
Deep Learning Project03

가구 이미지 데이터 분류 (CNN)

Kim Gyu San

김 규 산

2024년 6월 4일



목차

I. Introduction	3
II. 활용 방안.....	3
III. Workflow	4
III-1. Overview.....	4
III-2. Detailed Process	5
IV. 데이터 수집.....	5
IV-1. 가구 이미지 데이터 프레임 분류.	5
V. 유사도 분석.....	6
VI. CNN 사전 모델 훈련.....	6
VI-1. VGG16.	6
VI-2. Xception	7
VI. Fine tuning (미세조정).	7
VI-1. VGG16	7
VI. 모델 평가 및 검증.	8
VII. 결과 및 분석.....	10
VII-1. 유사도 분석을 통한 사전 훈련 모델 선택.....	10
VII-2. 훈련 속도의 최적화를 위한 Fine Tuning 진행.....	10
VII-3. 최종 분석.....	10

I. Introduction

- 이번 프로젝트의 목표는 가구 이미지를 CNN 모델로 훈련시켜, 실제 가구 이미지를 분석했을 때 정확하게 가구의 종류를 분류할 수 있는 딥러닝 모델을 구축하는 것입니다.
- 이 딥러닝 모델을 통해 가구 이미지를 분석하여 어떤 가구로 분류되어 있는지를 판별하는 것을 목표로 합니다.

II. 활용 방안.

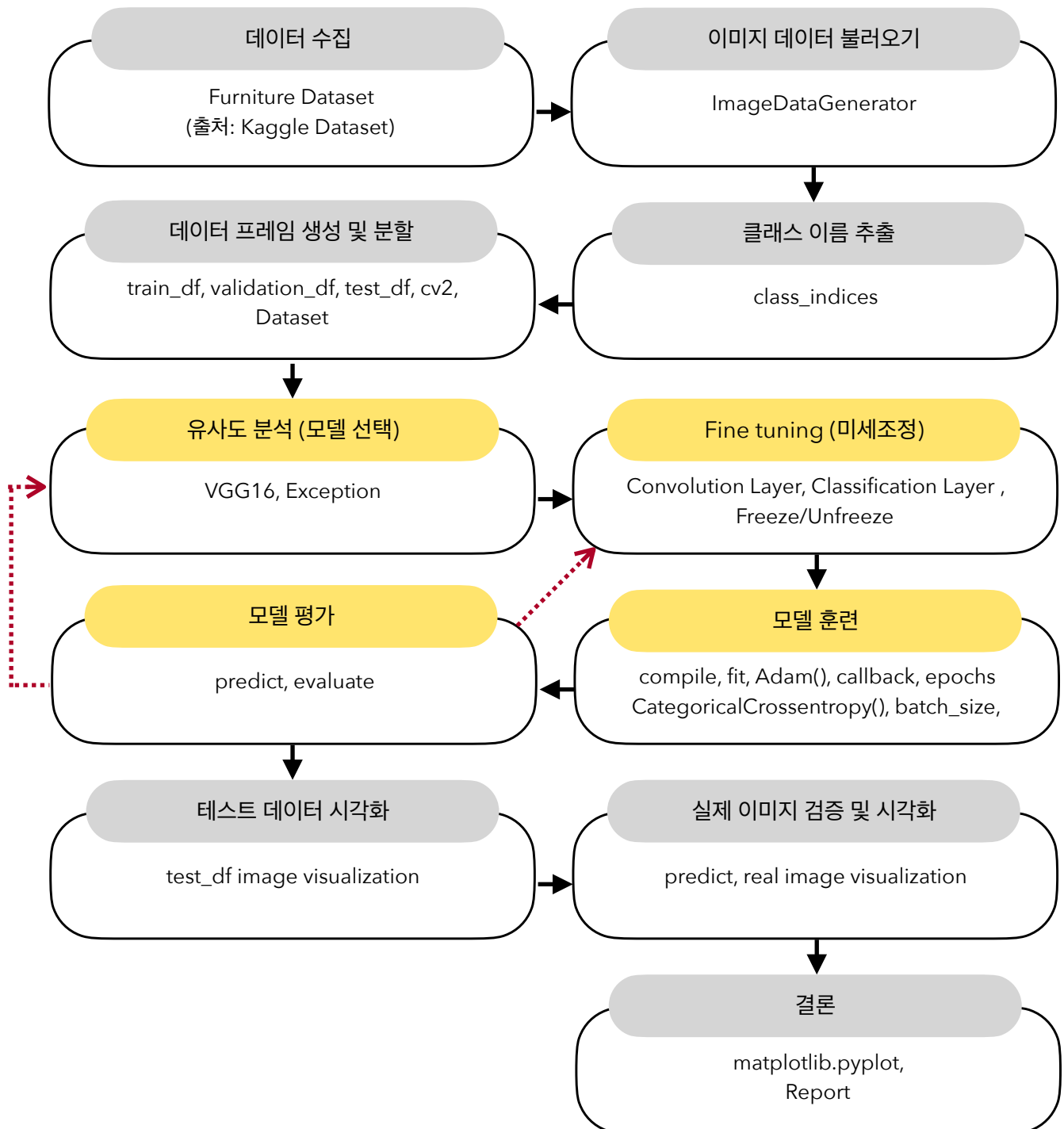
- AI가 가구 이미지를 판별하여 높은 분류 정확성을 보이도록 구성하여, 가구의 일부만 나타난 이미지에서도 정확하게 분류할 수 있을 것으로 기대됩니다.
- 또한, 가구의 형태에 따라 가격을 매기는 플랫폼에서, 사진만으로도 가구를 정확히 분류하여 가격을 자동으로 책정할 수 있어 효율성을 높일 수 있습니다.
- 가구 매장이나 온라인 쇼핑몰에서 고객이 사진을 업로드하면 해당 가구의 정보를 자동으로 제공하고 유사한 가구를 추천하는 기능을 구현할 수 있습니다.
- 인테리어 디자인 소프트웨어와 연계하여 가구 배치 시뮬레이션을 할 때, 가구의 종류와 특성을 자동으로 인식하여 배치의 편의성을 높일 수 있습니다.



* 출처: <https://m.edaily.co.kr/news/read?newsId=01882726629209968&mediaCodeNo=257>

III. Workflow

III-1. Overview



III-2. Detailed Process

단계	설명
0. 데이터 수집	- Furniture Dataset (출처: Kaggle Dataset)
1. 데이터 전처리	- ImageDataGenerator, class_indices를 사용한 이미지 불러오기.
	- train_df, validation_df, test_df 를 활용한 데이터 프레임 생성 및 분할.
	- cv2, cvtColor를 활용한 이미지 확인. - Dataset의 메소드 재정의
2. 유사도 분석	- Albumentations 라이브러리를 사용한 데이터 증강
3. 모델 선택	- VGG16, Exception
4. Fine tuning	- Convolution Layer, Classification Layer, Freeze/Unfreeze
5. 모델 훈련	- compile, fit, Adam, callback, epochs, CategoricalCrossentropy - batch_size
6. 모델 평가	- predict, evaluate, matplotlib.pyplot
7. 테스트 데이터 시각화	- predict, test_df image visualization
8. 실제 이미지 검증 및 시각화	- predict, real image visualization
9.. 결론	- Report

IV. 데이터 수집.

- Kaggle 홈페이지에서 일반적인 가구에 관련된 이미지를 수집 하였습니다.

IV-1. 가구 이미지 데이터 프레임 분류.

- Train_df, Validation_df, Test_df 3개의 데이터프레임으로 분류 하였습니다

	file_paths	target_names	targets	targets
0	./datasets/p_furniture/almirah_dataset/image_1...	almirah_dataset	0	2 1920
1	./datasets/p_furniture/almirah_dataset/image_1...	almirah_dataset	0	0 1920
2	./datasets/p_furniture/almirah_dataset/image_1...	almirah_dataset	0	1 1920
3	./datasets/p_furniture/almirah_dataset/image_1...	almirah_dataset	0	3 1920
4	./datasets/p_furniture/almirah_dataset/image_1...	almirah_dataset	0	4 1920
...	Name: count
14995	./datasets/p_furniture/tv dataset/image_995.jpeg	tv dataset	4	targets
14996	./datasets/p_furniture/tv dataset/image_996.jpeg	tv dataset	4	3 480
14997	./datasets/p_furniture/tv dataset/image_997.jpeg	tv dataset	4	0 480
14998	./datasets/p_furniture/tv dataset/image_998.jpeg	tv dataset	4	1 480
14999	./datasets/p_furniture/tv dataset/image_999.jpeg	tv dataset	4	4 480
				Name: count
				targets
				2 600
				4 600
				0 600
				3 600
				1 600
				Name: count

V. 유사도 분석.

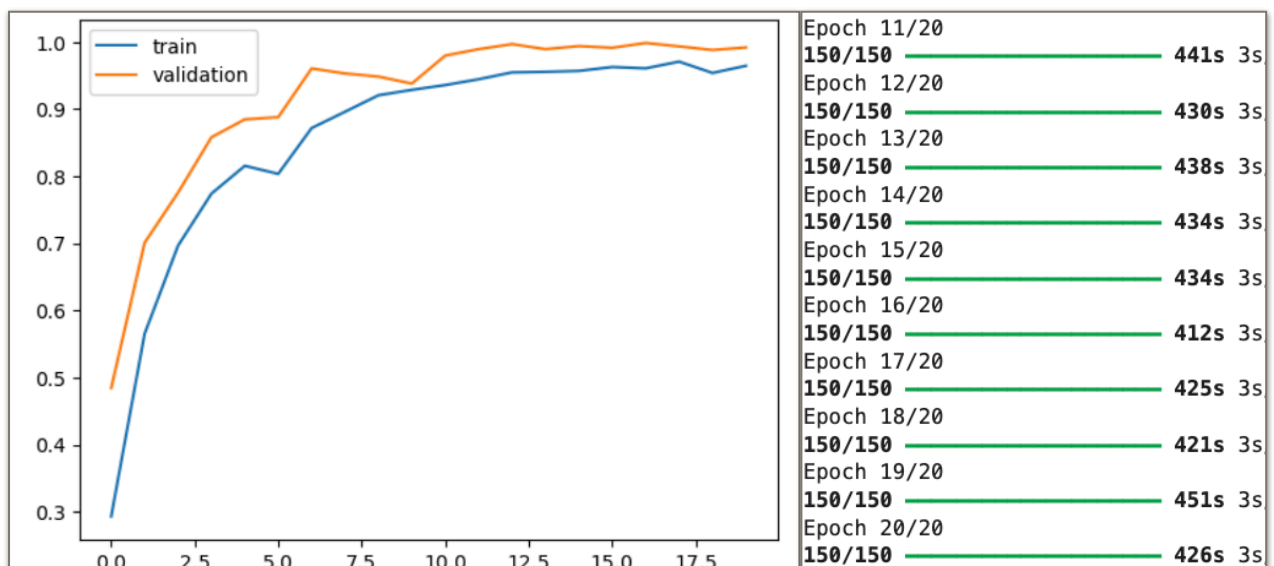
- VGG16, Xception 모델의 사전 훈련 모델에 대하여 시각화 및 유사도 분석을 진행하였습니다.
- VGG16 사전 모델에서 예측 정도가 높은 부분 타나났습니다.



VI. CNN 사전 모델 훈련.

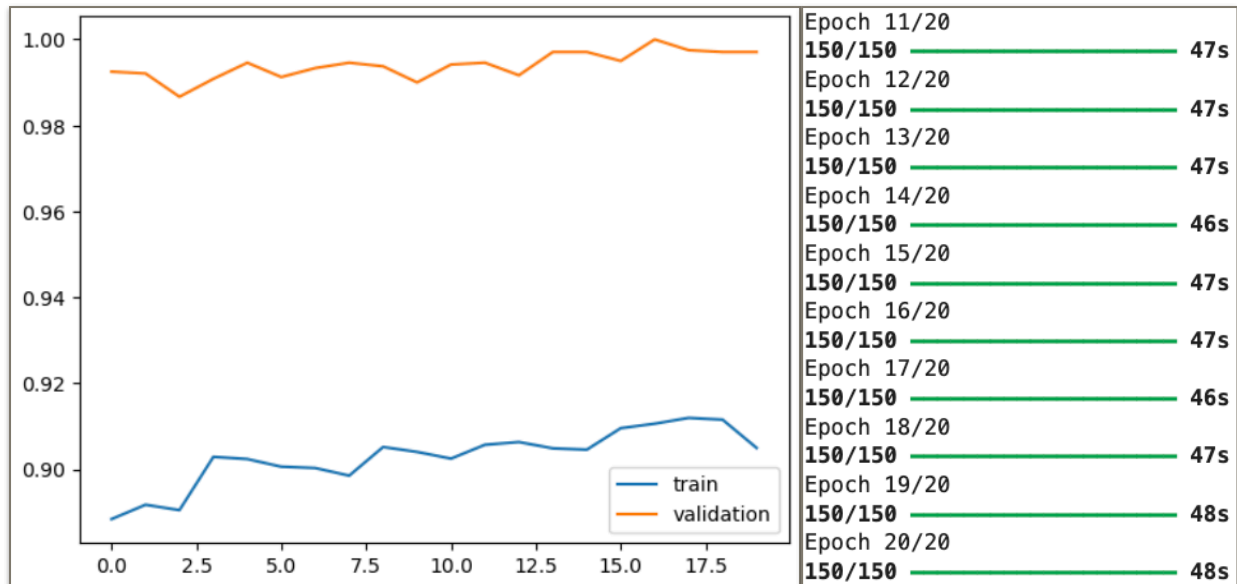
- VGG16 모델의 사전 훈련 속도를 측정하고 정확도를 검증하기 위해 훈련을 진행하였습니다.
- 훈련 속도에서 약 400s 이상의 훈련 시간을 확인할 수 있습니다.

VI-1. VGG16.



VI-2. Xception

- Xception 모델의 사전 훈련 속도를 측정하고 정확도를 검증하기 위해 훈련을 진행하였습니다.
- 훈련 속도에서 약 40s 이상의 훈련 시간을 확인하였지만 상대적으로 다른 모델에 비하여 상대적으로 훈련데이터의 정확도가 낮게 나타났습니다.

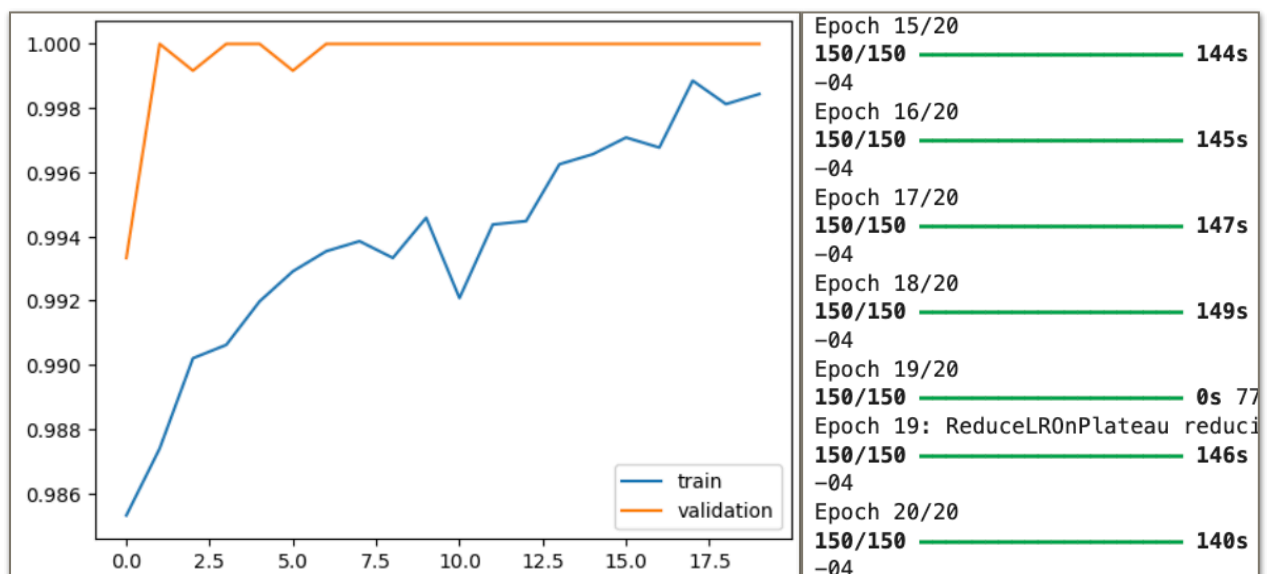


VI. Fine tuning (미세조정).

VI-1. VGG16

- 훈련속도의 효율을 높이기 위하여 fine tuning (미세조정) 을 이용하여 분류기 및 정확도를 검증하기 위해 훈련을 진행하였습니다.

1) Convolution Layer 전체 freeze. (훈련속도:130s)



VI. 모델 평가 및 검증.

VI-1. 테스트 데이터에 대하여 VGG16 모델 시각화.

- 8/8, 100%

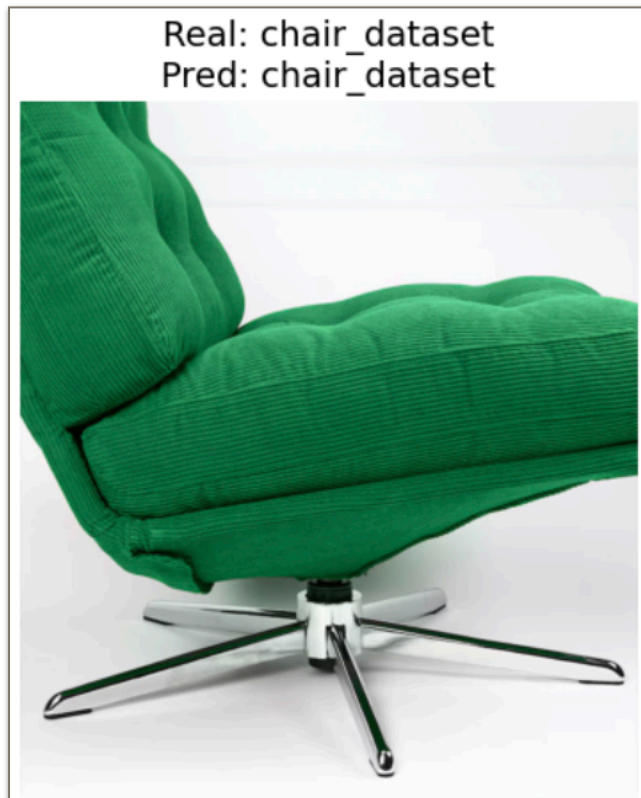


VI-2. 테스트 데이터에 대하여 Xception 모델 시각화.

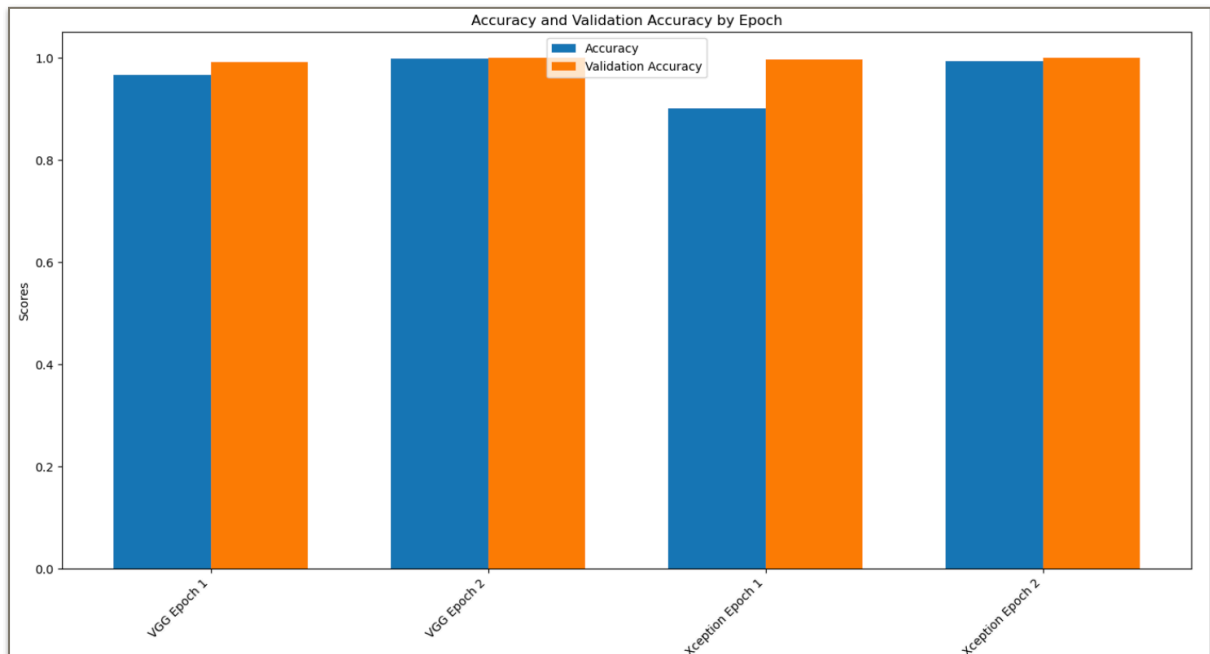
- 7/8, 87.5%



VI-4. 실제 이미지에 대하여 예측 결과 시각화.
- VGG16 모델 사용.



VI-5. Cycle에 따른 테스트 데이터 정확도와 검증 데이터 정확도 바 그래프.



VII. 결과 및 분석.

VII-1. 유사도 분석을 통한 사전 훈련 모델 선택

- CNN 사전 훈련 모델(VGG16, Xception)을 사용하여 테스트 데이터에 대해 유사도를 초기 검증하였습니다.
- VGG16 모델이 더 높은 유사도를 보여주었습니다.

VII-2. 훈련 속도의 최적화를 위한 Fine Tuning 진행.

- VGG16 모델의 훈련 속도를 약 800s -> 140s 줄이며 성능을 높게 유지하였습니다.
- Xception 모델은 분류기만 훈련하였을 때 훈련 속도는 빨랐지만, VGG16 모델보다 성능이 낮았습니다.
- 모델 특성에 따라 Convolution Layer 층을 높이는 것이 적절한 학습 시간과 성능을 얻는 데 중요합니다.

VII-3. 최종 분석.

- 유사도가 높은 VGG16 모델을 사용했을 때 높은 성능을 확인할 수 있었습니다.
- 훈련 속도의 효율성을 위해 Fine Tuning을 진행한 결과, 훈련 시간이 약 1/6로 줄어들었습니다.
- Xception 모델을 사용하여 훈련했을 때, 동일한 이미지 전처리 조건에서 훈련 시간이 더 짧았지만 정확도가 낮았습니다. 정확도를 높이기 위해서는 이미지 전처리 조건을 다르게 설정해야 했습니다.
- 유사도가 높은 모델을 사용하는 것이 최초 훈련에서도 높은 정확도를 가질 수 있다는 결론을 내렸습니다.