실시간 수화 번역 인식 모듈생성

MediaPipe, LSTM

20202658 이태범
20182806 서동혁
20182832 최준용

팀원 및 역할 소개

이태범

Team Elder

Specialize in Modeling
Data Generation, Handling
Data Preprocessing

서동혁

Team Younger

Specialize in Presentation
Data Generation, Handling
Data Preprocessing

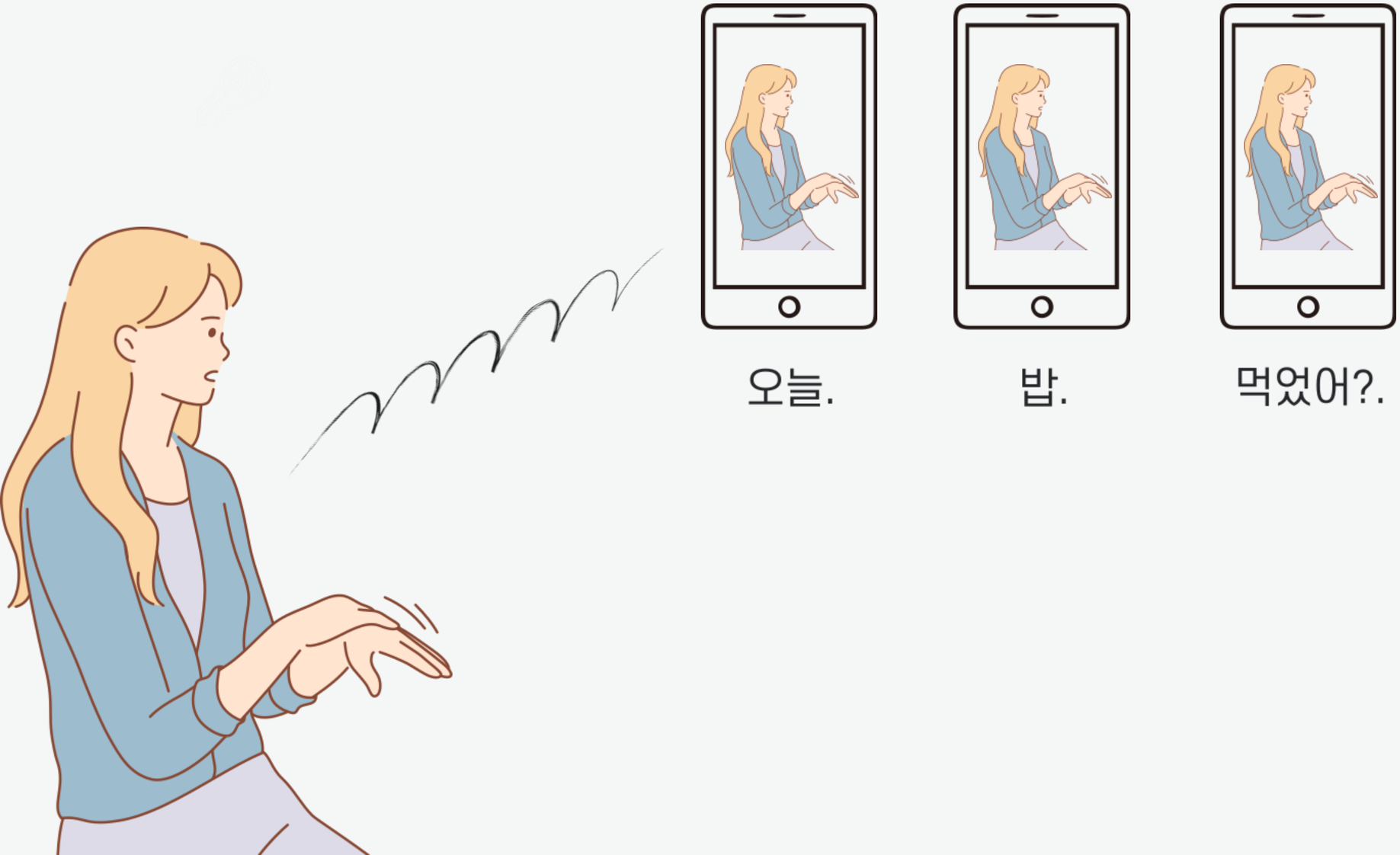
최준용

Team Member

Specialize in PPT
Data Generation, Handling
Data Preprocessing

Contents

중간발표 review



중간발표 review

- AI Hub의 수어 영상 데이터 사용

- 총 536,000 수어영상 클립(.mp4 파일)
- 수어문장 2000개, 수어단어 3000개, 지숫자/지문자 1000개에 대한 영상
- 스튜디오 직접 촬영으로 언어제공자 20명에게서 5각도 동시 촬영한 수어문장/단어 영상(500,000 수어영상 클립)
- 크라우드소싱 촬영으로 언어제공자 21명에게서 수집한 지수어 영상(21,000클립)
- 아바타로 제작한 수어문장/단어 영상(15,000 클립)

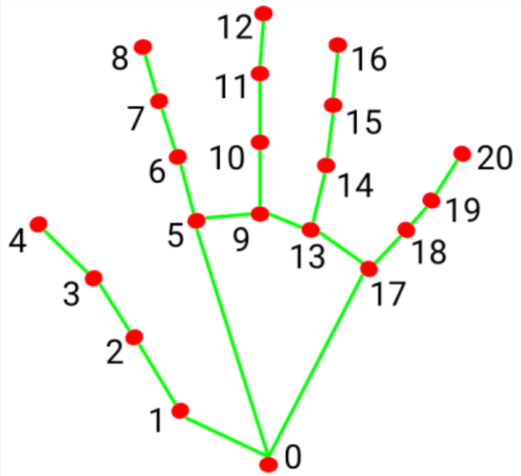
Keypoint 가공	Pose	25개 키포인트 각각의 x,y,confidence값
	left hand	21개 키포인트 각각의 x,y,confidence값
	right hand	21개 키포인트 각각의 x,y,confidence값
	face	68개 키포인트 각각의 x,y,confidence값

AI Hub

중간발표 review

- 컴퓨터 비전 라이브러리인 OpenCV와 구글에서 제공하는 AI 프레임워크인 MediaPipe를 이용하여 입력 영상을 모델에 넣기 위한 npy 파일로 변환

- MediaPipe의 Holistic Model을 이용하여 왼손, 오른손에 해당하는 Key Point를 추출



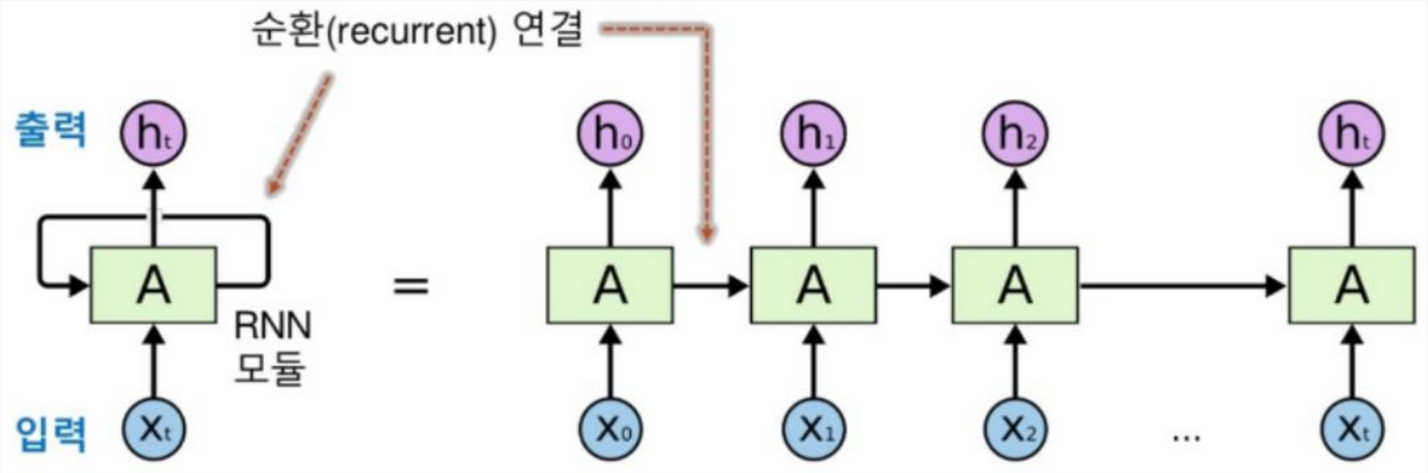
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

MediaPipe

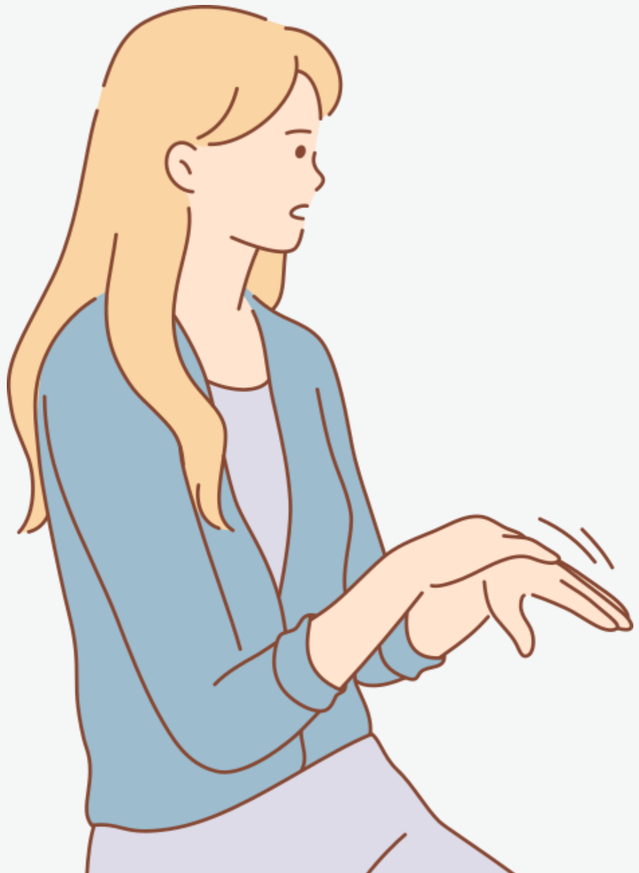
OpenCV

중간발표 review

- 시계열 데이터 처리에 주로 활용되는 *RNN 계열의 모델*을 활용 (RNN, LSTM, GRU)
- RNN 계열의 모델은 이전 상태에 대한 정보를 메모리 형태로 저장할 수 있기 때문에 시계열 데이터 분석에 적절
- 영상 데이터의 위치 정보에 대한 좌표 정규화, 벡터 정규화를 통한 모델의 성능 향상 기대
- 분류 문제에 적합한 평가지표(F1-Score 등) 활용 예정



프로젝트 소개 (주제 세부사항 변경 및 수정)



수어 단어 -> 지문자 활용

수어 단어의 동작들의 특성들이 너무 길고 복잡해서 판별하는데 어려움이 존재.
부족한 시간 -> 지문자를 분석해보자.

양손 joint -> 오른손 joint

양손으로 표현하는 단어. But 문자는 한손으로 표현가능
오른손의 joint들만 사용 하게끔 코드 생성.

사용 Data 변경

AI Hub Data -> 너무 복잡하고 data 크기 방대함.
직접 지문자 동영상 Data 생성.

실시간 영상 + 녹화 동영상 모두 활용 가능

OpenCV를 통해 실시간으로 확인 가능.
+ 녹화 동영상 (Ex. Youtube, mp4 등)에 직접 활용가능.

프로젝트 개요

데이터 수집

각각 자음, 모음에 대한 영상 촬영

데이터 전처리

입력 된 영상들을 .npy 형태로 변환 이후 하나의 npy로 병합

모델링

OpenCV, MediaPipe를 이용한 이미지 처리와 RNN 계열의 모델을 이용한 시계열 데이터 학습

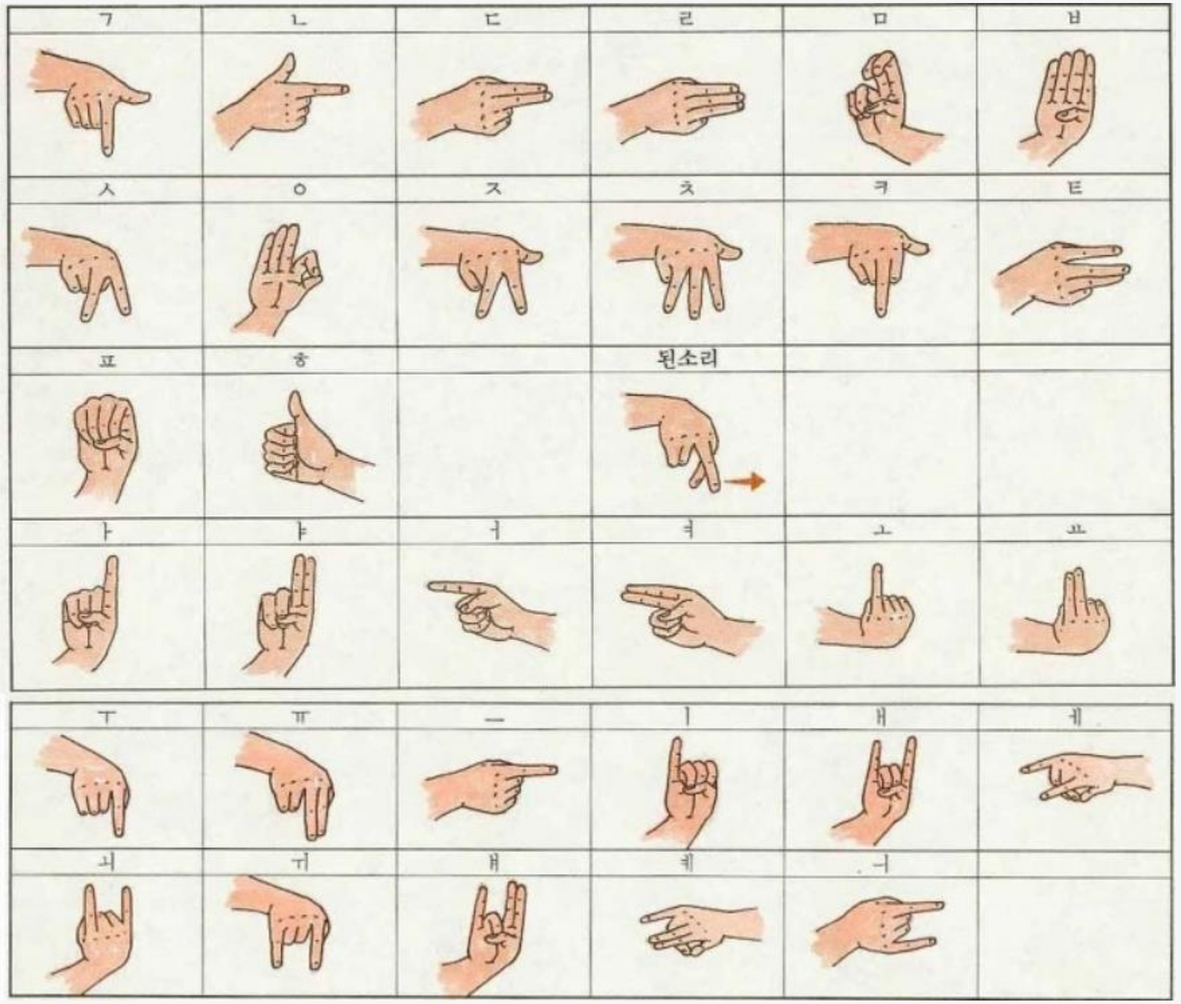
평가

모델의 정확도 평가 및 모델 시연

Data 설명

Finger Spelling

- 31개의 자음,모음에 대한 지문자 존재



출처: https://www.urimal.org/1222

Data 설명

Video Data

※ 동영상은 수없이 많은 사진들의 연속으로 구성
프레임 - 동영상을 구성하는 사진 한장 한장을 의미
FPS - 1초 단위로 몇 장의 프레임을 보여주는지

✓ **31개의 자음,모음에 대해 3명의 조원들 각각 학습 영상을 촬영**

- 기준 30 fps (일반적인 web cam의 fps)

'ㄱ'에 대한 지화

'ㅎ'에 대한 지화

'ㅏ'에 대한 지화

Data 설명

Data Processing

OpenCV

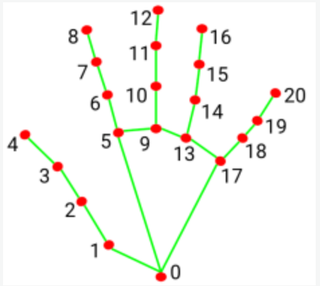
비디오 데이터 읽기
- cv2.VideoCapture

30fps = 1초당 30개의 이미지
하나의 비디오는 총 해당 이미지의 개수
만큼 데이터들이 저장되어 있음

MediaPipe

프레임 손동작 keypoint 검출
- mediapipe.HolisticDetector
- 한 손에 21개의 keypoints

우리가 필요로 하는 건 해당
손동작(오른손)의 keypoint 위치값들!



Keypoint 벡터, 각도 계산
- 벡터는 추가적으로 정규화 진행
- 각도로 joint를 얼마나 구부렸는지 파악

(20,2)의 vector 값 flatten, 15개의 angle 값,
label 값 을 concat해주어 최종적으로 (56,) 생성

if) data_length = 498

총 498번 반복해서
해당 프레임에서의 keypoint를 찾음

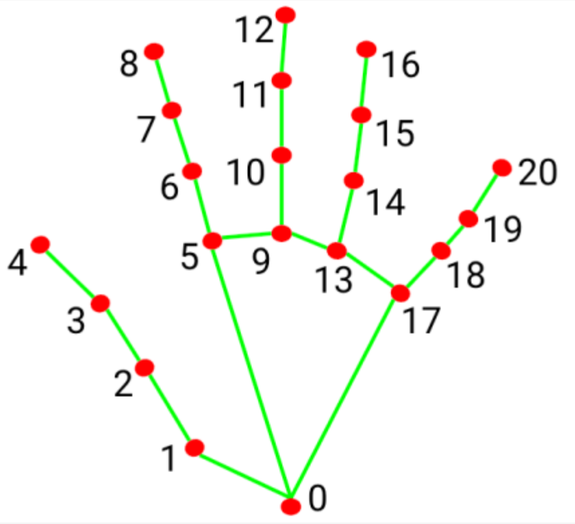
Data 의 shape은 (498,56)

출처: https://mediapipe.dev/index.html

Data 설명

Data Processing

Cf) Vector 값과 Angle을 뽑아내는 방법



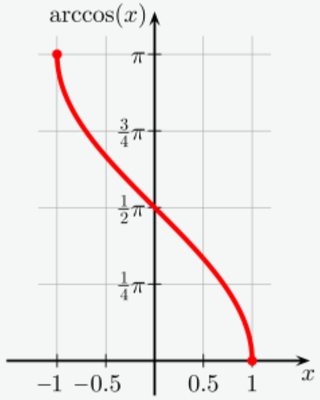

```
v1 = joint[[0,1,2,3,0,5,6,7,0,9,10,11,0,13,14,15,0,17,18,19], :2] # Parent joint
v2 = joint[[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20], :2] # Child joint
v = v2 - v1
# Normalize v
v = v / np.linalg.norm(v, axis=1)[:, np.newaxis]
```

Keypoint 값들의 좌표값 차이를 이용해서 Vector 값 계산

```
# Get angle using arcos of dot product
angle = np.arccos(np.einsum('nt,nt->n',
    v[[0,1,2,4,5,6,8,9,10,12,13,14,16,17,18],:],
    v[[1,2,3,5,6,7,9,10,11,13,14,15,17,18,19],:]))

angle = np.degrees(angle) # Convert radian to degree

angle_label = np.array([angle], dtype=np.float32)
```



Vector값들의 arccosine를 이용해서 Vector 값 계산

출처 : https://en.wiktionary.org/wiki/arccosine

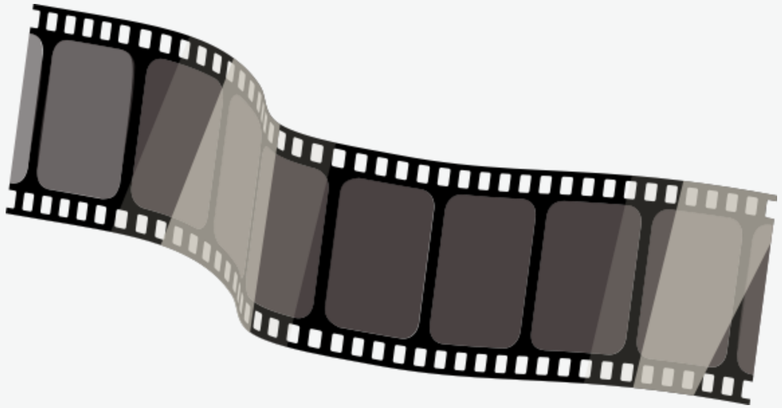
Data 설명

Generate Sequence Data

비디오 데이터는 여러 이미지가 Sequence 형태로 결합된 데이터!

Sequence_length = 10 , 각 자음,모음당 data_length - sequence length 만큼 반복

-우리는 총 10개의 프레임 단위로 sequence를 생성



10개의 프레임

ex) 만약 이전과 같이 data의 길이가 498이라면?

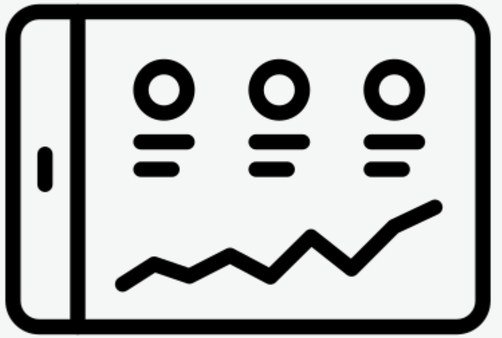
1~10, 2~11, 3 ~ 12 .... 489~498

해서 총 488개의 sequence 데이터가 생성되어짐

최종 data_shape = (488, 10,56) -> (data개수, seq_len, value)

마지막 value 값은 Label

모델링



각각 세개의 npy 파일 concat 진행하여 data로 활용

평가 지표 : Accuracy , F1 score를 지표로 사용

$$F1\ score = 2 \times \frac{recall \times precision}{recall + precision}$$

모델링

Frame Work

TensorFlow

Keras
Simple. Flexible. Powerful.

사용자 친화성과 확장성

일관되고 간결한 API

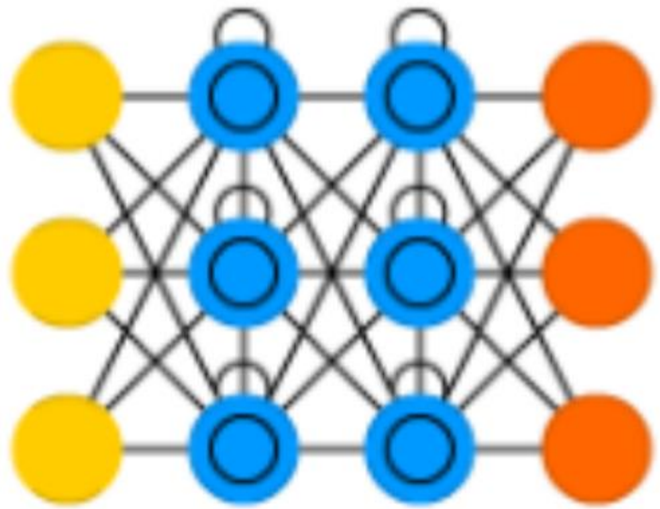
TensorFlow Lite

모바일, IoT 환경에서 TensorFlow 사용 가능
실시간으로 인식을 해주는 모델

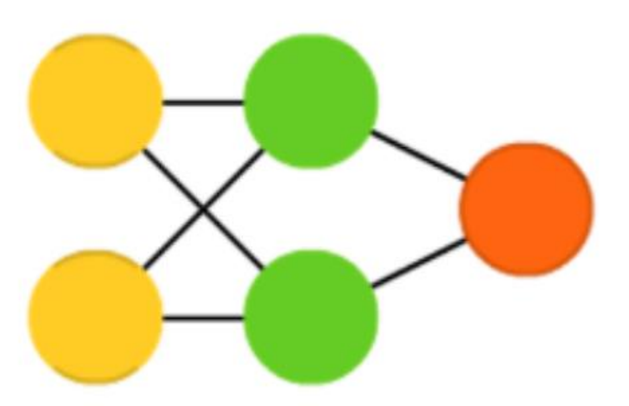
모델링

Model

Long / Short Term Memory (LSTM)



Feed forward neural networks (FFNN)



모델링

Modeling

- L2 Norm regularization
- ReLU
- Dropout(0.3)
- Categorical_CrossEntropy
- Adam
- ReduceLROnPlateau
- 21 Epoch (Early stopping)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 64)                30720
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense (Dense)                (None, 32)                2080
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 31)                1023
=================================================================
Total params: 33,823
Trainable params: 33,823
Non-trainable params: 0
_________________________________________________________________
```

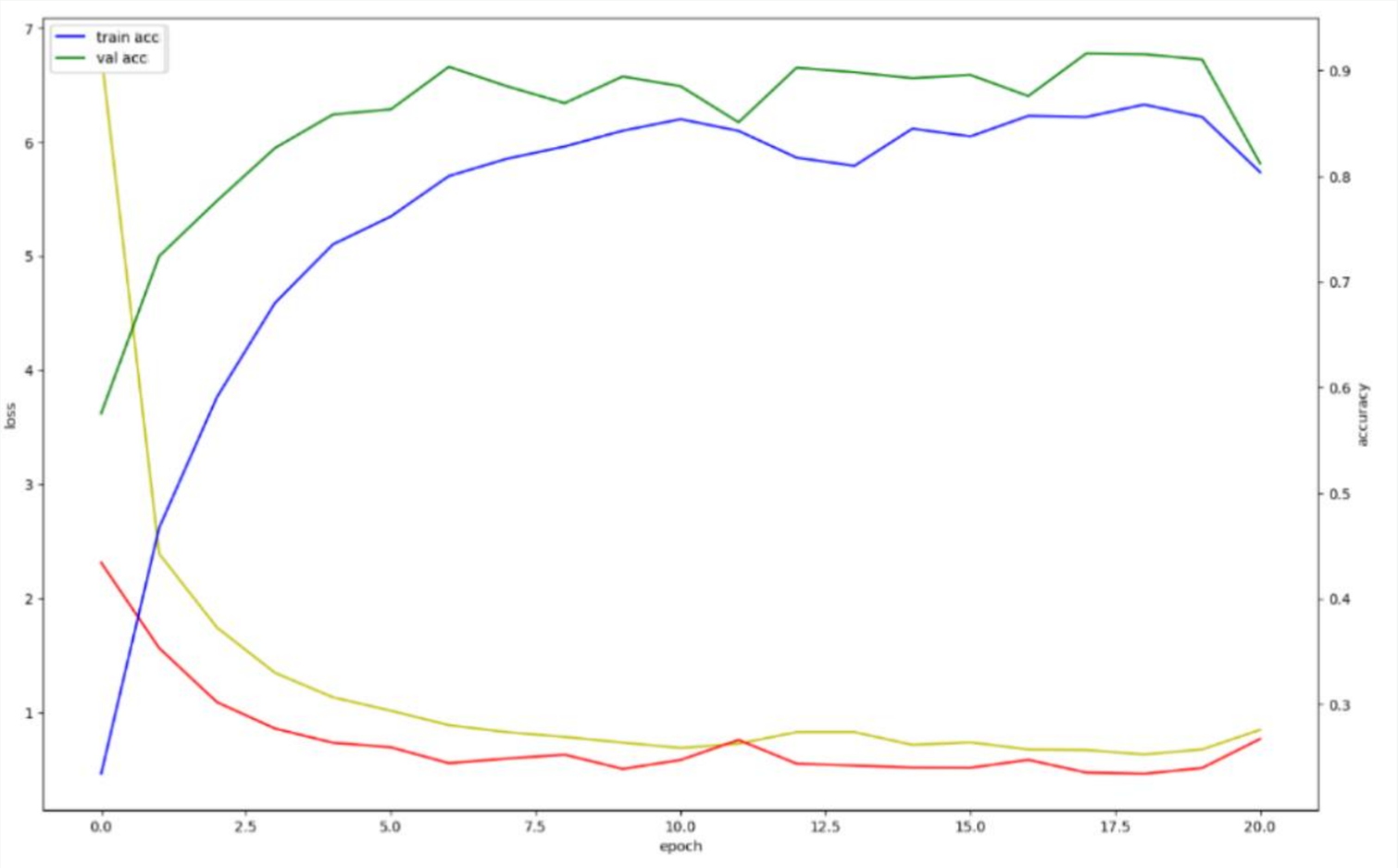
```python
model = Sequential([
    LSTM(64, activation='relu', input_shape=x_train.shape[1:3], kernel_regularizer=keras.regularizers.l2(0.01)),
    Dropout(0.3),
    Dense(32, activation='relu', kernel_regularizer=keras.regularizers.l2(0.01)),
    Dropout(0.3),
    Dense(len(actions), activation='softmax', kernel_regularizer=keras.regularizers.l2(0.01))
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc',metric_F1score])
```

평가

Accuracy

Score 향상



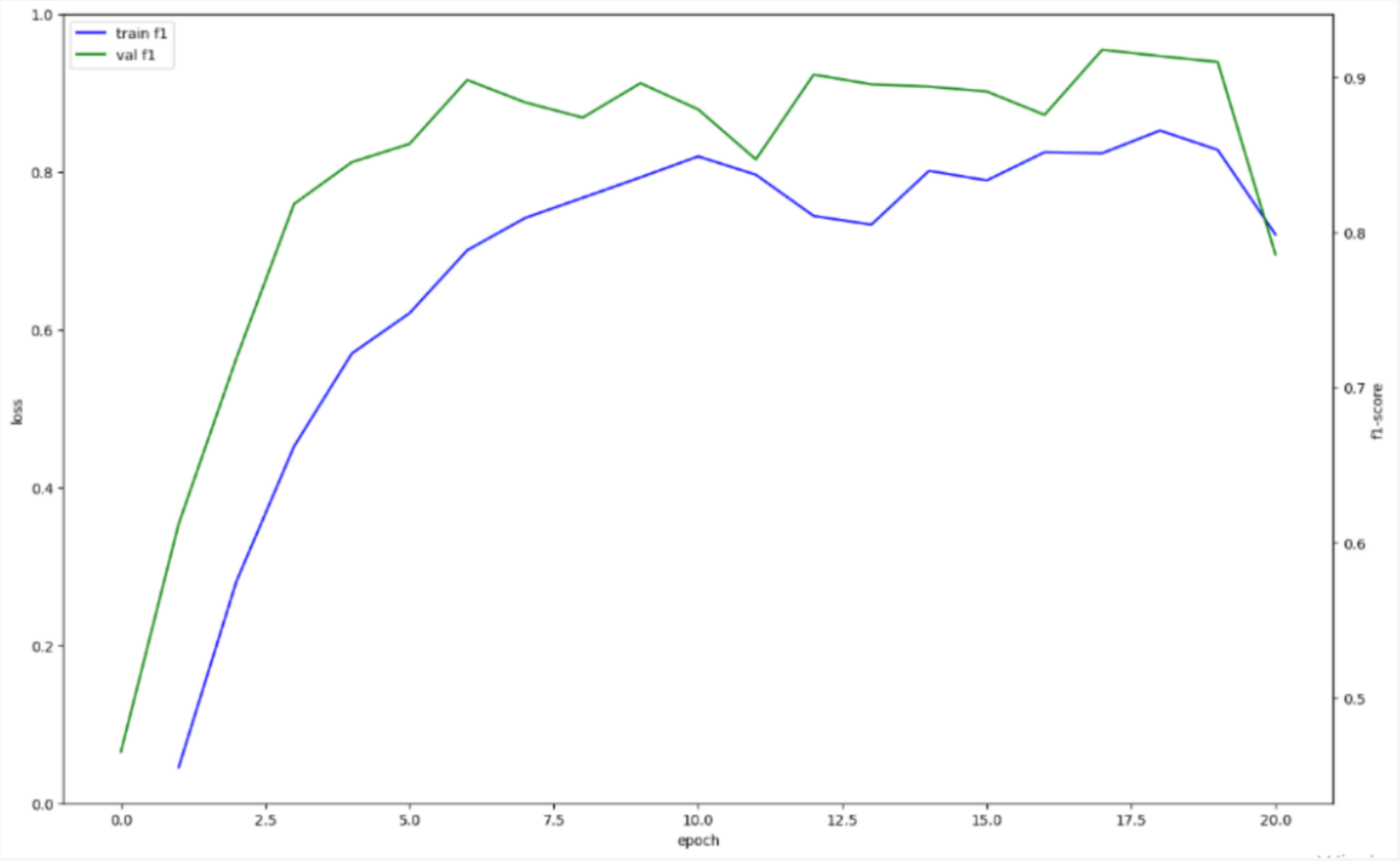
At Last Epoch

train loss : 0.7106, val loss : 0.7648
train Acc : 0.8440, val Acc : 0.8122

평가

F1-Score

Score 향상



At Last Epoch

train : 0.8384

validation : 0.7856

시안

녹화 동영상 - 자음

Youtube 동영상에 대입

유손생 Youtube : 수어 지문자 배우기 "수화로 내 이름은?"
https://www.youtube.com/watch?v=0eTc8GPMv74

시안

녹화 동영상 - 모음

Youtube 동영상에 대입

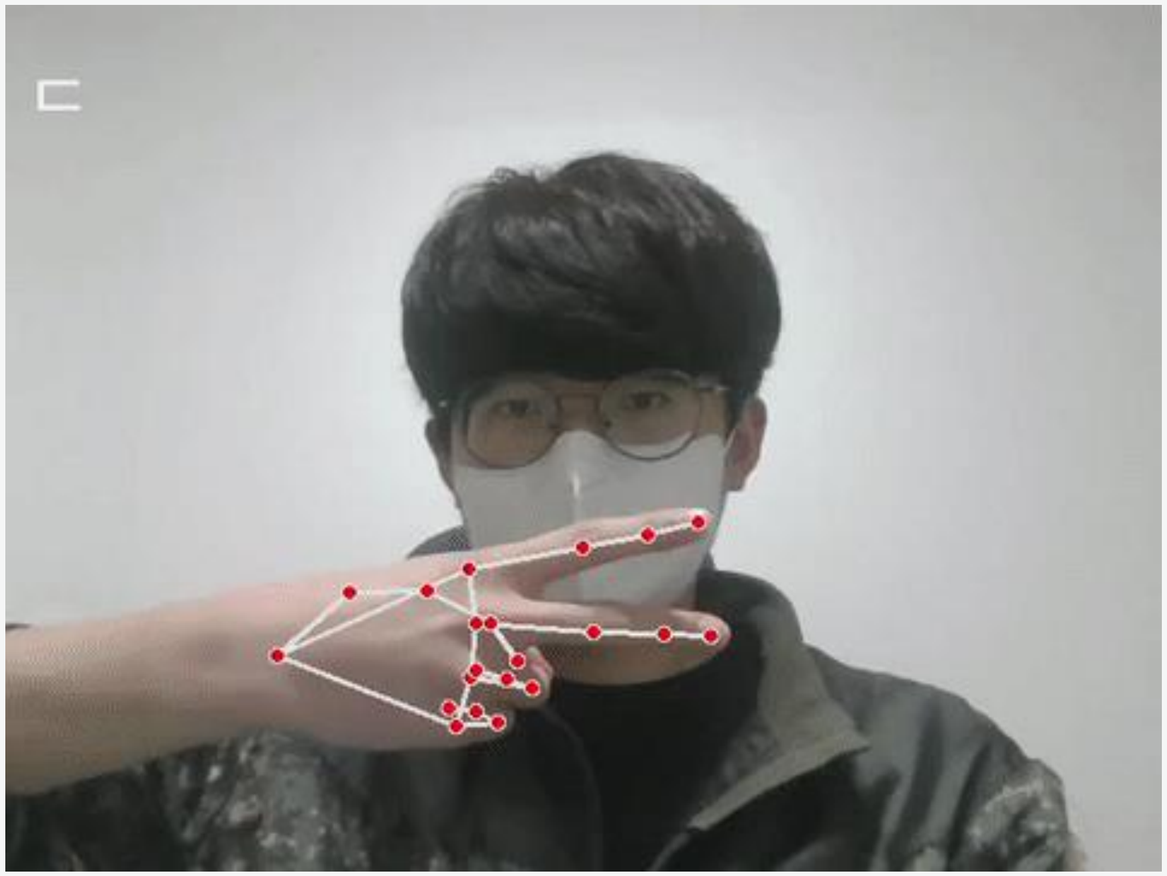
하이루비 Youtube : 수어(수화)배우기◇지화 [하이루비]
https://www.youtube.com/watch?v=CuwNdWOzPrA

시안

실시간 TFLite 모델 구동

시간 관계상 Q&A 시간에 진행

실시간 모델을 활용하여 "딥러닝"을 표현



개선방안

- 자모음이 아닌 단어 및 문장에 대한 학습
- 실시간 인식과 더불어 Jamos 패키지를 사용한 텍스트데이터 결합
- 양손의 joint 뿐만 아니라 얼굴, 팔 등의 비수지 신호에 대한 joint 첨가
- 더 많은 데이터 셋을 학습할 수 있다면 더 높은 성능 향상 기대

Thank you
&
QnA