

딥러닝 프로젝트 4조

패션 이미지를 통한 스타일 분류 프로젝트

김은비, 김현욱, 김휘담, 정우진

Contents

1. 주제 선정 이유
2. 데이터 수집 및 소개
3. 모델 설명과 결과
 - 3-1. 스타일 분류 모델
 - 3-2. 클러스터링
 - 3-3. 모델 활용 예시
4. 기대 효과
5. 느낀 점

01

주제 선정 이유

1. 온라인 쇼핑의 급속한 확산

소비자가 예전에 눈여겨봤던 상품 등
원하는 상품을 찾기 어려워짐

2. 기존 텍스트 기반의 검색

사용자의 의도를 정확히 반영하지 못하는 한계



이미지 기반 검색

유사 상품, 의류 등을 빠르고 정확하게 찾는 방법 제공

여러가지 패션 스타일의 고유한 특징점을 찾아
기계 학습으로 분류하여 입력한 이미지의 스타일과,
해당 이미지와 유사한 패션을 찾아내어
개인화된 정보를 제공할 수 있습니다.

02

데이터 수집 및 소개

#K-Fashion 이미지 # 인공지능 # 패션 이미지 데이터셋 # 이미지 저작도구

K-Fashion 이미지

분야 영상이미지 유형 이미지

구축년도 : 2020 갱신년월 : 2021-09 조회수 : 30,688 다운로드 : 3,048 용량 : 63.70 GB

- 총 1,200,000건 분량의 K-Fashion 이미지 데이터셋
- 대분류 10가지, 스타일 23가지 레이블링 정보
- 패션 전문가 및 AI 전문가의 의견을 거친 세부속성 정의

스타일 카테고리 정보

트래디셔널	매니시	페미닌	에스닉	컨템포러리	내추럴	젠더플루이드	스포티	서브컬처	캐주얼
클래식	매니시	페미닌	히피	모던	컨트리	젠더리스	스포티	레트로	밀리터리
프레피	통보이	로맨틱	웨스턴	소피스트 케이티드	리조트			키치/ 키덜트	스트리트
		섹시	오리엔탈	아방가르드				힙합	
								펑크	

패션 이미지 데이터



03

모델 설명과 결과 / 3-1. 스타일 분류 모델

ResNet50 전이학습

1. 학습할 이미지 개수와 라벨(스타일) 선정
2. ImageDataGenerator 클래스를 활용하여 이미지 전처리
3. ImageNet 데이터셋으로 사전 훈련된 모델 (VGG16, ResNet50 등)을 로드
4. 사전 훈련된 가중치를 고정하고, 새로운 층만 학습되도록 설정
5. 기존 모델의 출력에 새로운 층을 추가, 분류 개수와 softmax 함수 지정 후 learning rate, epoch, batch_size를 하이퍼파라미터 튜닝

```
resnet = ResNet50(input_shape= (256,256,3), weights='imagenet')

for layer in resnet.layers:
    layer.trainable = False

x = resnet.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
predictions = Dense(5, activation='softmax')(x)
model = Model(resnet.input, predictions)

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate = 0.001),
              loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit(train, validation_data=[val], epochs=10, batch_size=32)
```

03

모델 설명과 결과 / 3-1. 스타일 분류 모델

1. 스타일 5개 (리조트, 모던, 스포티, 컨트리, 페미닌) 선정하여 총 30,000건의 이미지 학습

	VGG16	ResNet50	EfficientNet
Accuracy	0.4840	0.6104	0.6577
Loss	1.5874	1.7704	1.3441

→ 복잡한 모델을 쓸 때 성능의 변화가 있음을 관찰 (VGG16 → ResNet50)

03

모델 설명과 결과 / 3-1. 스타일 분류 모델

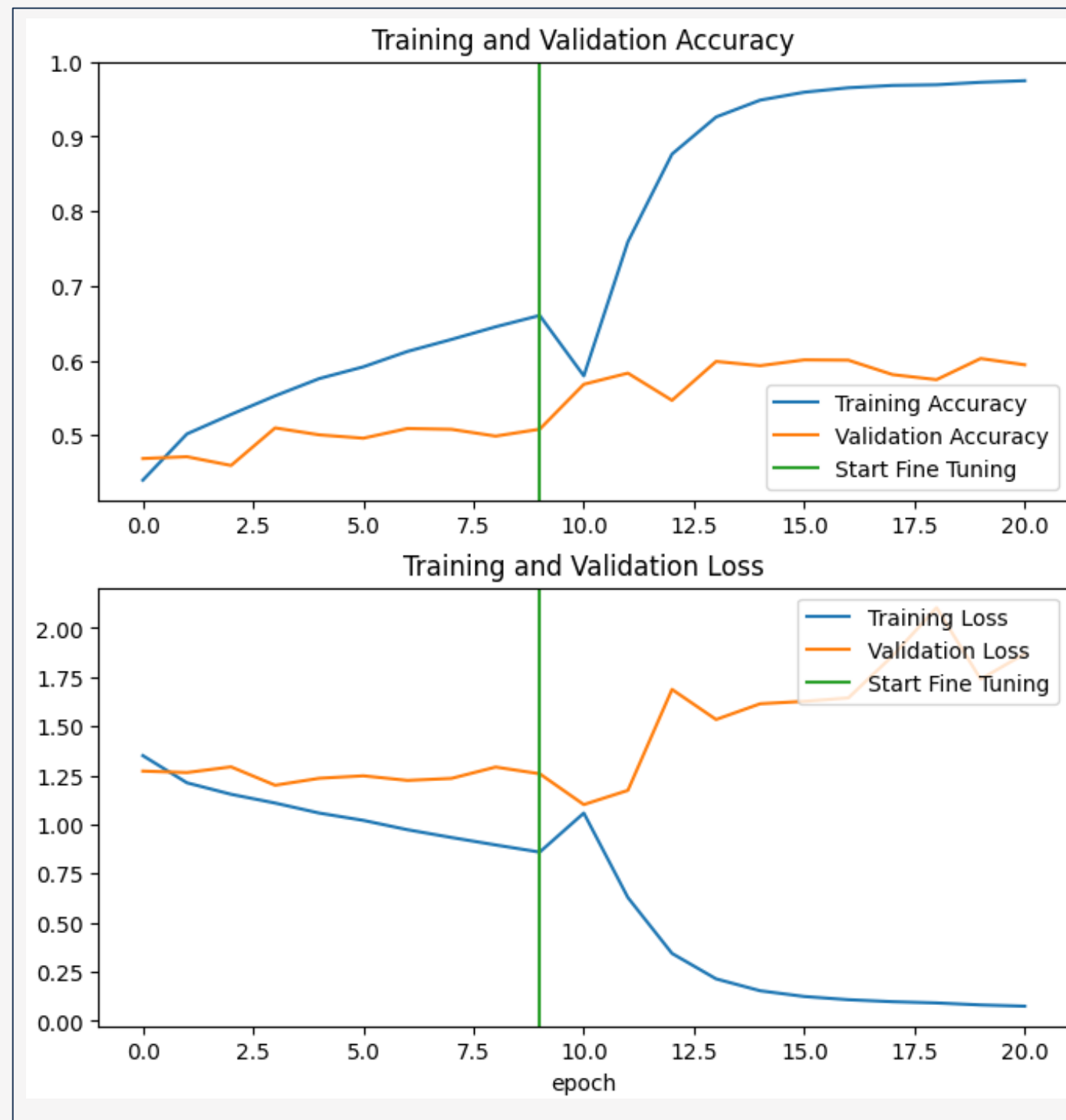
2. 스타일 21개 1000장씩 학습 - 모델: ResNet50

정확도(train)	정확도(vaild)	Batch size	epochs
30.37%	25.24%	32	40
21.39%	20.45%	16	25
35.83%	27.62%	64	25
48.38%	31.36%	128	25

→ batch를 높임에 따라 정확도(train)가 높아지지만, 정확도(vaild)의 값이 낮은 현상

03

모델 설명과 결과 / 3-1. 스타일 분류 모델



성능 향상을 위해 일부 레이어를
학습 가능하게 하여 추가적으로 학습

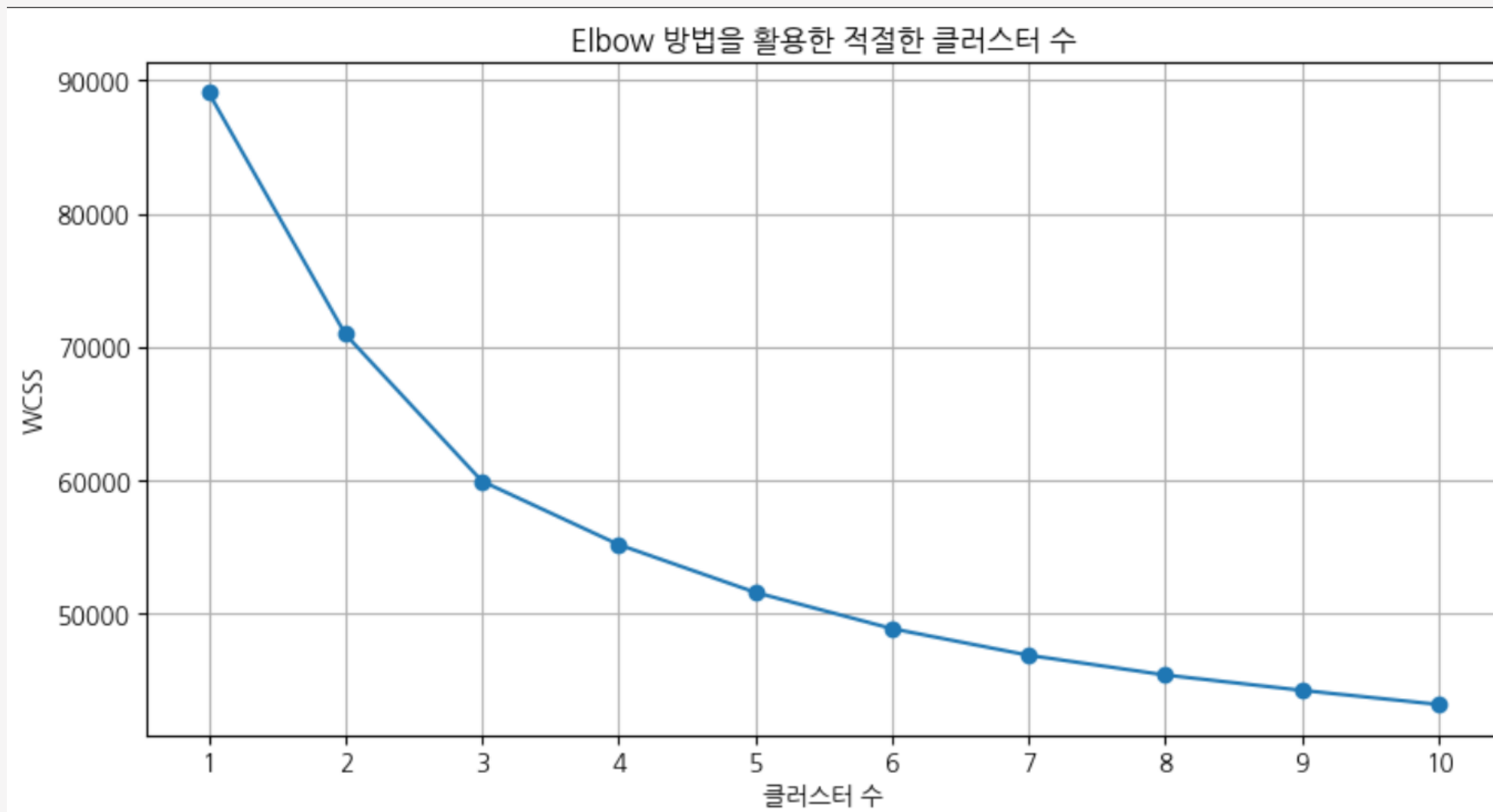
미세조정 후 accuracy가 향상되었지만
Train 데이터에 과적합, loss 증가 등의 문제 발생

→ 패션 데이터를 스타일별로 분류하는 데에 한계를 발견

→ “클러스터링”을 통해 이미지들이 가진 특징을 파악할 수 있을지
다시 분석을 시도하였습니다.

03

모델 설명과 결과 / 3-2. 클러스터링



원본 데이터의

총 23가지 스타일 클래스 모두 선별하여

1000장씩 랜덤 샘플링

→ 총 23000장으로 k-means 클러스터링

클러스터 수는 3개가 적합하다고 판단

03

모델 설명과 결과 / 3-2. 클러스터링

<Cluster 0>



클러스터 특징: 여성스러운 스타일, 블라우스/레이스 무늬 계열이 돋보임

스타일 분포는 로맨틱, 리조트, 컨트리 순으로 많음

03

모델 설명과 결과 / 3-2. 클러스터링

<Cluster 1>



클러스터 특징: 중성적이거나 차분하며 모던한 스타일

스타일 분포는 밀리터리, 매니시, 젠더리스 순으로 많음

03

모델 설명과 결과 / 3-2. 클러스터링

<Cluster 2>



클러스터 특징: 개성이 돋보이거나, 서브컬처적인 스타일

스타일 분포는 기타, 키치, 힙합, 프레피 순으로 많음

03

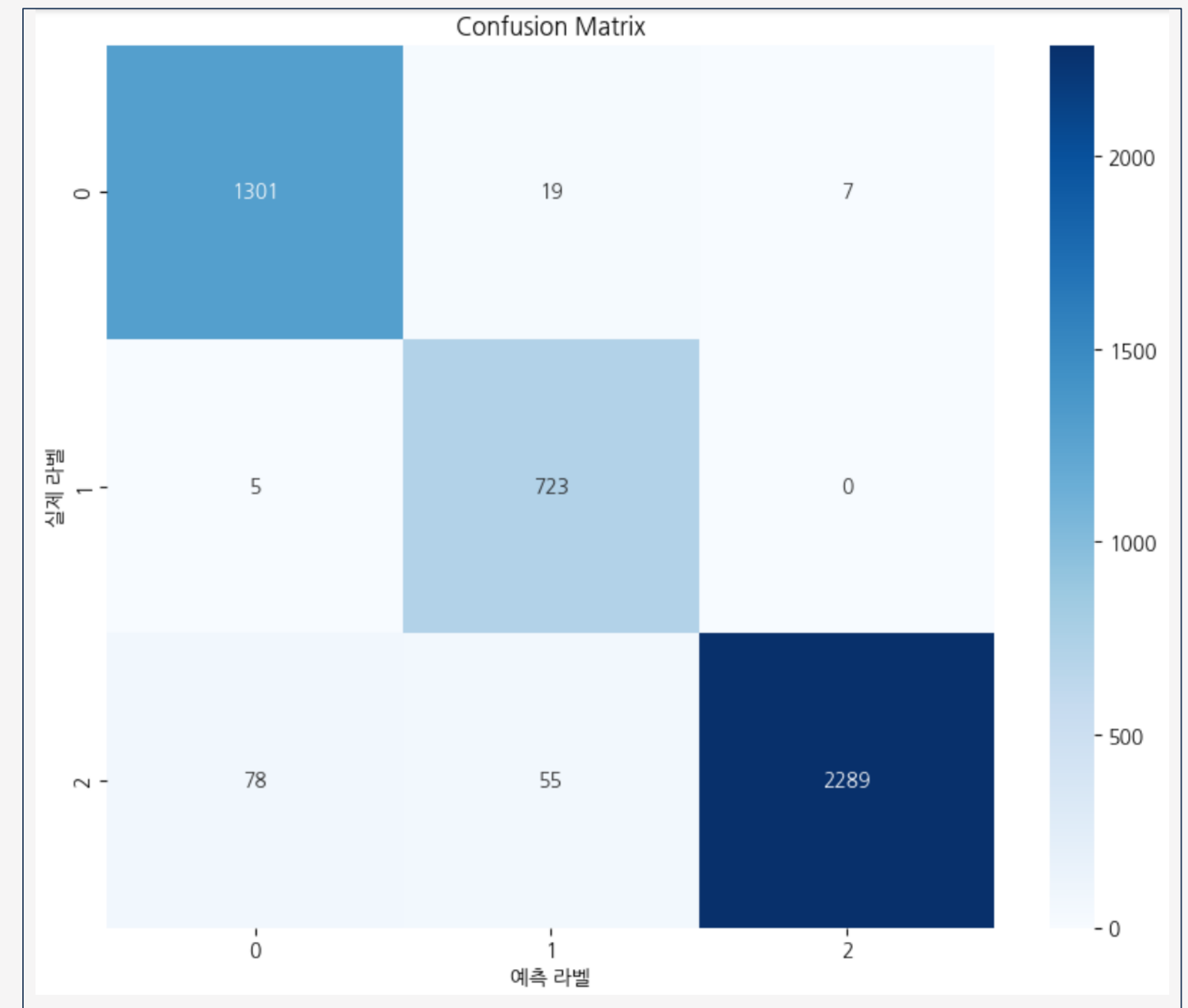
모델 설명과 결과 / 3-2. 클러스터링

Classification Report				
	precision	recall	f1-score	support
0	0.94	0.98	0.96	1327
1	0.91	0.99	0.95	728
2	1.00	0.95	0.97	2422
accuracy			0.96	4477
macro avg	0.95	0.97	0.96	4477
weighted avg	0.97	0.96	0.96	4477

각 클러스터를 라벨로 다시 모델링 진행

ResNet50 전이학습

Test Loss: 0.0608 / Test Accuracy: 0.9734



03

모델 설명과 결과 / 3-3. 모델 활용 예시

2. 스타일 예측

```
1/1 [=====] - Us 27ms/step
np.argmax(pred[0])          1
index[np.argmax(pred[0])]    cluster 1
pred[0][np.argmax(pred[0])] 0.9999984502792358
```

3. 모델이 예측한 스타일 내의 유사한 이미지 검색

image



1. 검색 이미지 준비



04

기대효과



사용자 관점

고객이 특정 의류 사진을 업로드하면 시스템이 해당 스타일을 분석하여 유사한 스타일의 의류를 추천

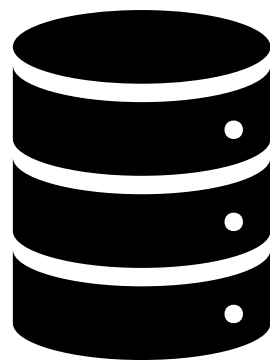
→ 더 빠르고 편리하게 원하는 제품을 발견



비즈니스 관점

쇼핑몰은 다양한 의류의 스타일을 자동으로 분류할 수 있고, 최신 트렌드인 스타일의 의류들을 식별하여 고객의 선호도에 따라 유사 의류들을 추천 가능

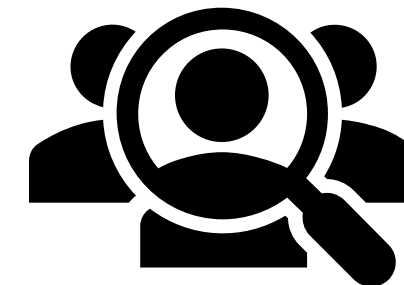
→ 고객의 쇼핑 만족도 개선, 쇼핑몰의 판매 기회를 확대



데이터 양이 많아서 모델 학습
시간이 오래 걸림
→ 스타일 라벨을 줄이고,
이미지 랜덤 샘플링 등 여러
시도를 거침



패션 이미지의 특성상 스타일을
정확하게 구별해내기가 어려움
→ 정해진 스타일 라벨 외에
클러스터링을 통한 그룹화 후
분류 모델 학습



각 클러스터별 특징을
파악하는 과정
→ 스타일을 한눈에 알아차리기
쉽지 않은 이미지도 존재

05

느낀 점

실제 라벨: 0
예측 라벨: 2



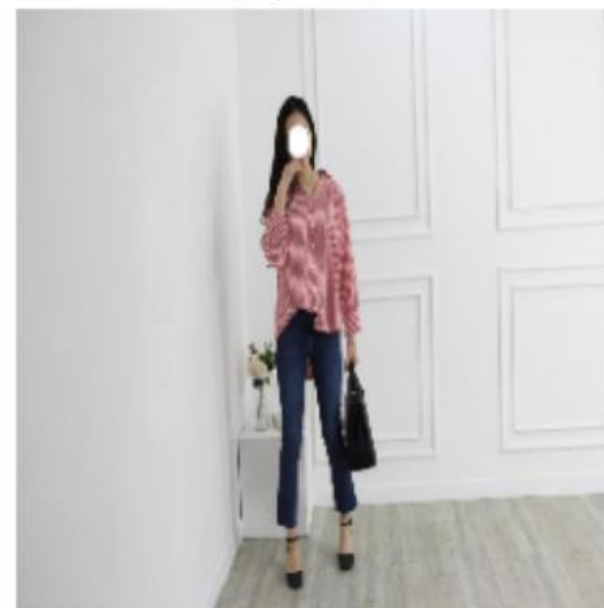
실제 라벨: 0
예측 라벨: 1



실제 라벨: 0
예측 라벨: 2



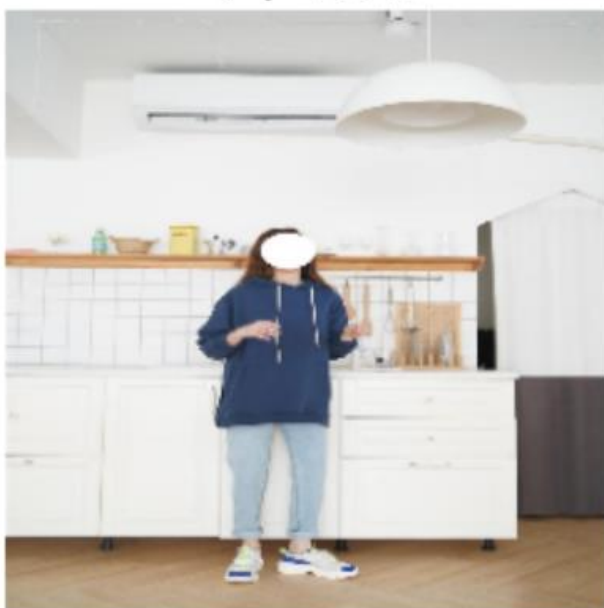
실제 라벨: 0
예측 라벨: 1



실제 라벨: 0
예측 라벨: 1



실제 라벨: 0
예측 라벨: 2



실제 라벨: 0
예측 라벨: 2



실제 라벨: 0
예측 라벨: 1



실제 라벨: 0
예측 라벨: 1



실제 라벨: 0
예측 라벨: 2



감사합니다
