

# 3주차 동작인식 AI 실습

## HandWard

손짓으로 설정하는 나만의 비밀번호

조영훈

# 목차

## ▶ 소개

프로젝트 소개

## ▶ 데이터 전처리

데이터 전처리 과정

## ▶ 결과 분석

학습 결과 및 성능 평가 시각화

## ▶ 결론

개선 방향

## ▶ 데이터 수집

데이터 수집 방법과 과정

## ▶ 사용 모델

사용된 딥러닝 모델 소개

## ▶ 기능 개발

개발한 AI 프로그램 소개  
결과물 시연

# HandWard 소개

## 프로젝트 소개

시각 장애우들을 위한 본인들만의 핸드 제스처를 통해 도어락 비밀번호를 설정하고 문을 열고 닫을 수 있는 프로젝트이다.



0



1



2



3



4



5



6



7



8



9

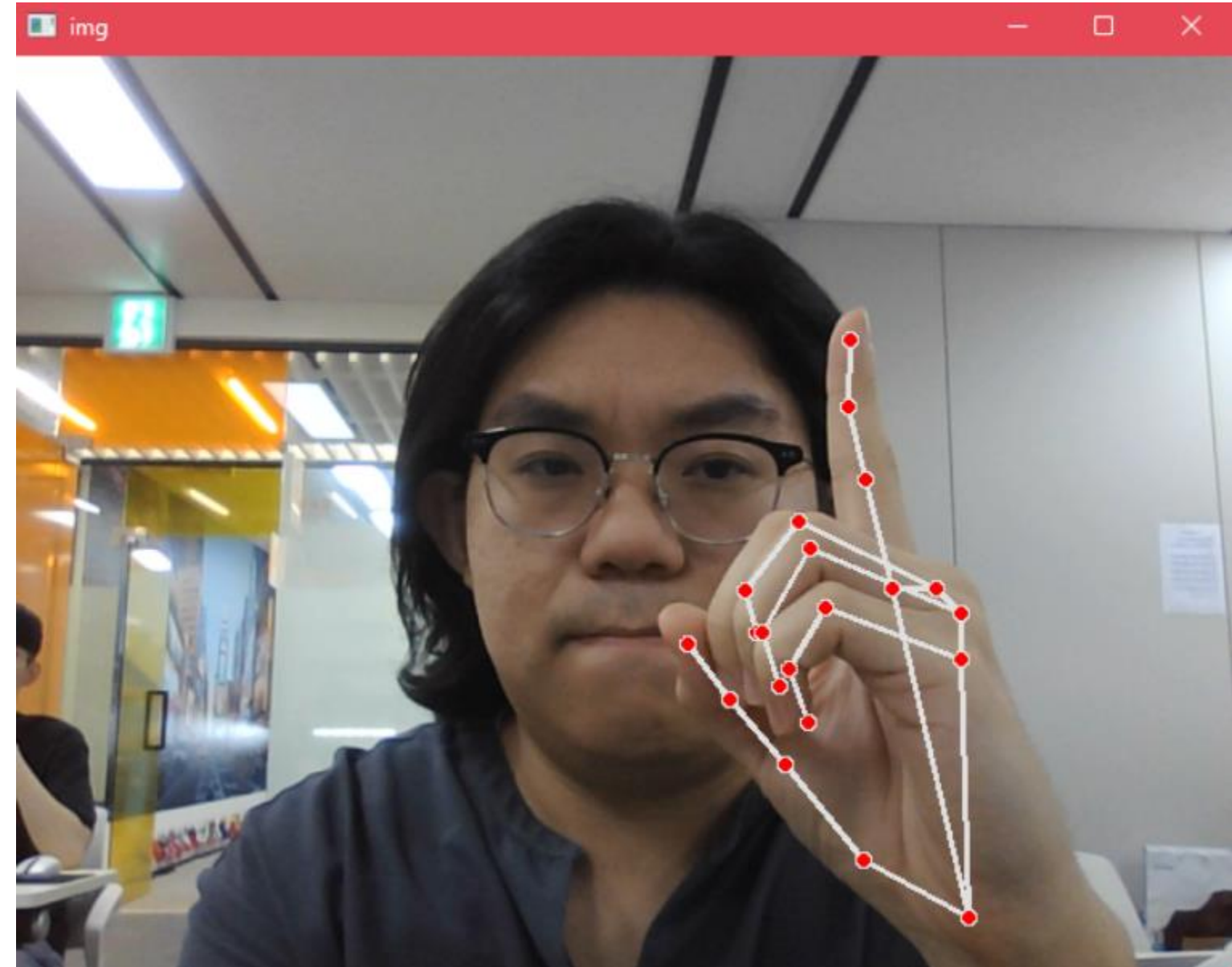


10

# 데이터 수집

## 데이터 수집 과정

웹 캠으로 인식한 손의 움직임을 통해  
수집한 핸드 제스처 데이터로  
손의 관절에 점을 위치시켜  
점들 간의 각도를 통해 움직임을 알아내  
.npy 파일로 변화 하여 저장한다.



# 데이터 전처리

## 데이터 전처리 과정

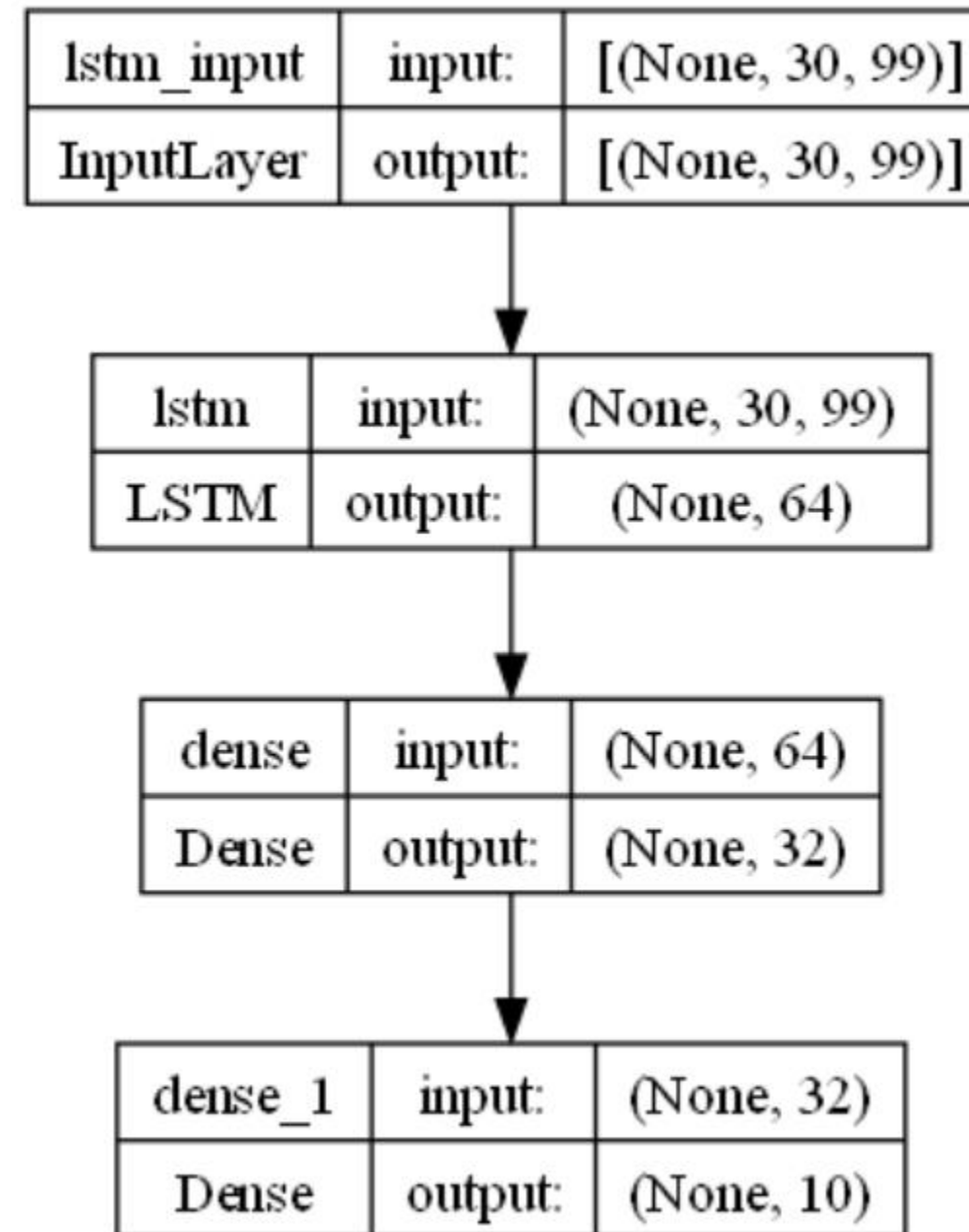
저장한 npy 파일을 불러와서 학습용 데이터와 라벨용 데이터를 나누고  
원 - 핫 인코딩을 진행하고  
학습용 데이터를 다시 학습용 데이터와 검증용 데이터 로 나누어 주었습니다.

```
def load_dataset(actions):  
    data = np.concatenate([  
        np.load(f"./data/dataset_HPW/seq_{action}_1689295505.npy") for action in actions  
    ], axis = 0)  
  
    return data
```

# 사용 모델

사용된 딥러닝 모델 소개

수업 시간에 사용했던 LSTM과  
망을 그대로 사용하였습니다

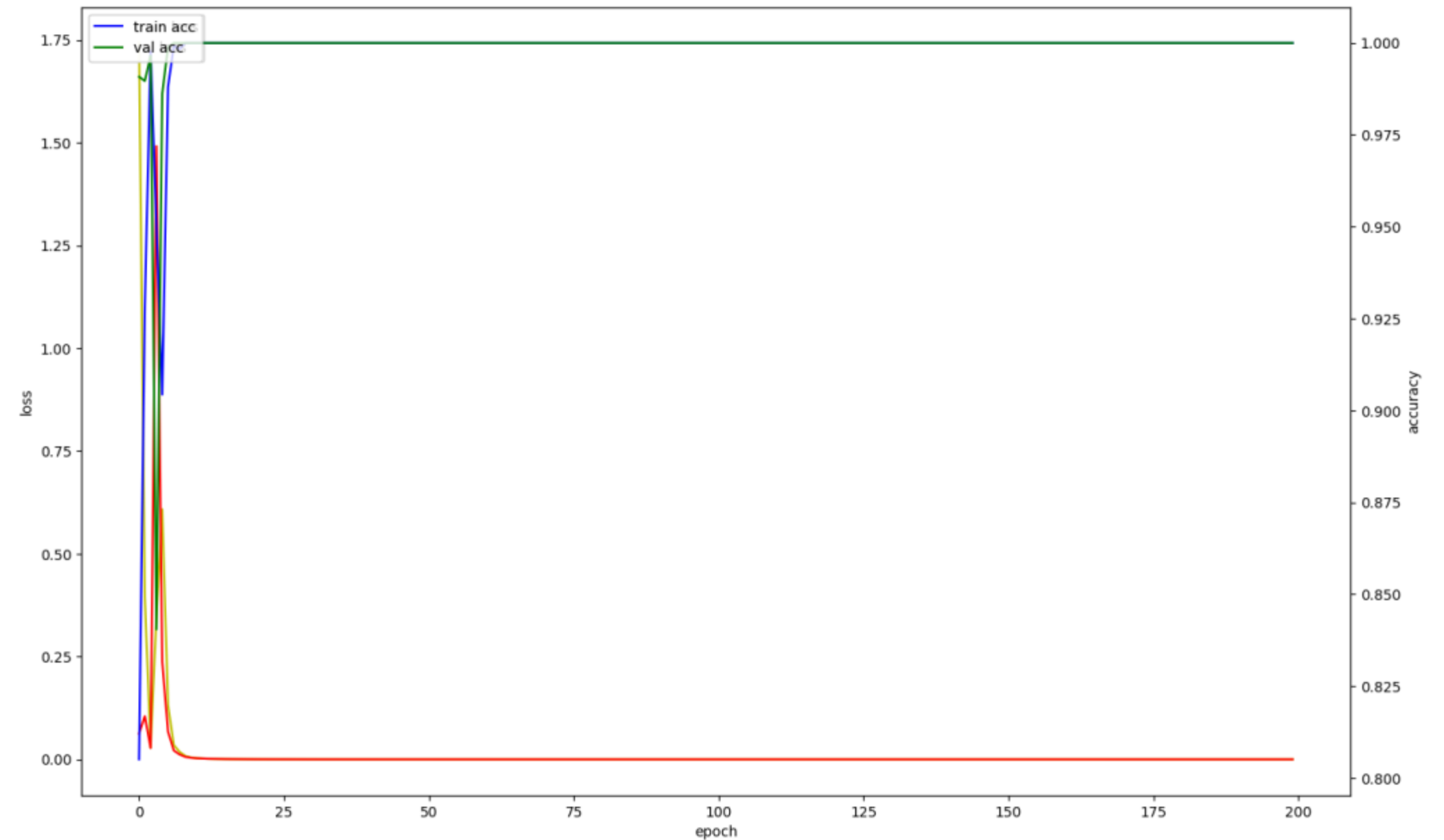




# 결과 분석

학습 결과 및 성능 평가 시각화

수집된 데이터가 편향적이라  
학습중 과적합이 생기는 모습을 볼 수 있습니다.



# 기능 개발

## 개발한 AI 프로그램 소개

비밀번호 등록 함수의 코드 일부분

상단의 코드는 수업시간에 사용했던 코드와 거의 유사하지만 비밀번호를 4번 입력 받기 위해 반복문이 추가됨

비밀번호 4자리를 입력받기 위해 리스트 하나를 추가 하였고 등록된 비밀번호를 화면에 나타내줌

```
# 텍스트 출력
cv2.putText(img, f'{this_action}', org=(int(res.landmark[0].x * img.shape[1]),
                                         int(res.landmark[0].y * img.shape[0] + 20)),
            fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1,
            color=(255, 255, 255), thickness=2)

set_password[idx] = this_action

cv2.putText(img, f'PassWard {set_password} is saved', org=(50,50),
            fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1,
            color=(255, 255, 255), thickness=2)

cv2.imshow('img', img)
if cv2.waitKey(1) == ord('q'):
    break

break
cap.release()
cv2.destroyAllWindows()

return set_password
```



# 기능 개발

## 개발한 AI 프로그램 소개

등록된 비밀번호와 입력된 비밀번호를  
비교하는 함수의 코드 일부분

마찬가지로 상단의 코드는 유사하지만  
등록된 비밀번호와 입력된 비밀번호가  
일치 할 경우와 다를 경우에  
화면에 출력되는 문장이 달라지도록  
조건문을 추가

```
if (input_password[0] == set_password[0]) and \
    (input_password[1] == set_password[1]) and \
    (input_password[2] == set_password[2]) and \
    (input_password[3] == set_password[3]):
    cv2.putText(img, f'{input_password} is correct', org=(50,50),
                fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1,
                color=(255, 255, 255), thickness=2)

    compared_value = 1
else :
    cv2.putText(img, f'{input_password} does not match.', org=(50,50),
                fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1,
                color=(255, 255, 255), thickness=2)

    compared_value = 0

cv2.imshow('img', img)
if cv2.waitKey(1) == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()

return (input_password, compared_value)
```

# 기능 개발

결과물 시연  
비밀번호 비교

입력된 비밀번호 : ['0', '7', '1', '4']  
비밀번호가 일치 합니다. 문이 열립니다.

# 결론

개선 방향

## 결론 1

### 다양한 데이터 확보

현재 웹캠으로 직접 촬영하면 데이터를 수집하였고 개인 프로젝트로 진행되어 다양한 사람들의 핸드 제스처를 구하기가 힘들어 편향된 데이터로 학습을 하여 과적합 문제가 나타나고 있습니다

## 결론 2

### 기능에 대한 개선

현재 영상 화면에 출력되고 있는 텍스트들을 시각 장애우들을 위해 TTS를 사용한 음성으로 처리할수 있다면 더 좋을 것 같습니다.

# THANK YOU

감사합니다.