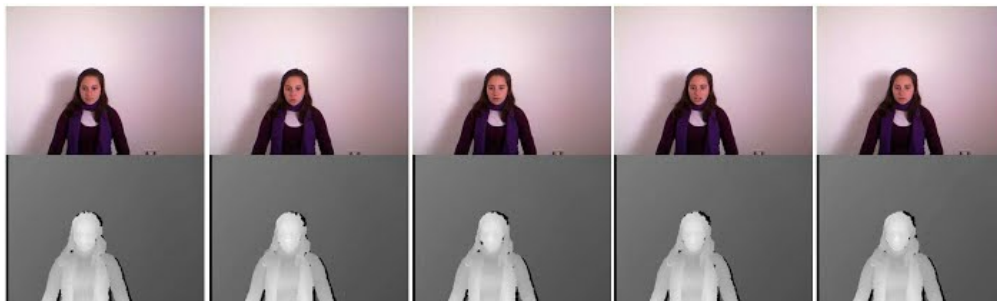
▼ [::Datasets](#)

MIRACL-VC1

[::Calendar](#)

[::Datasets](#) >

MIRACL-VC1



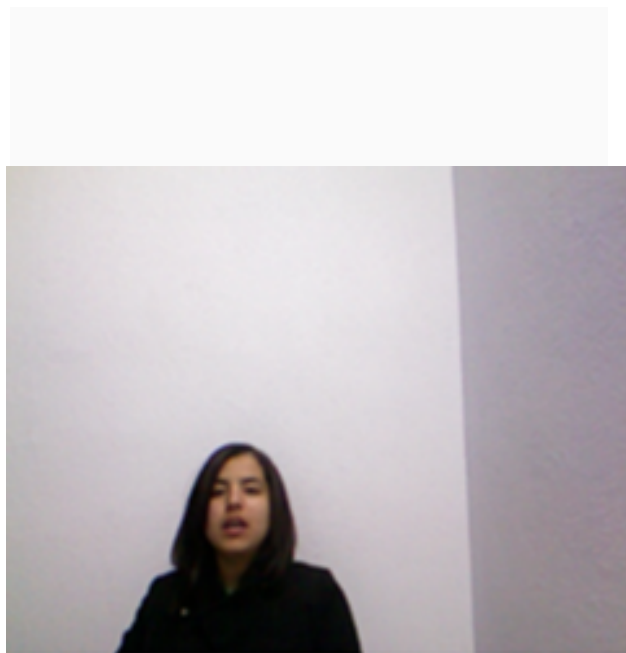
MIRACL-VC1 is a lip-reading dataset including both depth and color images. It can be used for diverse research fields like visual speech recognition, face detection, and biometrics. Fifteen speakers (five men and ten women) positioned in the frustum of an MS Kinect sensor and utter ten times a set of ten words and ten phrases (see the table below). Each instance of the dataset consists of a synchronized sequence of color and depth images (both of 640x480 pixels). The MIRACL-VC1 dataset contains a total number of 3000 instances.

ID	Words	ID	Phrases
1	<i>Begin</i>	1	<i>Stop navigation.</i>
2	<i>Choose</i>	2	<i>Excuse me.</i>
3	<i>Connection</i>	3	<i>I am sorry.</i>
4	<i>Navigation</i>	4	<i>Thank you.</i>
5	<i>Next</i>	5	<i>Good bye.</i>
6	<i>Previous</i>	6	<i>I love this game.</i>
7	<i>Start</i>	7	<i>Nice to meet you.</i>
8	<i>Stop</i>	8	<i>You are welcome.</i>
9	<i>Hello</i>	9	<i>How are you?</i>
10	<i>Web</i>	10	<i>Have a good time.</i>

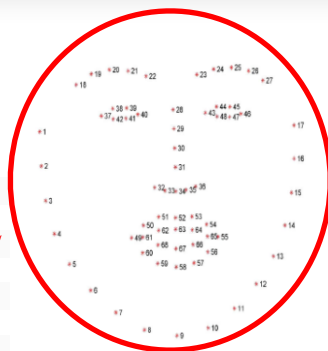
10개의 단어와 10개의 문장



2) DataSet 전처리



원본



입술DataSet

+augmentation



얼굴DataSet

+augmentation

얼굴영역 검출

입술영역 검출



3) 모델 설명

Lip Reading Word Classification

Abiel Gutierrez
Stanford University
abielg@stanford.edu

Zoe-Alanah Robert
Stanford University
zrobert7@stanford.edu

Abstract

We present a variety of models and methods for predicting words from video data without audio. Previous work exists in this subject area, but it is limited and very recent. In this paper we use the MIRACL-V1 dataset [0] containing videos of ten people speaking ten words. We pre-process the data by using existing facial recognition software to detect and crop around the subject's face in all frames of the video and then use the sequence of frames as input to the model. We explore a CNN + LSTM Baseline model, a Deep Layered CNN + LSTM model, an ImageNet Pretrained VGG-16 Features + LSTM model, and a Fine-Tuned VGG-16 + LSTM model. This paper discusses the effects of dropout, hyperparameter tuning, data augmentation, seen vs unseen validation splits, batch normalization, and other techniques for adjusting these models. We achieve a validation accuracy of 79% and a test accuracy of 59% on our best model.

1. Introduction

This paper investigates the task of speech recognition from video without audio. We present several neural network models with varying successes in this classification task. The input data to our algorithm is sequences of still images taken from frames of video footage. We use different neural network models to output one of 10 words that are spoken (or mouthed) by a face in the input images. We explore and combine a number of different models including CNNs, RNNs, and existing publicly available pre-trained networks to assist in mouth recognition.

This is an interesting learning task given that video traffic is growing at a high rate throughout the web, and this model could help extract data and process it to gain insights into the action or topics occurring in a video. Applications of a visual audio classifier range from play prediction in sporting events to profanity detection on social media sites to a live action lip reading mobile application.

Recently, Wand et al. [12] introduce reading at the word level, which we discuss

In the past, research efforts have been on gesture recognition rather than recognition, making this for a new and explore. There are a few existing system for lip reading, although most do not use but instead other machine learning / advanced visual speech recognition Google's DeepMind LipNet [1] network only a few months ago.

2. Related Work

In this section, we outline the existing work done in the field. As previously approaches have involved machine learning do not touch on deep learning. It has on recently that deep learning methods have produced state-of-the-art results.

Pei et al [2] used Random Forest Mani this same task, extracting patic spatiotemporally and then mapping these patterns. Rekik et al [3] used Hidden Markov solve this problem with color and depth images. They extracted a 3D rendition mouth, and generated a variety of features obtained a 62.1% classification accuracy the MIRACL-V1, performing speaker in. One of the first works to use deep lip recognition was Hinton et al.[4], where were used for acoustic processing. (include learning multimodal audio-visu [5, 6] and learning visual features to the traditional classifier structures like RNN have gone beyond word-level: Noda et al to predict phonemes, and Shaikh et al. [8] to predict visemes. Koller et al. [10] also using an image classifier CNN. More go by Graves et al. [11] has been consider development end-to-end deep speech recognition to their development of the convolutional loss (CTC), which allows CNNs.

Preprocessing was an important part of working with this dataset. First, we utilized a python facial recognition library, dlib, in conjunction with OpenCV and a pre-trained model [2] to isolate the points of facial structure in each image and crop it to only include the face of the speaker, excluding any background that could interfere with the training of the model. We had to limit the size of every facial crop to a 90x90 pixel square in order to create uniform input data sequences for the model.

paper. Chung & Zisserman [13] made use of the work of Graves et al. by using spatiotemporal CNNs for word classification on the BBC TV dataset. Assael et al. [1] created LipNet, a phrase predictor that uses spatiotemporal convolutions and bidirectional GRUs and achieved a 11.4% WER on unseen speakers. Our model is primarily inspired by this work. We also took inspiration from Garg et al. [14], where a pre-trained VGG was used for transfer learning on the MIRACL-V1 dataset. A much more comprehensive list of lip reading works can be found in Zhou et al. [15].

3. Dataset and Features

We used the MIRACL-VC1 data set [0] containing both depth and color images of fifteen speakers uttering ten words and ten phrases, ten times each. The sequence of images represents low quality video frames. The data set contains 3000 sequences of varying lengths of images of 640 x 480 pixels, in both color and depth representations, collected at 15 frames per second. The lengths of these sequences range from 4 to 27 image frames. The words and phrases are as follows:

Words: begin, choose, connection, navigation, next, previous, start, stop, hello, web
Phrases: Stop navigation, Excuse me, I am sorry, Thank you, Good bye, I love this game, Nice to meet you, You are welcome, How are you, Have a good time

For the sake of time and utilizing smaller data sizes, we focused on building a classifier that can identify which word is being uttered from a sequence of images of the speaker as input. We ignored the set of phrase data and also the depth images for the spoken word data. We built classifiers for both seen and unseen people. (Seen meaning that the model is trained on all people saying all words but saves certain trials for test and validation. Unseen removes people from training and adds them to exclusively to either testing or validation. The split is thirteen people for train, one for validation, and one for test.) The resulting datasets are (1200/150/150) (train/test/validation) examples for seen and (1300/100/100) (train/test/validation) examples for unseen. The class label distribution for the dataset is even as each person performs the same number of trials per word.

Preprocessing was an important part of working with this dataset. First, we utilized a python facial recognition library, dlib, in conjunction with OpenCV and a pre-trained model [2] to isolate the points of facial structure in each image and crop it to only include the face of the speaker, excluding any background that could interfere with the training of the model. We had to limit the size of every facial crop to a 90x90 pixel square in order to create uniform input data sequences for the model.

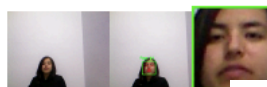


Figure 1: (left to right) Original Input image (pu in the MIRACL-VC dataset), OpenCV and dlib software labelling key points on around a det cropped image

One issue with this data set is its small size: the number of training sequences, we p augmentation. We tripled the data set in size horizontally flipped version of each image a pixel-jittered version of each image.

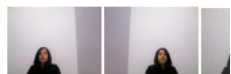


Figure 2: (left to right) Original Input image (pu in the MIRACL-VC dataset), a horizontally flipped jittered image.

In summary, each model receives a sequence as input – with anywhere from 4 to 10 the sequence – and produces a single word label as output.

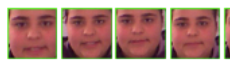


Figure 3: Example full input sequence of length 10 speaking "begin."



Figure 4: Example full input sequence of length 10 speaking "hello."

4. Methods

In this section we describe the different models created to solve the lip reading problem. We

models: a Baseline CNN + LSTM network; a more robust and deep layered CNN + LSTM network inspired by Deep Mind's LipNet[1]; an LSTM network placed on top of bottleneck features developed by a VGG16 network pre-trained on ImageNet; and the same LSTM network on top of VGG16 with fine-tuning of the last convolutional block.

4.1 CNN + LSTM Baseline

Our first model ran every image of our sequenced input through a Convolutional Neural Network and then fed the flattened outputs as a sequence into a Long Short Term Memory Recurrent Neural Network, which produced a single output, making it a many-to-one RNN. We then added a Fully Connected layer that mapped to 10 units, and used a softmax activation layer to produce the probabilities of every word, of which we took the highest:

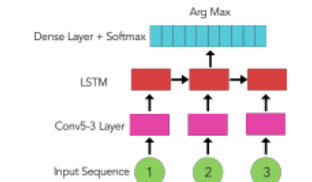


Figure 5: CNN + LSTM Baseline model layer architecture diagram

Our convolutional layer had a kernel size of 5x5 and depth of 3 filters – inspired from LipNet's architecture [1] – and was added to help the model make sense of the high-level features of the images. It achieves this by running the kernel across the image, mapping the dot products of the pixel overlaps to a new layer, and stacking together the layers produced by every filter:

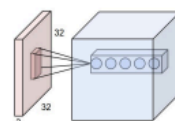


Figure 6: Structural diagram courtesy of CS231N at Stanford - <http://cs231n.github.io/convolutional-networks/>

The LSTM was added to package the entire sequence of CNN outputs into a single layer without losing the temporal understanding of the video frames. In particular, an LSTM fixes the vanishing gradient problem present in vanilla RNNs, which inhibits the backpropagation of gradients to occur [16]. It does so by adding 4 gates (input (i), forget (f), output (o), new memory (c)) whose activations can be learned, in order to control whether or not to hold on to information:

$$\begin{aligned} i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) && \text{(Input gate)} \\ f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) && \text{(Forget gate)} \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) && \text{(Output/Exposure gate)} \\ c_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) && \text{(New memory cell)} \\ h_t &= f_t \odot c_{t-1} + i_t \odot c_t && \text{(Final memory cell)} \end{aligned}$$

Given that we use softmax as our last activation, our loss function is cross entropy loss:

$$\text{loss} = - \sum_i \log \left(\frac{\exp(W_{x_i})}{\sum_j \exp(W_{x_j})} \right)$$

Finally, we used the Adam Optimizer to better navigate through the loss function.

4.2 Deep Layered CNN + LSTM

We expanded on our baseline by first adding 2 more layers of CNNs, in order to develop an understanding of more intricate features in our input images. We made our LSTM bidirectional, to avoid overweighing the output with frames in the latter parts of the sequence, and added dropout and batch normalization after every CNN layer. We kept our dropout probability at 0.2 given that we performed it several times across the model. We also interspersed 2x2 Max Pooling layers with strides of 2 between the CNNs. This model is even more similar to LipNet's [1]:

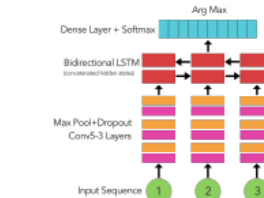


Figure 7: Deep Layered CNN + LSTM model layer architecture diagram

참고논문

A. Gutierrez and Z.-A. Robert, Lip Reading Word Classification, ed, 2017.

얼굴 데이터셋
CNN, VGG 모델



3) 모델 설명

Data. Lip

model	train_acc	train_loss	val_acc	val_loss
CNN + LSTM	1.0	0.003	0.66	1.53
✓ VGG-16 + LSTM	0.93	0.19	0.68	1.29
Xception + LSTM	0.97	0.08	0.54	1.76
✓ MobileNet + LSTM	0.97	0.09	0.73	1.09
EfficientNet + LSTM	0.91	0.25	0.66	1.09

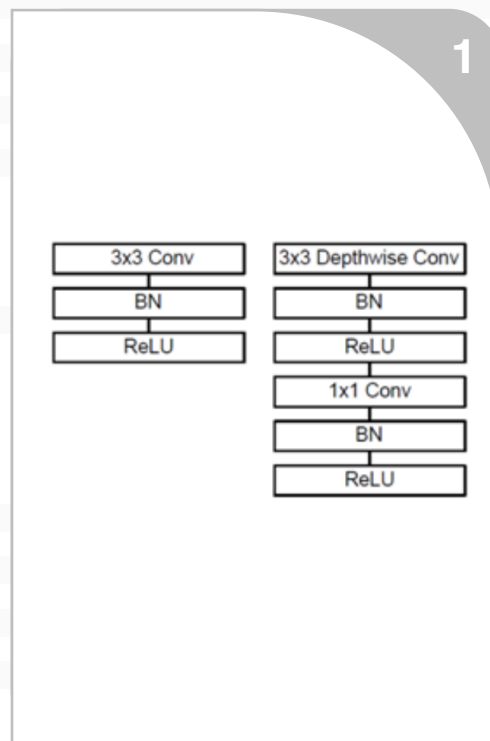
Data. Face

model	train_acc	train_loss	val_acc	val_loss
CNN + LSTM	0.99	0.04	0.49	2.28
VGG-16 + LSTM	0.98	0.08	0.54	1.59
Xception + LSTM	0.96	0.12	0.51	1.99
MobileNet + LSTM	0.73	0.87	0.30	2.28
EfficientNet + LSTM	0.97	0.09	0.57	1.54

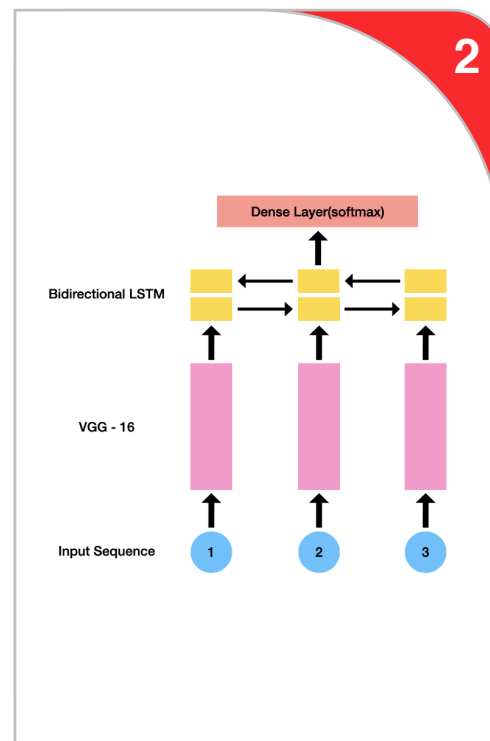
입술 · 얼굴 data에 대한 5개의 모델 성능 실험

전체적으로 Lip data에서 더 좋은 성능

MobileNet+LSTM & VGG-16+biLSTM 좋은 성능



MobileNet + LSTM

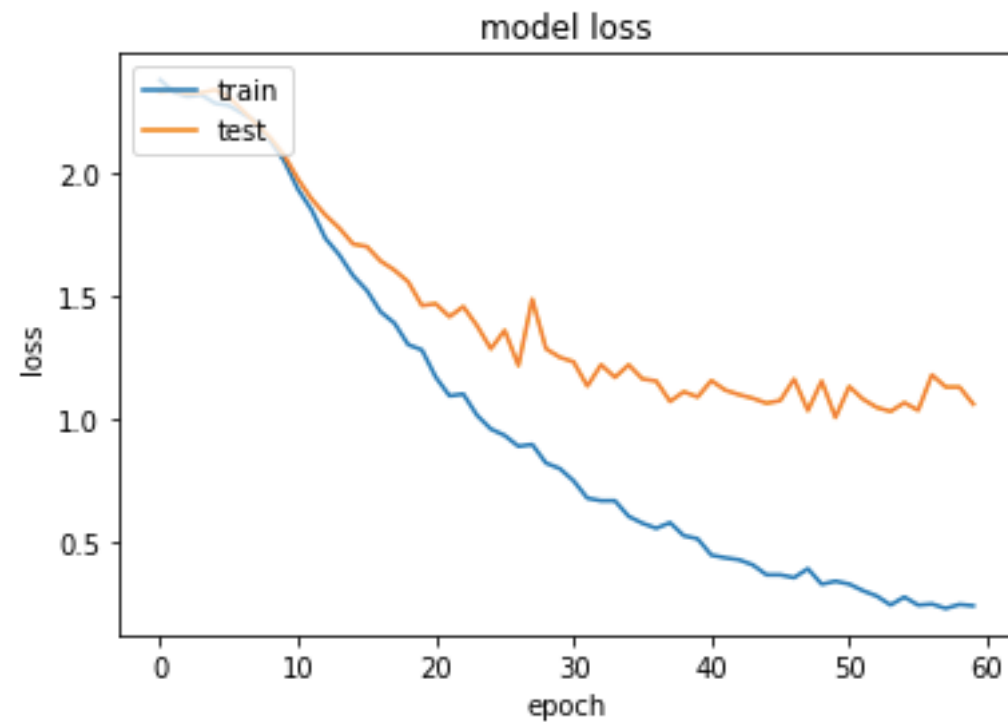
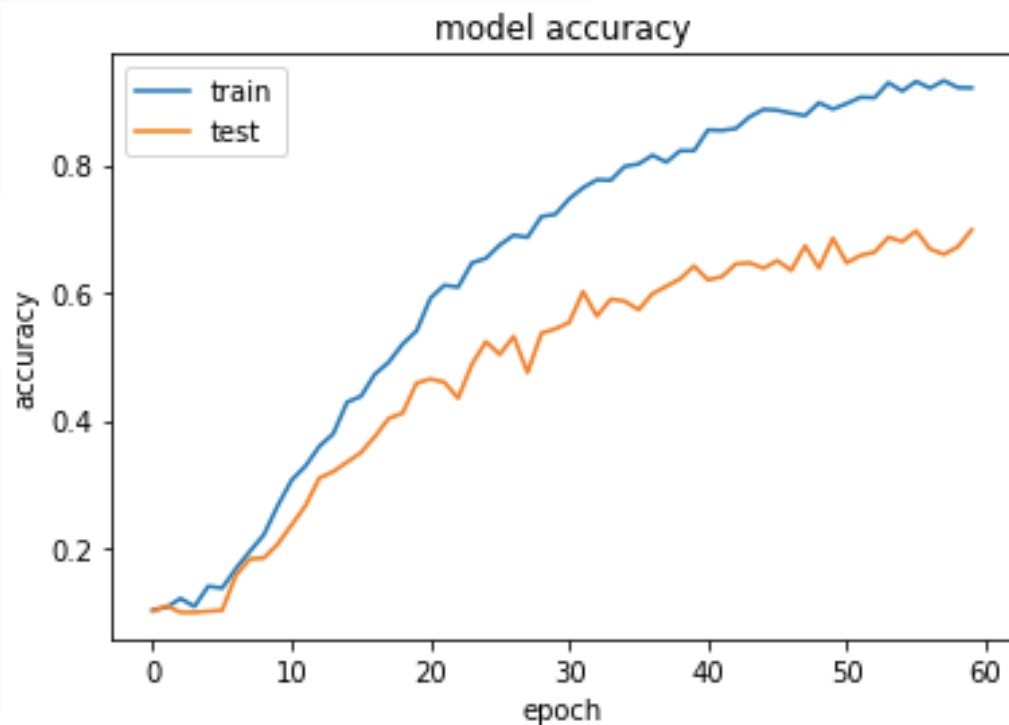


VGG-16 + LSTM



4) 실험 결과

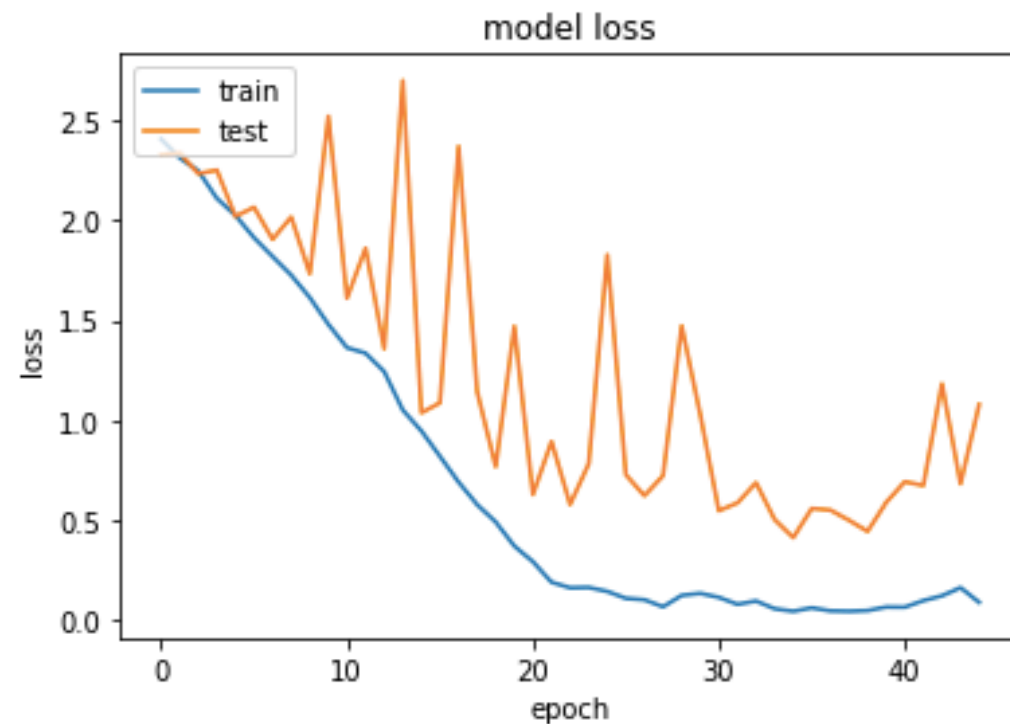
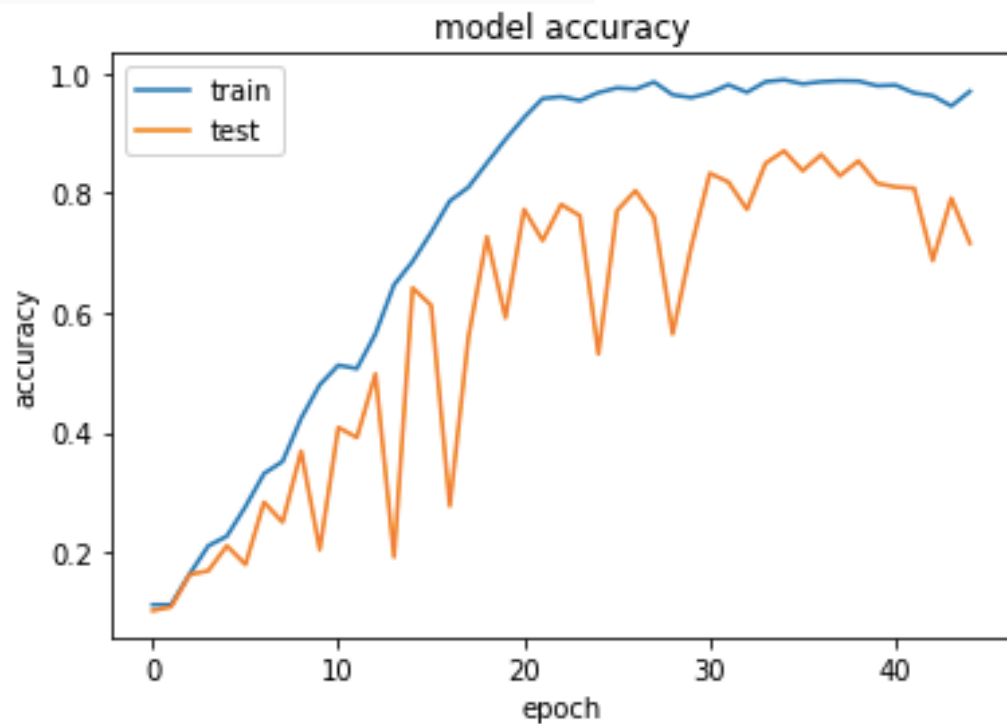
VGG-16 + LSTM 모델





4) 실험 결과

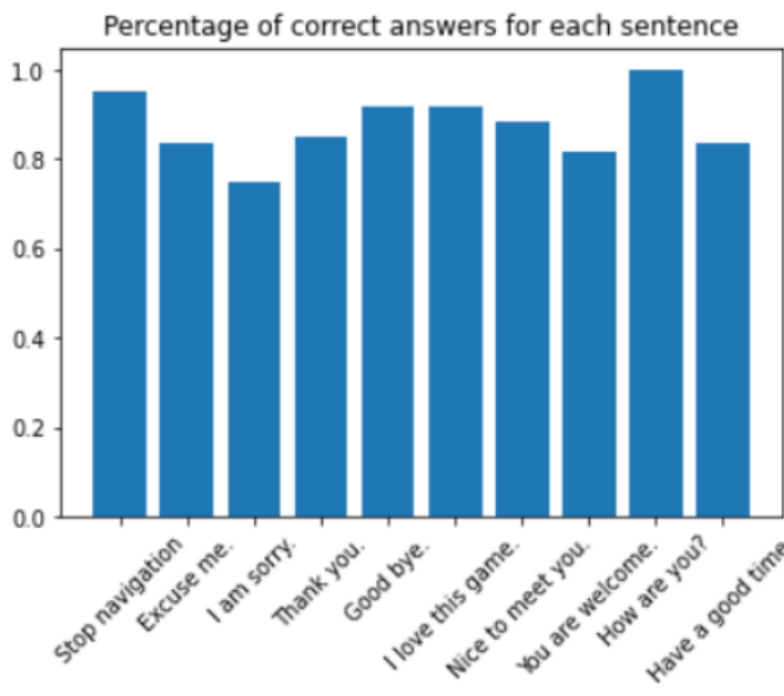
MobileNet + LSTM 모델



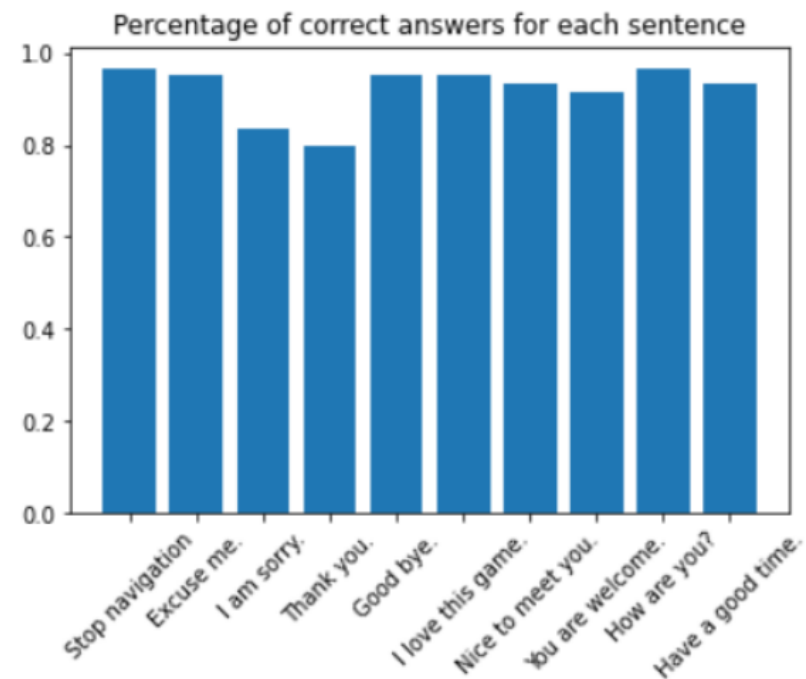


4) 실험 결과

두 모델 예측을 통한 문장 별 정답 비율



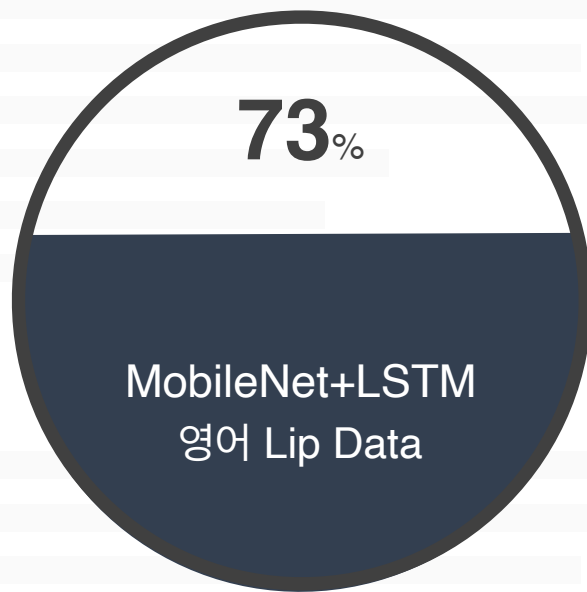
VGG-16 + LSTM



MobileNet + LSTM



4) 실험 결과



모델 test 결과

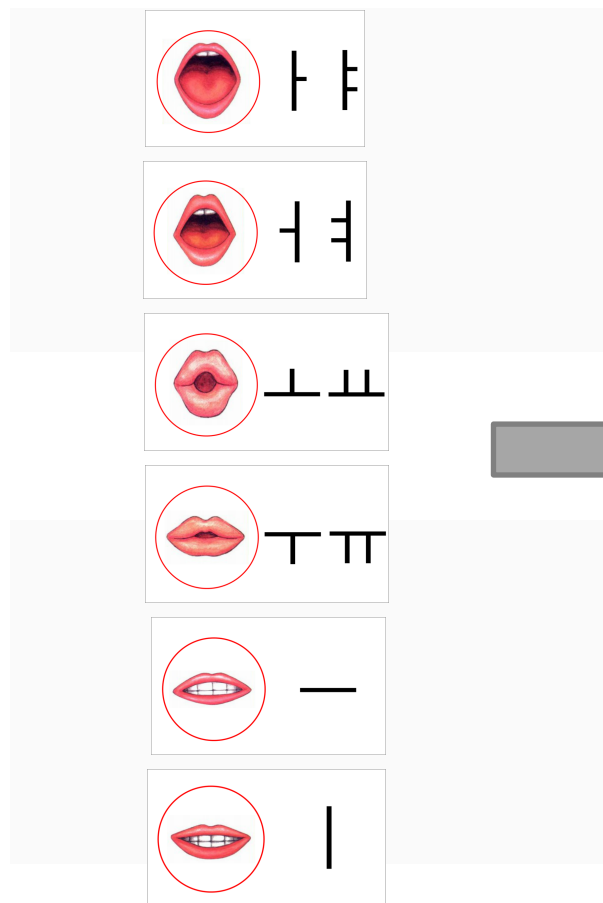
이전 논문 대비 20% 가량 높은 성능



3. 한국어 *Data*를 이용한 입 모양 인식



1) DataSet 수집 방법



한글 모음 별 입모양

문장

- 도와주세요
- 힘내세요
- 누구세요
- 안녕하세요
- 조심히가세요
- 죄송합니다
- 감사합니다
- 좋아요
- 싫어요

입 모양이 좀 더 두드러지게 나타나는
모음 위주의 조합으로 문장을 구성

음절 코드

가	00	요	14
다	01	구	15
사	02	누	16
아	03	주	17
하	04	니	18
감	05	히	19
안	06	싫	20
합	07	심	21
어	08	힘	22
녕	09	계	23
도	10	내	24
조	11	세	25
승	12	와	26
중	13	죄	27

문장의 길이와 말의 속도 차이로 발생하는
Frame 관련 문제점을 해결하기 위해 한 음절 씩 수집



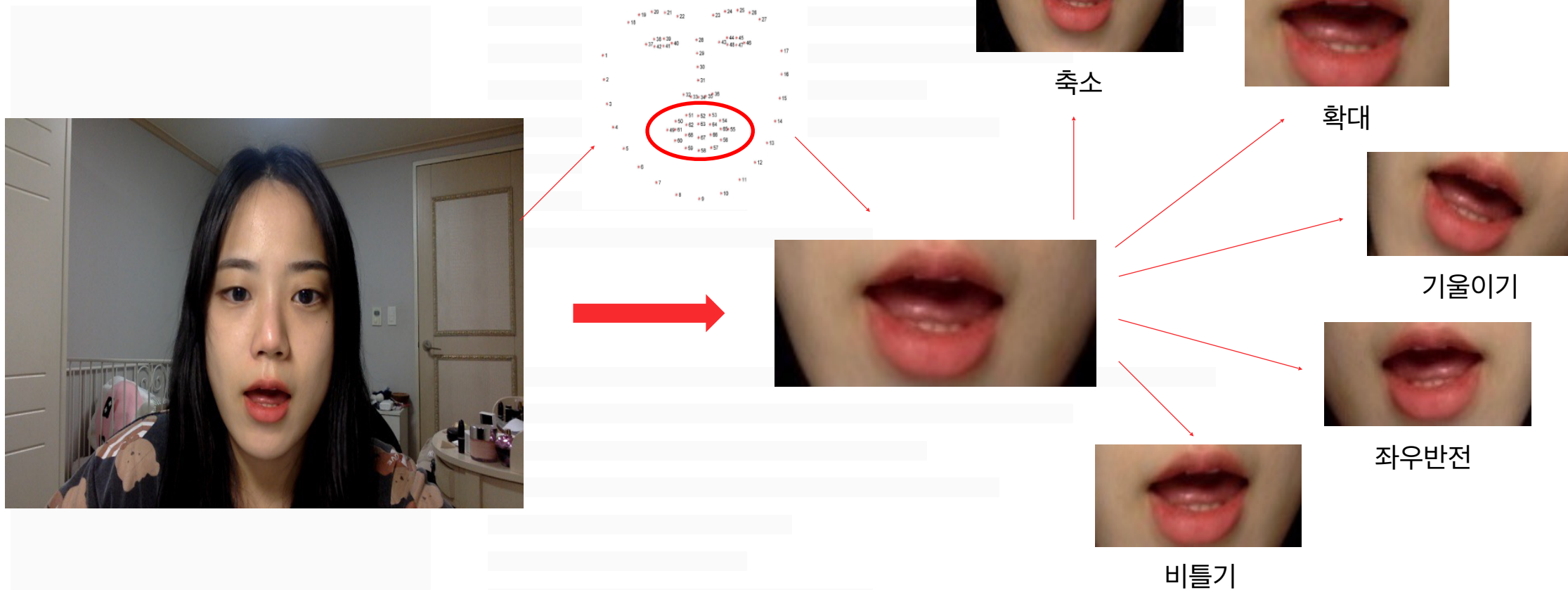
1) DataSet 수집 방법

앞말	끝말
도와주	세요
힘내	
누구	
안녕하	
안녕히계	
조심히가	
죄송	합니다
감사	
좋아	요
싫어	

- 문장의 끝 말은 반복
Ex) ~세요, ~합니다, ~요 등등
→ 앞 말과 끝 말의 분리
- 이후 머신러닝을 통해 앞 말과 끝 말을 합친 후,
추천 해주는 역할을 하는 모델 구현
→ 단순 classification 보다 성능↑



2) DataSet 전처리

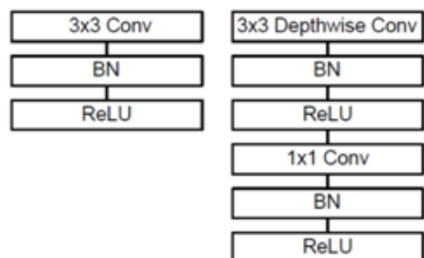


입술 영역 검출 후 200X100 사이즈로 cropping

Augmentation으로 데이터 다양화, 증량



3) 모델 설명



MobileNet + LSTM

영어 Dataset 훈련 과정에서 가장 높은 성능을 보인 MobileNet 사용

음절 코드

가	00	요	14
다	01	구	15
사	02	누	16
아	03	주	17
하	04	니	18
감	05	히	19
안	06	싫	20
합	07	심	21
어	08	힘	22
녕	09	계	23
도	10	내	24
조	11	세	25
송	12	와	26
좋	13	죄	27

수집한 28가지 음절 데이터

모음 단어 코드

ㅏ	하 가 사 아 다	00
ㅑ	안	01
ㅓ,ㅕ	감, 합	02
ㅖ	녕 어	03
ㅗ,ㅛ	도 조 송 좋 요	04
ㅜ,ㅠ	주 누 구	05
ㅣ	히 싫 니	06
ㅡ	힘 심	07
ㅗ ㅖ ㅓ	내 계 세	08
ㅓ	와	09
ㅖ	죄	10

유사한 모음과 받침을 묶어 11가지로 분류



3) 모델 설명

```
1 def clustering(str, list1, list2):  
2     score = []  
3     for i in range(len(str_list)):  
4         ratio = SequenceMatcher(None, str, list1[i]).ratio()  
5         score.append(ratio)  
6     index = score.index(max(score))  
7     return list2[index]
```

○ difflib의 SequenceMatcher 함수 사용

```
1 str = '오아'  
2 clustering(str, str_list, answer_list)
```

'좋아'

```
1 str = '오아오'  
2 clustering(str, str_list, answer_list)
```

'좋아요'

```
1 str = '오아암이'  
2 clustering(str, str_list, answer_list)
```

'좋아합니다'

```
1 str = '오아오오'  
2 clustering(str, str_list, answer_list)
```

'좋아요'

```
1 str = '암아'  
2 clustering(str, str_list, answer_list)
```

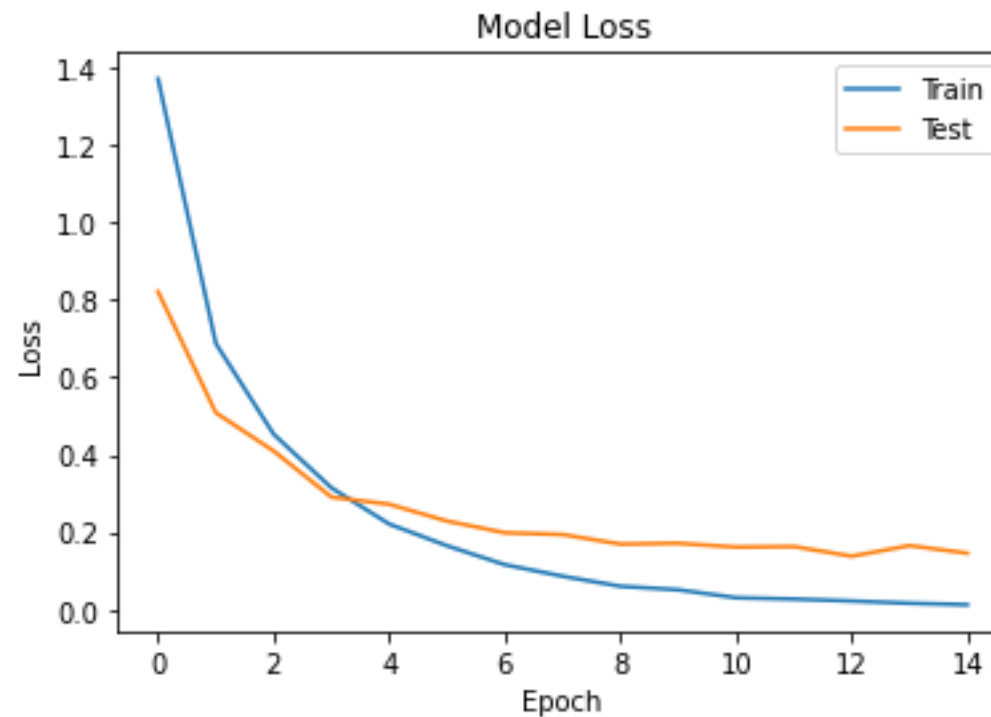
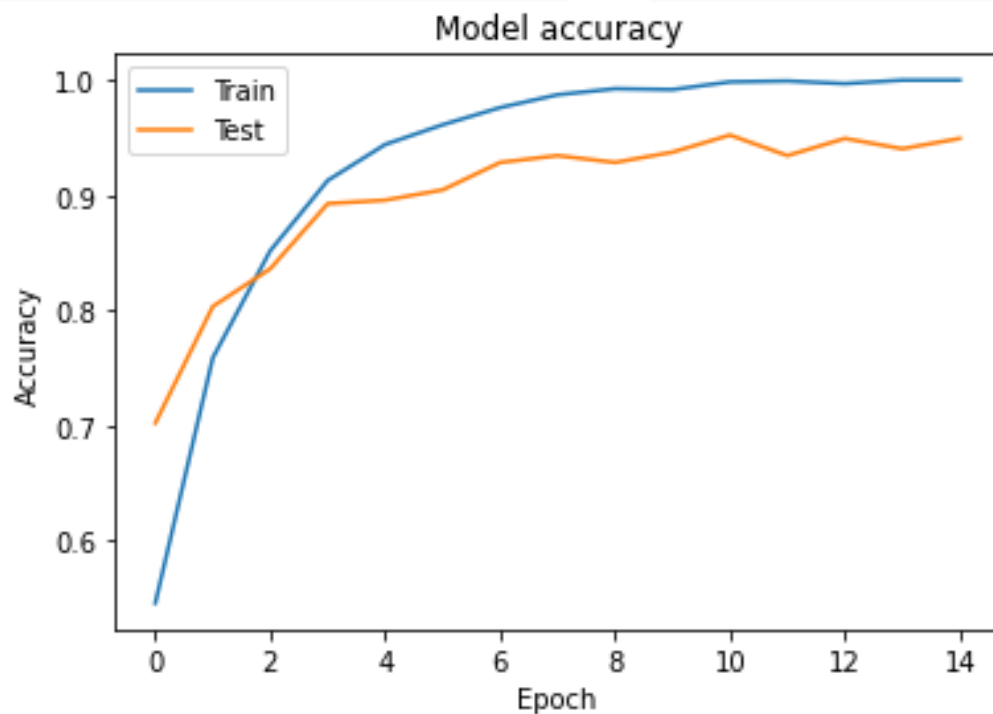
'감사'

모음이 완전히 일치하지 않아도
가장 근접한 문장(혹은 단어)를 찾아내어 반환
앞말과 끝말의 연결을 통해 좋아, 좋아요, 좋아합니다
등 다양한 표현 가능



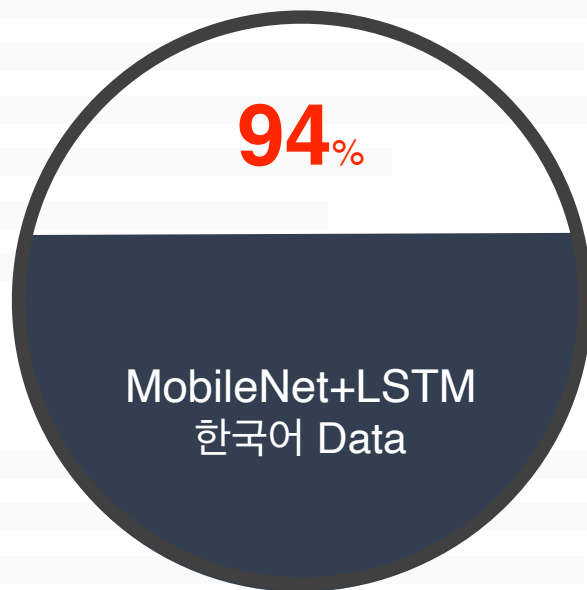
4) 실험 결과

MobileNet + LSTM 모델 성능





4) 실험 결과



모델 test 결과

기존에 존재하던 영어 Dataset을 이용한 연구보다
직접 수집한 한국어 Dataset을 이용한 연구에서 더 좋은 성능



4. 결론



활용 방안 및 기대효과

Barrier free

비장애인과 장애인간의 원활한 의사소통

수화 통역, 사회복지 분야 활용

동영상 플랫폼에서의 자막 기능

발전 방향성

- 음성 인식 기반 시스템과 결합해

multimodal model로 제작

- KoNLPy를 사용한 형태소 분석, 문장 추천 model제작