

캡스톤 디자인 I

20221999 이경림

20191770 김하현

20222028 송채린

2025/06/11



실시간 수어 텍스트 번역 프로그램

+

02



목차

01

프로젝트 개요 및 목표

02

시스템 구성도 및 전체 구조

03

주요 기능 설명

04

알고리즘 및 설계 방법

05

성능 및 구현 결과

06

팀원 역할

07

시연 영상

08

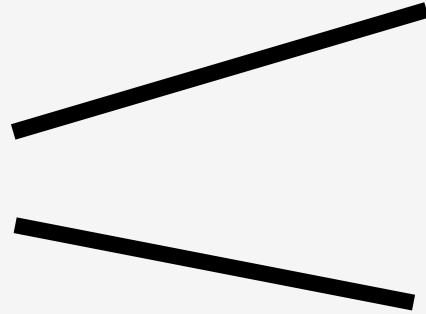
추후 계획

01

프로젝트 개요

- 목표
- 개요

“



Key



:

Value



카메라로 인식해 수어 -> 텍스트 변환

청각장애인과 비장애인 간의 원활한 의사소통을 지원.

프로젝트 개요

개요

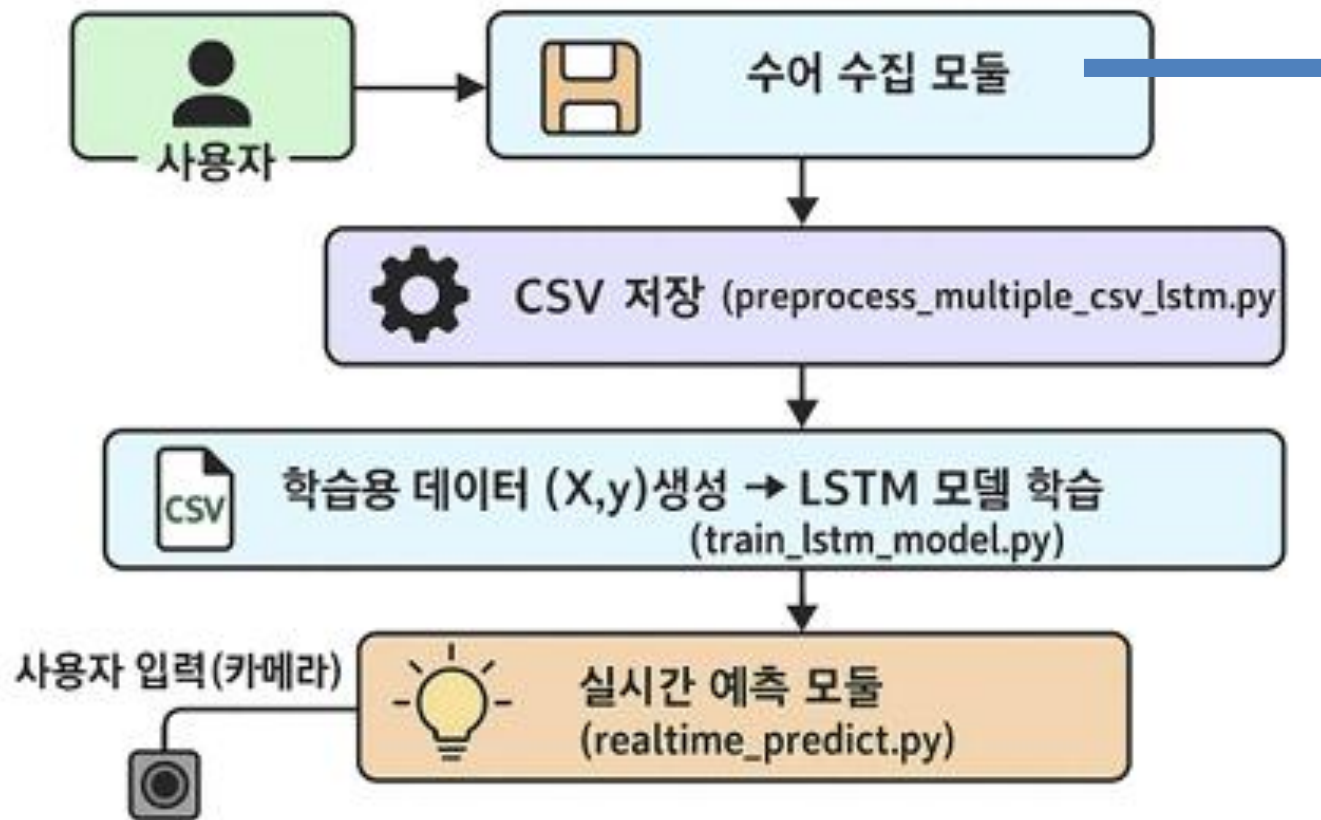
- 수어 사용자와 비수어 사용자 간의 소통 장벽을 해소하기 위해 개발된 수어 텍스트 번역 시스템.

02

시스템 구성도 및 전체 구조

- 아키텍처
- 전체 시스템 구성도

시스템 구성도 및 전체 구조



처리 구조

1. Mediapipe로 손 키포인트 추출 (양손, 총 126차원)
2. Deque로 시퀀스 구성 (30 프레임)

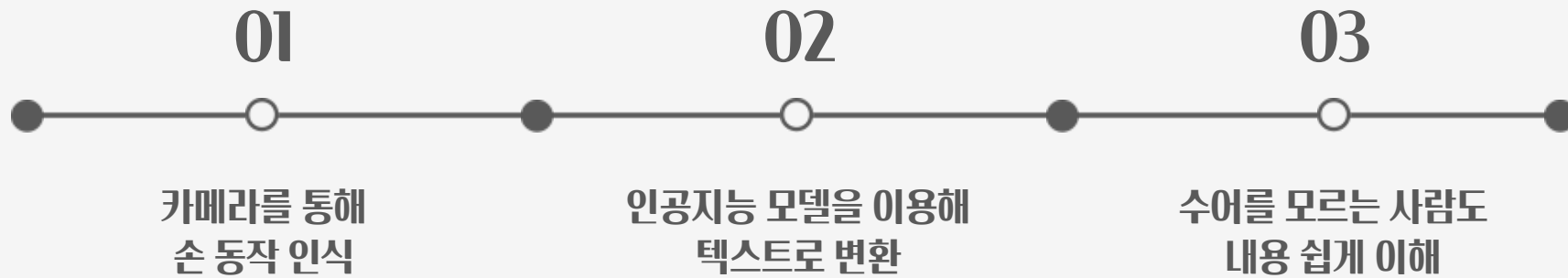
+

03

주요 기능 설명

- 주요 기능

주요 기능 설명



+

04

알고리즘 및 설계 방법

- 설계 및 AI 모델

10

- ▼ data
 - 나_sequences.csv
 - 너_sequences.csv
 - 만나다_sequences.csv
 - 반갑다_sequences.csv
 - 사랑하다_sequences.csv
 - 안녕하세요_sequences.csv
 - 처음_sequences.csv
 - 헤어지다_sequences.csv
 - american_sequences.csv
 - yes_sequences.csv

- > holistic
- > **models**
- > processed_holistic
- > processed_lstm
- > scripts

- ▼ videos
 - ▶ american.mp4
 - ▶ yes.mp4

- ◆ .gitignore

- 🔗 preprocess_multiple_csv... M
- 🔗 realtime_predict.py M
- ≡ requirements.txt
- 🔗 run_all_dual_hand.py
- 🔗 save_sequence_data.py
- 🔗 train_lstm_model.py

+

알고리즘 및 설계 방법

○ 데이터 수집 : save_sequence_data.py

양손 126차원 좌표 × 30프레임을 csv로 저장

○ 전처리 및 라벨링 : preprocess_multiple_csv_lstm.py

여러 CSV 병합, LabelEncoder로 인코딩, One-hot 변환

○ 모델 학습 : train_lstm_model.py

LSTM + Dropout 사용, best model 저장

○ 실시간 예측: realtime_predict.py

예측 결과를 한글로 화면에 출력 (draw_korean_text())

알고리즘 및 설계 방법

○ 데이터 구조

- 30 x 126 시퀀스 입력 (양손 3D 좌표 63 x 2손)

○ 모델 구조

```
Sequential([
  LSTM(64, return_sequences=False),
  Dropout(0.3),
  Dense(num_classes, activation='softmax')
])
```

- 교차검증 사용해 정확도 향상

○ 예측 방식

- Deque로 15 프레임 버퍼 → 3초 정도의 움직임 인식
- 확률 기반으로 최고 confidence 예측값 추출

data

- 나_sequences.csv
- 너_sequences.csv
- 만나다_sequences.csv
- 반갑다_sequences.csv
- 사랑하다_sequences.csv
- 안녕하세요_sequences.csv
- 처음_sequences.csv
- 헤어지다_sequences.csv
- american_sequences.csv
- yes_sequences.csv

> holistic

> models

> processed_holistic

> processed_lstm

> scripts

videos

▶ american.mp4

▶ yes.mp4

🔍 .gitignore

🔗 preprocess_multiple_csv... M

🔗 realtime_predict.py M

📄 requirements.txt

🔗 run_all_dual_hand.py

🔗 save_sequence_data.py

🔗 train_lstm_model.py



+

05

성능 및 구현 결과

15

+

성능 및 구현 결과

성공 기능

실시간 예측, 예측 라벨
+ 확률 출력, 자동 실행 스크립트

정확도 측정

Train : 1.0

Validation : 0.94

미구현 기능

영상통화 번역, 통역사 서비스
연동, 접근성 기능

성능 및 구현 결과

Epoch 1/30

15/17 ————— 0s 4ms/step - accuracy: 0.1927 - loss: 2.1522

Epoch 1: val_loss improved from inf to 1.73068, saving model to models/gesture_lstm_model_dual.h5

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

17/17 ————— 1s 14ms/step - accuracy: 0.2087 - loss: 2.1287 - val_accuracy: 0.3529 - val_loss: 1.7307

Epoch 2/30

16/17 ————— 0s 3ms/step - accuracy: 0.4276 - loss: 1.5719

Epoch 2: val_loss improved from 1.73068 to 1.47114, saving model to models/gesture_lstm_model_dual.h5

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

17/17 ————— 0s 6ms/step - accuracy: 0.4270 - loss: 1.5682 - val_accuracy: 0.5882 - val_loss: 1.4711

Epoch 3/30

15/17 ————— 0s 4ms/step - accuracy: 0.5901 - loss: 1.3290

Epoch 3: val_loss improved from 1.47114 to 1.23990, saving model to models/gesture_lstm_model_dual.h5

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

17/17 ————— 0s 6ms/step - accuracy: 0.5968 - loss: 1.3176 - val_accuracy: 0.6176 - val_loss: 1.2399

Epoch 4/30

16/17 ————— 0s 3ms/step - accuracy: 0.7501 - loss: 1.0474

Epoch 4: val_loss improved from 1.23990 to 0.97648, saving model to models/gesture_lstm_model_dual.h5

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

17/17 ————— 0s 6ms/step - accuracy: 0.7516 - loss: 1.0390 - val_accuracy: 0.7647 - val_loss: 0.9765

Epoch 5/30

16/17 ————— 0s 3ms/step - accuracy: 0.7696 - loss: 0.8347

Epoch 5: val_loss improved from 0.97648 to 0.70363, saving model to models/gesture_lstm_model_dual.h5

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

17/17 ————— 0s 6ms/step - accuracy: 0.7721 - loss: 0.8266 - val_accuracy: 0.8235 - val_loss: 0.7036

성능 및 구현 결과

```

Epoch 25: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0197 - val_accuracy: 0.9412 - val_loss: 0.2237
Epoch 26/30
14/17 ----- 0s 4ms/step - accuracy: 1.0000 - loss: 0.0183
Epoch 26: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0179 - val_accuracy: 0.9412 - val_loss: 0.2258
Epoch 27/30
15/17 ----- 0s 4ms/step - accuracy: 1.0000 - loss: 0.0157
Epoch 27: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0158 - val_accuracy: 0.9412 - val_loss: 0.2197
Epoch 28/30
15/17 ----- 0s 4ms/step - accuracy: 1.0000 - loss: 0.0135
Epoch 28: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0137 - val_accuracy: 0.9412 - val_loss: 0.2318
Epoch 29/30
15/17 ----- 0s 4ms/step - accuracy: 1.0000 - loss: 0.0162
Epoch 29: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0163 - val_accuracy: 0.9412 - val_loss: 0.2283
Epoch 30/30
14/17 ----- 0s 4ms/step - accuracy: 1.0000 - loss: 0.0108
Epoch 30: val_loss did not improve from 0.21512
17/17 ----- 0s 6ms/step - accuracy: 1.0000 - loss: 0.0116 - val_accuracy: 0.9412 - val_loss: 0.2343

```

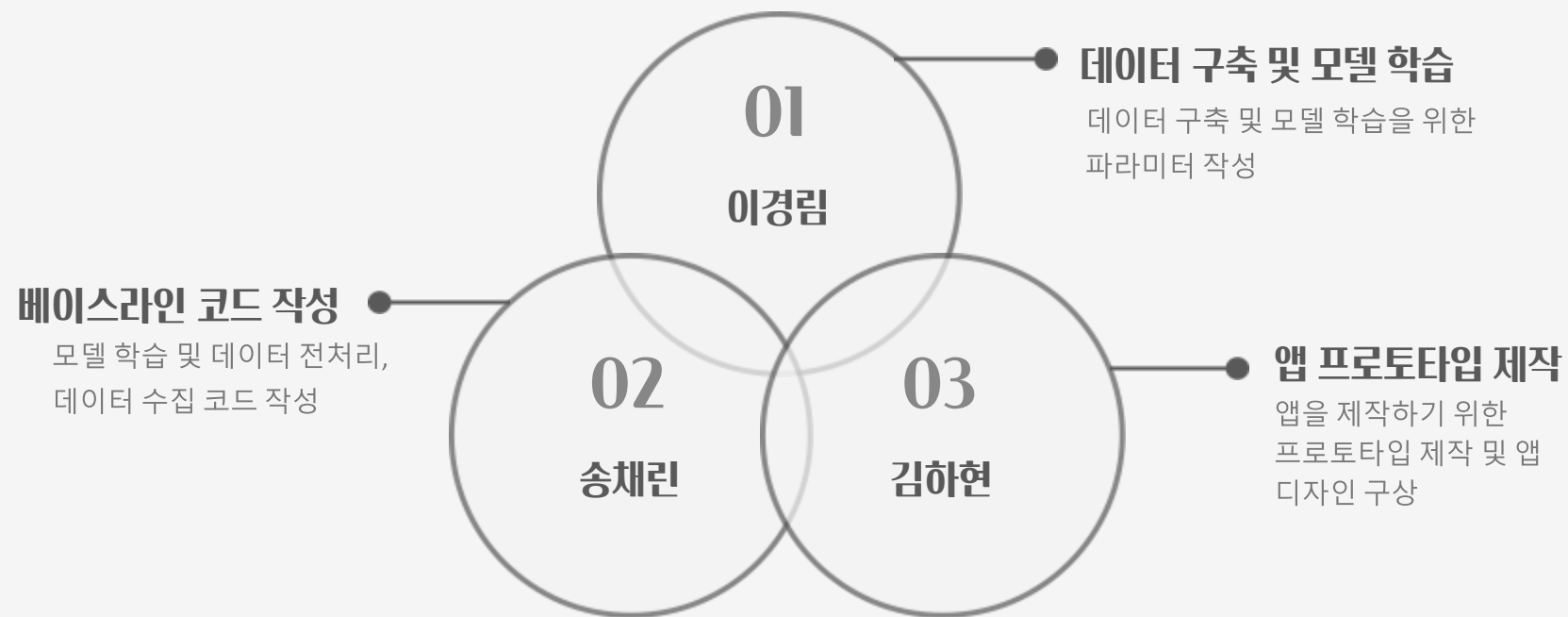
최종 정확도와 손실

06

팀원 역할

- 팀원 역할

팀원 역할



+

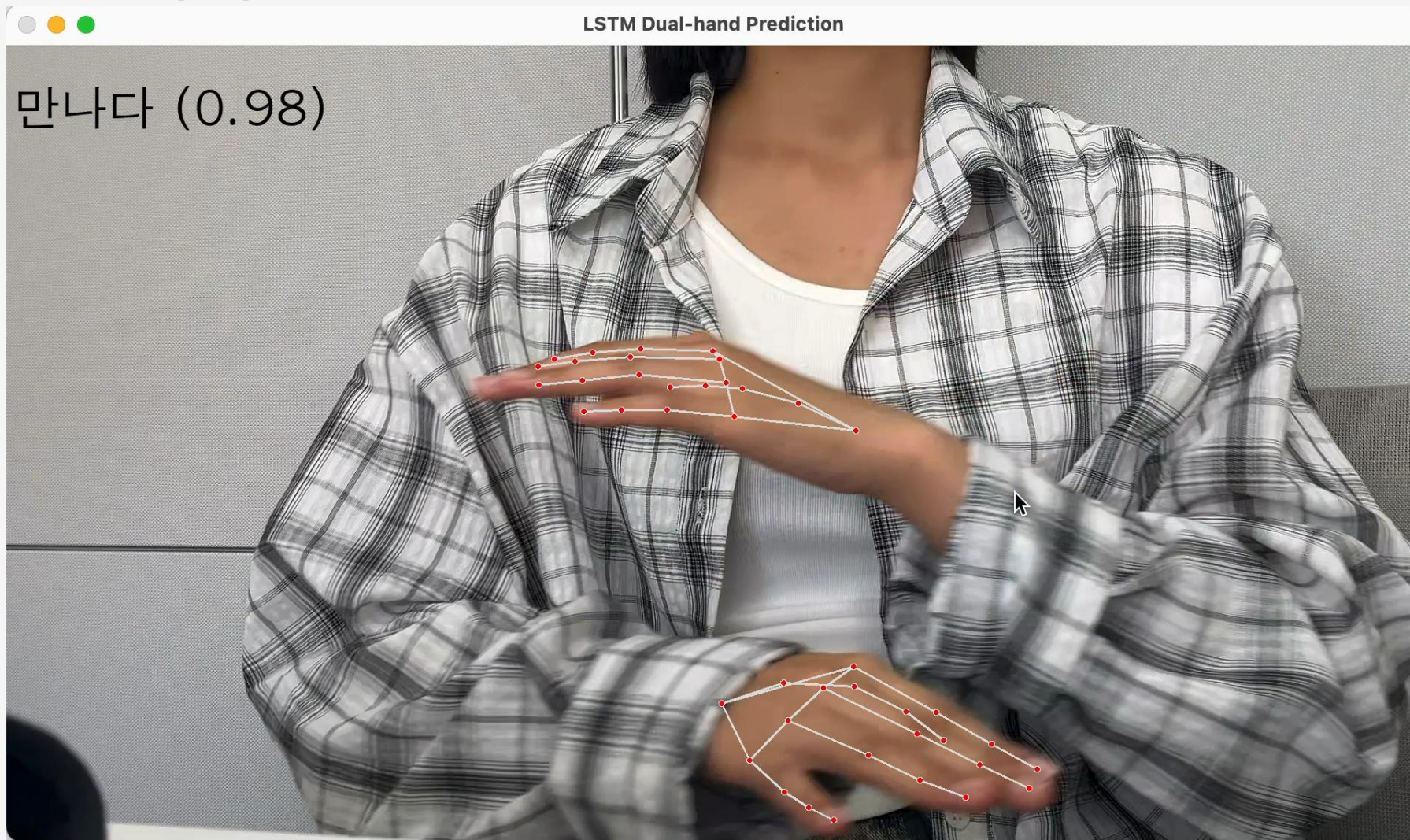
07

시연 영상

- 시연 영상

+

시연 영상 1



+

08

추후 계획

19

02 제목을 입력해주세요

추후 계획

- 모델 고도화 : 데이터 증대
- 자연어 처리 : 문장 단위 번역 가능케
- 영상통화 앱 개발
+ 전문 수어 통역사 연결 서비스

캡스톤 디자인 I

20221999 이경림
20191770 김하현
20222028 송채린

2025/06/11



발표를 들어주셔서
감사합니다 :))