

14주차 연구노트

○ 14주차 계획 (13주차 피드백 반영)

● 음성 텍스트화 + Time stamp 처리 (Google Cloud API)

※ '구문 → 단어'로 폴더 트리 구조 변경 (5.30)

→ 한글은 자음과 모음으로 이루어져 있는데, 입술 모양으로 모든 음절을 구분짓기 어려움.

※ '단어 → 음절'로 폴더 트리 구조 변경 (6.2)

조음 위치			양순음	치조음	경구개음	연구개음	성문음
조음 방법							
장앵음	파열음	평음	ㅂ	ㄷ		ㄱ	
		유기음	ㅃ	ㄸ		ㅋ	
		경음	ㅍ	ㅌ		ㆁ	
	마찰음	평음		ㅅ			ㅎ
		유기음					
		경음		ㅆ			
	파찰음	평음			ㅈ		
		유기음			ㅊ		
		경음			ㅉ		
공명음	비음		ㅁ	ㄴ		ㅇ	
	유음			ㄹ			

⇒ 입술(양순음)로 구분할 수 있는 자음은 ㅂ, ㅃ, ㅍ, ㅁ 밖에 없음

⇒ 따라서, '양순음 + 모음'의 음절 단위로 데이터를 수집하기로 결정.

● 음절(모음) 단위 데이터 직접 추출(총 7명 + a)

원 단어

모음화

안녕 → 안여
 아마도 → 아마오
 힘들어 → 임으어
 놀라워 → 오아워

괜찮아	→	웬안아
누구	→	우우
좋아	→	오아
고마워	→	오아워
즐려	→	오여
사랑해	→	아앙애

⇒ **추출할 음절** : 아 여 마 오 애 임 으 어 워 웬 우 안 (총 12개 음절)

31 32

● Augmentation(총 4가지 방식) 진행 → 데이터량 ↑

1. Flip(좌우 반전)
2. Crop(자르기)
3. Noise(노이즈)
4. Distort(왜곡)
5. Brightness(밝기 조절)

→ ~~현재 데이터 량 : 12음절 * 5(원본 + augmentation) * 10(이미지 개수) * 7셋 = 4200개 (6.1)~~

→ ~~현재 데이터 량 : 12음절 * 5(원본 + augmentation) * 10(이미지 개수) * 13셋 = 7800개 (6.2)~~

→ ~~현재 데이터 량 : 12음절 * 6(원본 + augmentation) * 10(이미지 개수) * 28셋 = 20160개 (6.3)~~

→ 현재 데이터 량 : 12음절 * 6(원본 + augmentation) * 10(이미지 개수) * 34셋 (Val : 27개 + Test : 7개) = 24480개 (6.4)

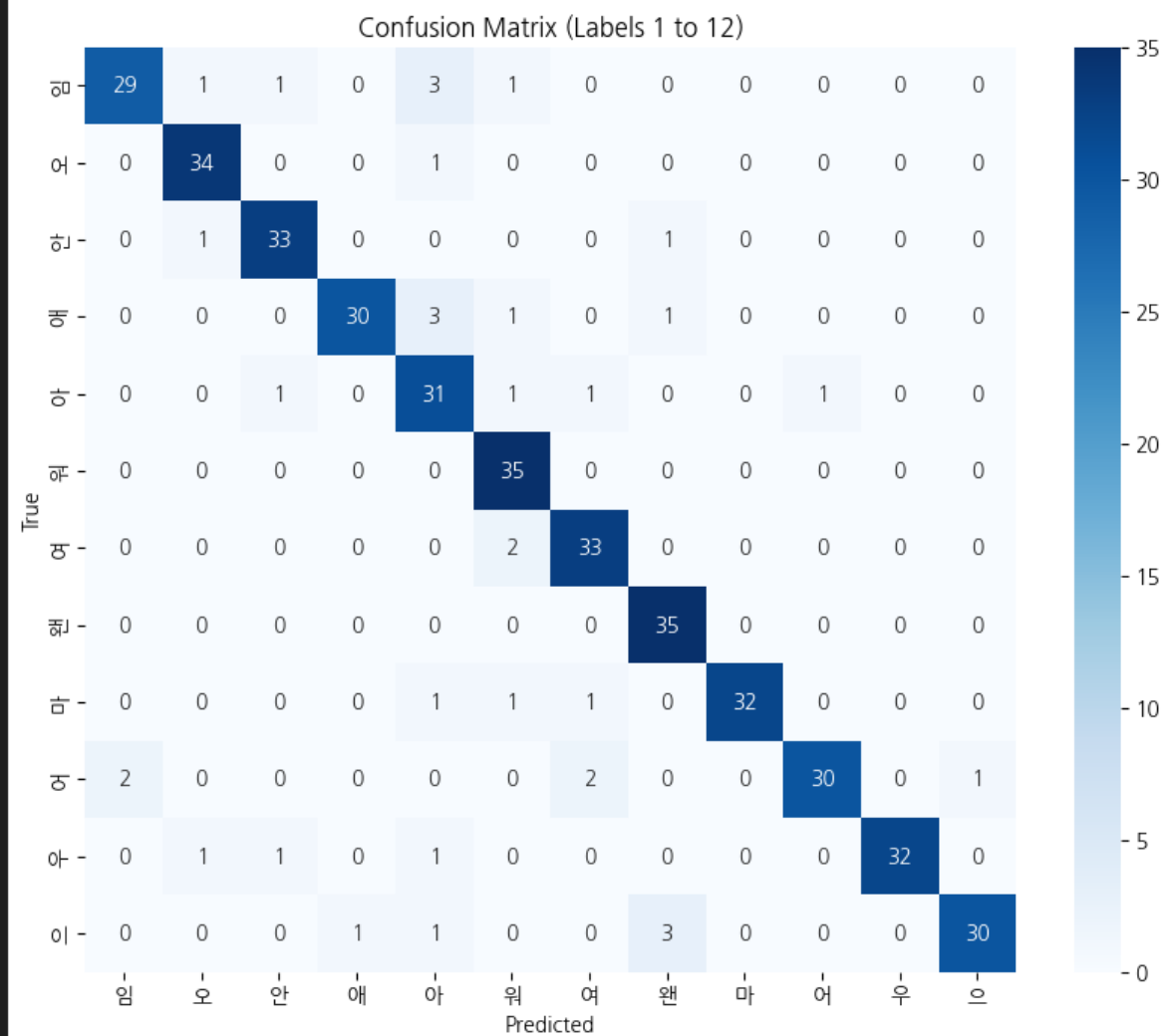
→ **Augmentation이 시퀀스 데이터에 대한 Train을 방해하는 것 같아 원본 데이터로만 진행 (6.2)**

⇒ **정확도 항상 확인**

⇒ 추가 원본 데이터 수집 필요

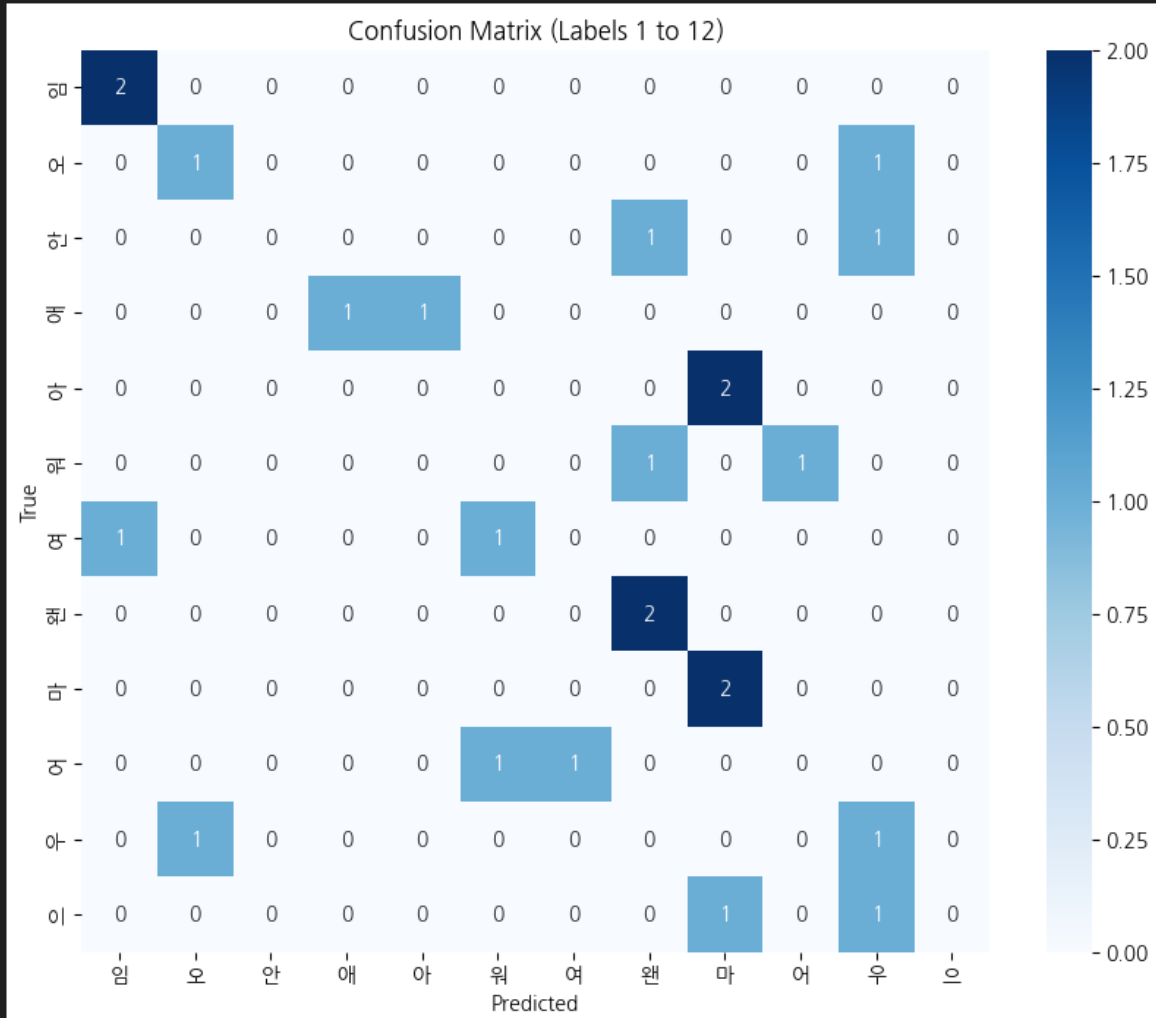
- Trainset에 대한 Accuracy 측정 Dropout(0.2) , Epoch : 50 → **정확도 : 91.43%**

Evaluating: 100% | 105/105 [00:01<00:00, 52.51it/s]
Accuracy: 0.9143



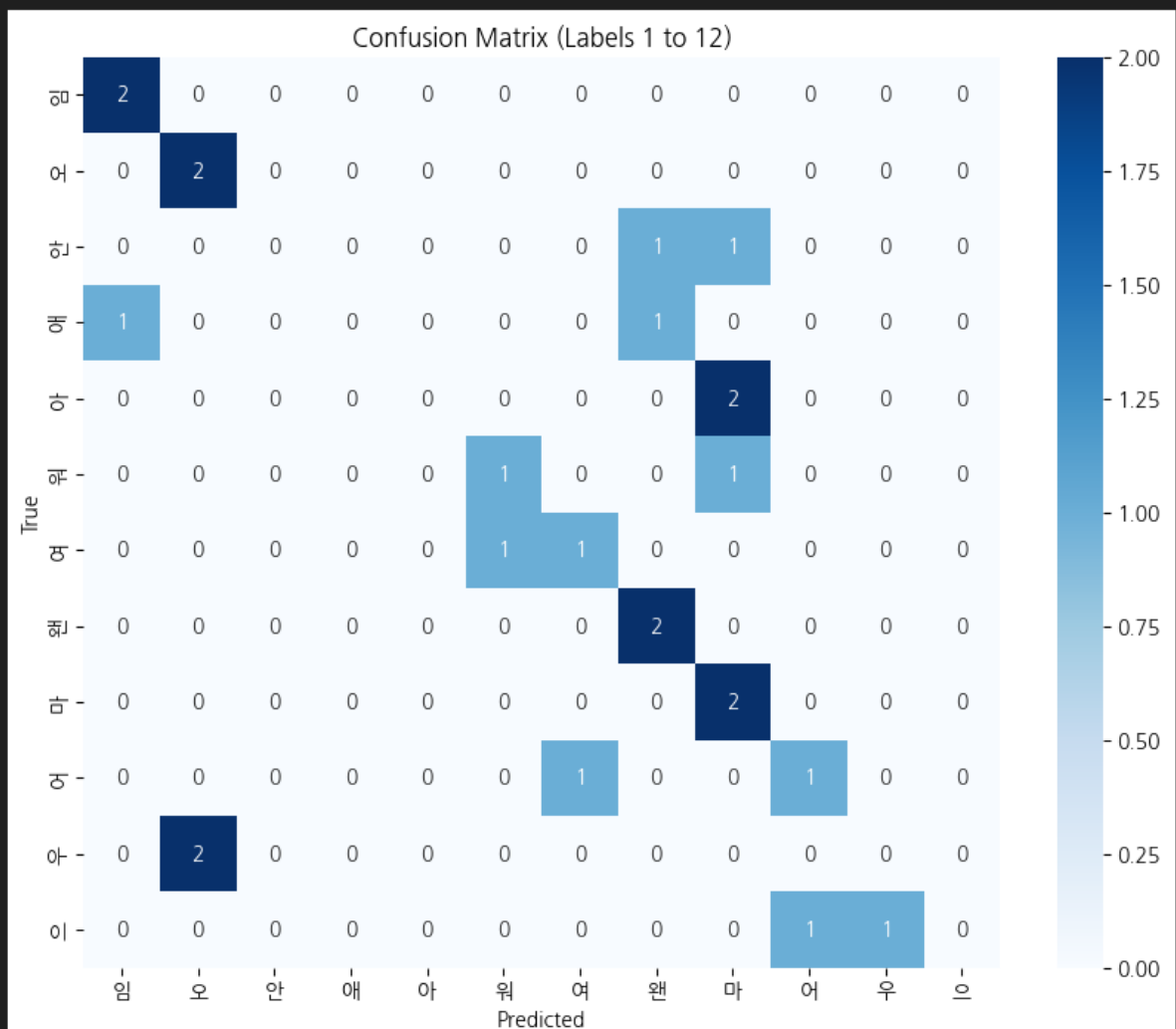
- Testset에 대한 Accuracy 측정 (**Dropout : 0.2**) , Epoch : 50 → **정확도 : 37.5%**

Evaluating: 100% | 6/6 [00:00<00:00, 27.76it/s]
Accuracy: 0.3750

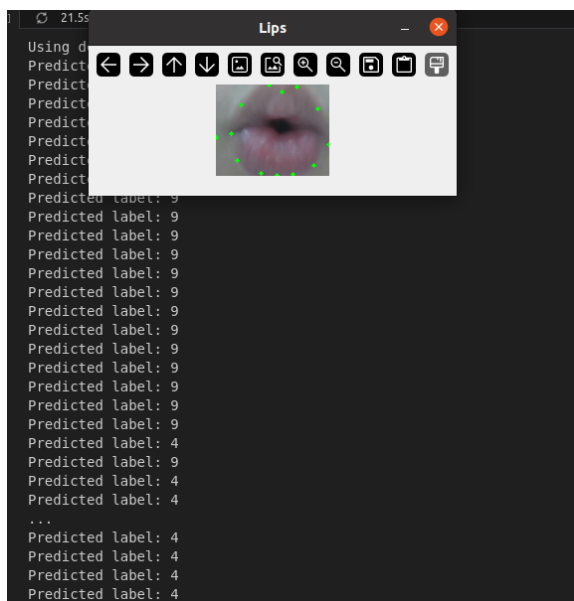


- Testset에 대한 Accuracy 측정 (**Dropout : 0.8**) , Epoch : 50 → **정확도 : 45.8%**

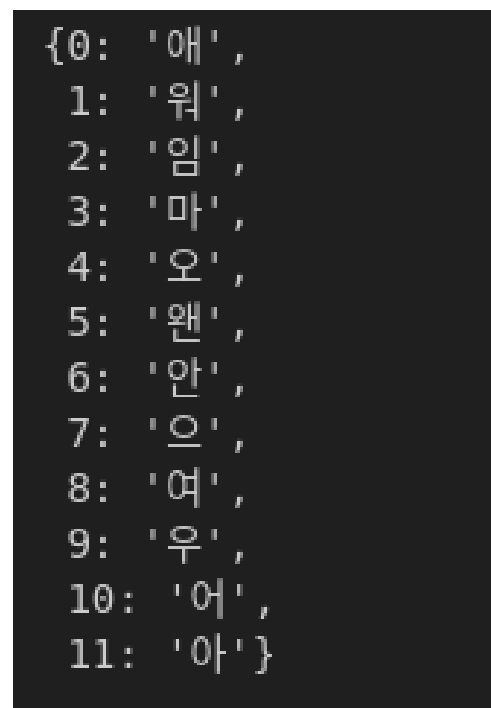
Evaluating: 100% | 6/6 [00:00<00:00, 27.97it/s]
Accuracy: 0.4583



● Depth 카메라(D435i)를 통해 실시간으로 입술 부분만 잘라내기



▲ 입술 모양 인식 후 결과 출력



▲ 사전에 라벨링된 음절들

- ⇒ 사전에 숫자로 각 음절을 라벨링
- ⇒ D435i가 입술 부분만 인식해 입 모양 확인
- ⇒ 라벨링 된 음절과 인식한 음절이 같음을 확인

- 학습 코드 (MobileNet V2 + LSTM)

[lip_reading_train.ipynb](#)

- 입술 부분만 잘라서 저장하는 코드

[lip_crop.ipynb](#)

- Augmentation 코드

[augmentation.ipynb](#)

○ 문제점 파악

- Augmentation이 의미가 있는 걸까?
- 지속적으로 Accuracy가 50%를 넘지 못하는 문제 발생
- 데이터셋이 너무 적어 충분한 학습과 테스트가 이루어지지 않음
- Epoch 설정값 만큼 돌면서 Loss값이 최소일 때 멈추지 못함

○ 수정사항

1. 데이터셋 증가 : 4200 → 28800장 수집 (더 늘릴 예정)
2. Early Stopping → Val dataset을 기준으로 patience 설정한 epoch 을 지났는데도 loss 가 떨어지지 않으면 중단
3. Random Augmentation → 먼저 Augmentation 을 적용하는 것이 아닌, 학습 시에 랜덤으로 Augmentation을 적용
4. 5-fold validation
5. Train dataloader 파라미터 수정 : Shuffle = False → True
6. Batch 사이즈 조정해보기
7. Optimizer 에 weight_decay 추가하기 → Overfitting 방지
⇒ weight_decay = 1e-4

○ 수정 사항 반영 후 결과 확인

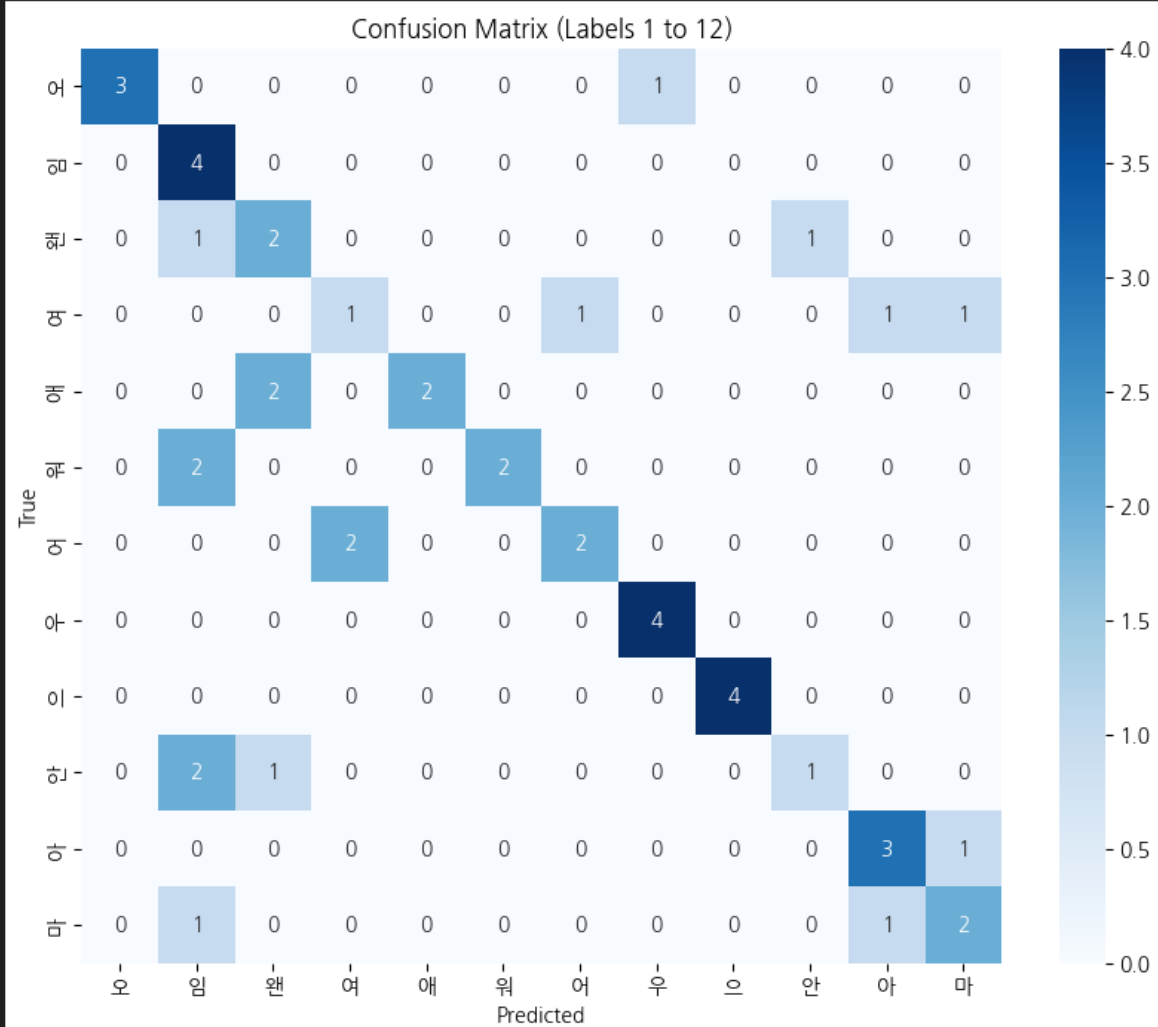
```
Epoch 1/100: 0%|          | 0/40 [00:00<?, ?it/s]
Epoch 1/100: 100%|████████| 40/40 [00:05<00:00, 6.79it/s]
Epoch [1/100], Train Loss: 2.3223, Train Accuracy: 20.70%, Val Loss: 1.8717, Val Accuracy: 45.57%
Epoch 2/100: 100%|████████| 40/40 [00:04<00:00, 8.11it/s]
Epoch [2/100], Train Loss: 1.6949, Train Accuracy: 49.68%, Val Loss: 1.3529, Val Accuracy: 51.90%
Epoch 3/100: 100%|████████| 40/40 [00:04<00:00, 8.11it/s]
Epoch [3/100], Train Loss: 1.2729, Train Accuracy: 56.37%, Val Loss: 1.1614, Val Accuracy: 49.37%
Epoch 4/100: 100%|████████| 40/40 [00:05<00:00, 7.92it/s]
Epoch [4/100], Train Loss: 1.0684, Train Accuracy: 63.69%, Val Loss: 1.1038, Val Accuracy: 55.70%
Epoch 5/100: 100%|████████| 40/40 [00:05<00:00, 7.98it/s]
Epoch [5/100], Train Loss: 0.8290, Train Accuracy: 75.48%, Val Loss: 0.9797, Val Accuracy: 58.23%
Epoch 6/100: 100%|████████| 40/40 [00:04<00:00, 8.02it/s]
Epoch [6/100], Train Loss: 0.6725, Train Accuracy: 79.94%, Val Loss: 0.8633, Val Accuracy: 62.03%
Epoch 7/100: 100%|████████| 40/40 [00:05<00:00, 7.97it/s]
Epoch [7/100], Train Loss: 0.6478, Train Accuracy: 78.66%, Val Loss: 0.9774, Val Accuracy: 63.29%
Epoch 8/100: 100%|████████| 40/40 [00:05<00:00, 7.97it/s]
Epoch [8/100], Train Loss: 0.5098, Train Accuracy: 88.22%, Val Loss: 0.8720, Val Accuracy: 62.03%
Epoch 9/100: 100%|████████| 40/40 [00:05<00:00, 7.99it/s]
Epoch [9/100], Train Loss: 0.3970, Train Accuracy: 89.17%, Val Loss: 0.8871, Val Accuracy: 68.35%
Epoch 10/100: 100%|████████| 40/40 [00:05<00:00, 7.92it/s]
Epoch [10/100], Train Loss: 0.3228, Train Accuracy: 92.04%, Val Loss: 1.0224, Val Accuracy: 64.56%
Epoch 11/100: 100%|████████| 40/40 [00:05<00:00, 7.96it/s]
Epoch [11/100], Train Loss: 0.3594, Train Accuracy: 91.08%, Val Loss: 0.8590, Val Accuracy: 63.29%
Epoch 12/100: 100%|████████| 40/40 [00:05<00:00, 7.85it/s]
Epoch [12/100], Train Loss: 0.2628, Train Accuracy: 93.95%, Val Loss: 0.9036, Val Accuracy: 64.56%
Epoch 13/100: 100%|████████| 40/40 [00:05<00:00, 7.96it/s]
Epoch [13/100], Train Loss: 0.2073, Train Accuracy: 96.50%, Val Loss: 0.9429, Val Accuracy: 70.89%
Epoch 14/100: 100%|████████| 40/40 [00:05<00:00, 7.87it/s]
Epoch [14/100], Train Loss: 0.2222, Train Accuracy: 96.50%, Val Loss: 1.0631, Val Accuracy: 68.35%
Epoch 15/100: 100%|████████| 40/40 [00:05<00:00, 7.91it/s]
Epoch [15/100], Train Loss: 0.1449, Train Accuracy: 96.50%, Val Loss: 0.9174, Val Accuracy: 67.09%
Epoch 16/100: 100%|████████| 40/40 [00:05<00:00, 7.95it/s]
Epoch [16/100], Train Loss: 0.1119, Train Accuracy: 98.73%, Val Loss: 1.1230, Val Accuracy: 65.82%
Early stopping
모델 학습 완료
```

⇒ 이전에 비해 데이터 셋이 월등히 늘어나 Validation Accuracy가 높아짐을 확인할 수 있다.

⇒ 실제로 설정해준 Epoch은 100이지만, Train Loss가 일정 수준에서 더 낮아지지 않을 경우 Train을 멈춤

- **Accuracy : 62.5%**

Evaluating: 100% | 12/12 [00:00<00:00, 36.76it/s]
Accuracy: 0.6250



⇒ 수정 사항들을 적용하기 전에는 Accuracy가 10퍼센트도 되지 않았는데, 수정 사항 적용을 통해 프로그램을 개선해 현재 62.5%의 정확도를 보임을 확인할 수 있다.

○ 앞으로 남은 과제

1. 데이터셋 추가 확보
2. 실제 구현 시나리오 만들기