

Fashion-based Classification of Koreans and Non-Koreans using CNN

이름: 김서영

학과: 인공지능융합전공

Abstract—나라별로 고유의 패션 문화를 갖는다. 따라서, 패션 이미지를 기반으로 한국인과 비한국인을 분류하고자 한다. 이미지 기반 이진 분류이므로 CNN을 활용하여 모델을 구축하였고, 97%의 정확도를 보였다. 한국인 데이터셋의 특이성, 그리고 한국인 데이터셋과 비한국인 데이터셋의 배경의 차이 등으로 과적합이 발생하였다. 보다 보편성을 갖는 사진들로 다양한 민족의 패션을 담은 데이터셋을 구성한다면 한국인과 비한국인의 분류를 넘어 다양한 민족을 구분하는 데에 도움이 될 것이다.

[Keywords—CNN, Classification, Computer Vision, K-fashion, Korean]

I. INTRODUCTION

김성복의 연구에 따르면, 한 시대의 지배적인 이상이 인간의 마음속에 확립되고 생활 속에 뿌리 내려 의복으로 표현되고 구체화된다고 하였다. [1] 음정선의 연구에서는 패션 스타일링은 이미지를 연출하는 것으로서 아이템, 컬러, 소재, 문양, 액세서리, 헤어, 메이크업, 모델 등 각각의 요소들이 상호작용을 하여 만들어지는 것이라고 하였다. 더불어, 한국적 패션디자인의 조형특성을 알아보기 위해 형태, 색채, 재질, 문양, 디테일, 트리밍으로 분류하였다. [2] 형태, 색채, 문양, 디테일, 트리밍 등의 특성은 표면적으로 나타난다. 따라서, 패션 이미지를 통해 한국인과 비한국인을 분류하고자 한다.

II. METHOD

먼저, 데이터는 Kaggle 과 AIHub 에서 수집하였다. Kaggle 에서는 비한국인 패션을 카테고리별로 구분하여 이미지 url 주소를 나타낸 csv 파일을 다운로드하였고, AIHub 에서는 한국인의 패션을 카테고리로 분류한 이미지 데이터를 다운로드하였다. 비한국인 데이터셋은 csv 형태로 url 로 나타나 있었기에 PIL(Python Imaging Library) 라이브러리를 사용하여 이미지를 png 형태로 다운로드하였다. 한국인 데이터셋은 jpg 형태의 원천데이터와 json 형태의 라벨링데이터로 구성되어 있는데, 그 중 원천데이터를 사용하였다. 비한국인 데이터셋은 url 을 png 로 변환하여

총 15,702 장의 이미지를 수집하였고, 한국인 데이터셋은 레트로 6,833 장, 로맨틱 73,173 장, 리조트 83,181 장, 매니시 8,488 장, 모던 79,947 장, 밀리터리 3,633 장, 섹시 6,322 장, 소피스티케이티드 32,626 장, 스트리트 449,494 장, 스포티 14,701 장, 아방가르드 3,671 장, 오리엔탈 4,083 장, 웨스턴 1,712 장, 젠더리스 12,386 장, 컨트리 32,365 장, 클래식 39,411 장, 키치 4,840 장, 톰보이 7,578 장, 핑크 382 장, 페미닌 88,325 장, 프레피 2,471 장, 히피 6,544 장, 힙합 1,240 장, 기타 4,400, 총 967,806 장의 이미지를 수집하였다. 이후, 비한국인 데이터셋과 한국인 데이터셋의 크기를 균일하게 하기 위해서 한국인 데이터 중 일부를 랜덤 추출하는 전처리 과정을 거쳤다. 랜덤 추출은 각 폴더별로 진행하여 카테고리별 비율을 유지하였다. 따라서, 비한국인 이미지 데이터 15,702 장, 한국인 이미지 데이터 14,456 장으로 최종 데이터셋을 구성하였다.

Train Data, Validation Data, Test Data 의 비율은 64:16:20 으로 설정하였다.

본 연구의 데이터의 형태는 이미지이고, 본 연구의 목적은 분류이기에 CNN 모델을 사용하였다. 그 중, 비한국인과 한국인을 구분하는 이진 분류이기에 CNN 의 Hidden Layer 의 activation function 으로는 ReLU 를, Output Layer 의 activation function 으로는 Sigmoid 를 사용하였고, Optimizer 는 Adam, loss function 은 Cross Entropy 를 사용하였다. Batch size 는 256, learning rate 는 0.001 로 설정하였고, epoch 는 10 으로 설정하였다. 자세한 모델 구조는 [그림 1] ~ [그림 4]과 같다.

```
ConvolutionalNeuralNet(
  (blocks): Sequential(
    (0): ConvolutionBlock(
      (layers): Sequential(
        (0): Conv2d(3, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): ReLU()
        (2): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): Conv2d(8, 8, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        (4): ReLU()
        (5): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): ConvolutionBlock(
      (layers): Sequential(
        (0): Conv2d(8, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): ReLU()
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): Conv2d(16, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        (4): ReLU()
        (5): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (2): ConvolutionBlock(
      (layers): Sequential(
        (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): ReLU()
        (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
        (4): ReLU()
        (5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
)
```

[그림 1] 모델 구조

```

(3): ConvolutionBlock(
  (layers): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(4): ConvolutionBlock(
  (layers): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(5): ConvolutionBlock(
  (layers): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)

```

[그림 2] 모델 구조

```

(6): ConvolutionBlock(
  (layers): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(7): ConvolutionBlock(
  (layers): Sequential(
    (0): Conv2d(512, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (4): ReLU()
    (5): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
)
(layers): Sequential(
  (0): Linear(in_features=1024, out_features=50, bias=True)
  (1): ReLU()
  (2): BatchNorm1d(50, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): Linear(in_features=50, out_features=2, bias=True)
  (4): Sigmoid()
)
)

```

[그림 3] 모델 구조

```
Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  capturable: False
  differentiable: False
  eps: 1e-08
  foreach: None
  fused: None
  lr: 0.001
  maximize: False
  weight_decay: 0
)
CrossEntropyLoss()
```

[그림 4] 모델 구조

III. RESULT

모델 학습 결과는 [그림5]와 같다.

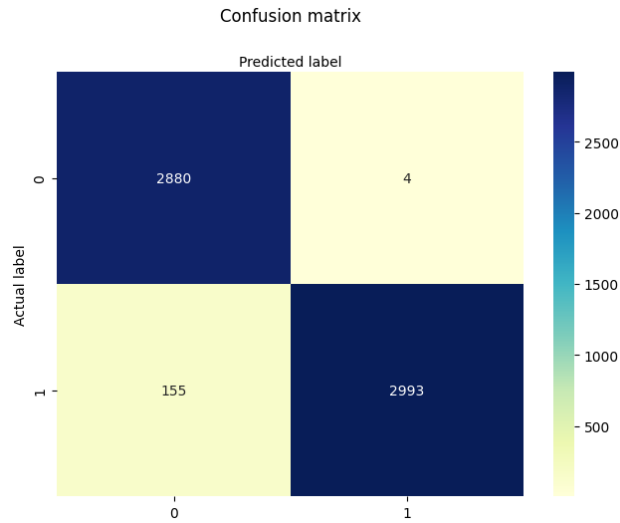
Epoch(1/10):	train_loss=3.7924e-01	valid_loss=3.5570e-01	lowest_loss=3.5570e-01
Epoch(2/10):	train_loss=3.4700e-01	valid_loss=3.4233e-01	lowest_loss=3.4233e-01
Epoch(3/10):	train_loss=3.4281e-01	valid_loss=4.2380e-01	lowest_loss=3.4233e-01
Epoch(4/10):	train_loss=3.4715e-01	valid_loss=3.4495e-01	lowest_loss=3.4233e-01
Epoch(5/10):	train_loss=3.4107e-01	valid_loss=3.4003e-01	lowest_loss=3.4003e-01
Epoch(6/10):	train_loss=3.3826e-01	valid_loss=3.3804e-01	lowest_loss=3.3804e-01
Epoch(7/10):	train_loss=3.4047e-01	valid_loss=5.7999e-01	lowest_loss=3.3804e-01
Epoch(8/10):	train_loss=3.3917e-01	valid_loss=3.6179e-01	lowest_loss=3.3804e-01
Epoch(9/10):	train_loss=3.3866e-01	valid_loss=3.4265e-01	lowest_loss=3.3804e-01
Epoch(10/10):	train_loss=3.3909e-01	valid_loss=3.5660e-01	lowest_loss=3.3804e-01

[그림 5] 모델 학습 결과

모델 평가 결과는 [그림6]과 같으며, 97%의 정확도, 99%의 정밀도, 95%의 재현율, 97%의 F1 스코어를 보였다. Confusion Matrix 결과는 [그림7]과 같다.

```
# of test_dataset : 6032
Accuracy: 0.9736
Accuracy: 0.9736405835543767
Precision: 0.9986653319986654
Recall: 0.9507623888182973
F1score: 0.974125305126119
```

[그림 6] 모델 평가 결과



[그림 7] Confusion matrix

IV. CONCLUSION

특징을 추출하여 분류하는 CNN을 활용한 한국인과 비한국인 분류라는 점에서 한국인 패션의 특성을 연구하는 의상학 연구에 사용할 수 있다. 더 나아가, 패션 유통 업계에서는 각 나라별 주력 상품을 선별하는 데에 사용할 수 있다. 또한, 이미지를 통한 한국인과 비한국인 분류라는 점에서, 한국인들이 외국에서 언어 등의 이유로 한국인을 찾아야 하는 상황에서 한국인을 선별하는 데에 도움을 받을 수 있다. 본 연구에서는 한국인과 비한국인으로 이진분류 모델을 구축하였지만, 한국인, 중국인, 일본인 등으로 분류를 더 세분화하면 한국인뿐만 아니라 세계 각국의 사람들에게도 도움이 될 것이다.

본 연구에서 구축한 모델은 97%의 상당히 높은 정확도를 보였지만 이는 과적합으로, 보완할 필요가 있다. 과적합을 보이는 이유는 데이터의 한계점 때문이다. 먼저, [그림8]과 같이 비한국인 이미지 데이터셋은 실외에서 촬영한 사진이 대다수이지만, [그림9]와 같이 한국인 이미지 데이터셋은 배경이 화려하지 않은 실내 스튜디오에서 촬영한 사진이 대다수이다. 따라서 한국인과 비한국인의 구분에 ‘패션’외에 이미지의 배경 등이 영향을 미칠 수 있다. 또한, [그림10]과 같이 한국인 데이터셋 중 일부 이미지에는 측면에 라벨링이 포함되어 있는 경우가 있었다. 따라서 데이터셋이 패션 이미지만을 온전히 담지 못했고, 이것이 모델의 성능에 영향을 주었을 수도 있다. 더불어, 한국인 데이터의 원천데이터가 하나의 코디에 대한 여러가지 자세를 각각의 이미지로 구성되었기에 패션이 겹치는 이미지셋이 존재했다. 보다 다양한 패션으로 데이터셋을 구성하지 못한 것이 데이터셋의 한계점이다.



[그림 8] Non-Korean dataset image



[그림 9] Korean dataset image



27206.jpg



27215.jpg

[그림 8] Korean dataset construction

V. REFERENCES

[1] S. B. Kim., "Beyond Orientation - the National Image and Identity of Korean Fashion.", *video culture*, no.8, pp.17-30, 2003.

[2] J. S. Eum., "A Study on the Trend and Expression of Korean Fashion Styling in Recent Times", *Asian Journal of National Anthropomorphology*, no.11, pp.143-156, 2012.