

컴퓨터 비전 REPORT



정보통신공학과 12201853 김가현
정보통신공학과 12192152 허영은

제출일자 2022.12.09

1. Project 개요

1) 계획 동기 및 목표

현대 사회로 가면서 유행하는 외모를 따라가는 것이 아닌, 본인의 개성을 찾아 장점을 살리고 약점을 보완하는 방식의 미가 중요시되고 있다. 이러한 사회적 모습에 따라 '퍼스널 컬러 (personal color)' 진단이 유행하여 본인에게 맞는 스타일링 및 색채를 찾아가는 분위기로 이어지고 있다.

하지만 퍼스널 컬러 진단은 1회 평균 10만원 정도로 비용이 매우 많이 들고, 주관적인 평가가 많이 들어가는 만큼 진단해주시는 분마다 판단해주시는 퍼스널 컬러가 다르다. 따라서 퍼스널 컬러를 자가 진단할 수 있는 모델을 구축해보고자 위 프로젝트를 기획하게 되었다.

프로젝트 목표는 퍼스널 컬러 진단 기준에서 가장 크게 영향을 미치는 "피부색"에 따라 퍼스널 컬러를 결정하도록 하며, 정성 평가로 결정되는 퍼스널 컬러를 값으로 정량화하여 평가하는 퍼스널 컬러 진단 모델을 구축하는 것으로 설정하였다.

2) 이론적 배경

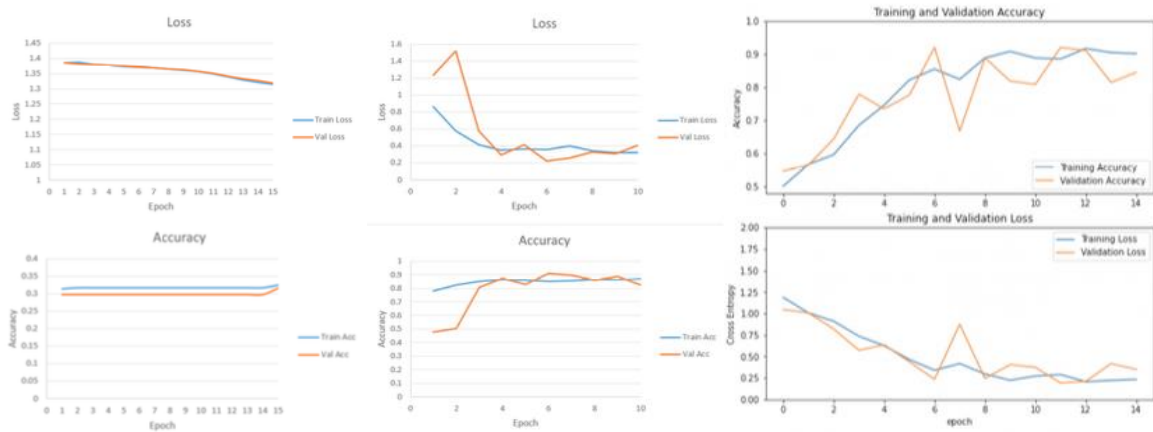
퍼스널 컬러 (personal color)란, 사람의 피부톤과 가장 잘 어울리는 색상을 찾는 색채학 이론이다. 사람의 피부톤에 따라 크게 웜(warm), 쿨(cool)로 분류되며, 이 안에서 아래 사진과 같이 봄웜(spring warm), 가을웜(autumn warm), 여름쿨(summer cool), 겨울쿨(winter cool) 4가지로 나뉜다. 더 세부적으로 나누면 12가지 혹은 이상이 나올 수 있지만, 프로젝트에서는 대표적인 4가지 퍼스널 컬러로만 분류하기로 결정하였다. 실제로는 피부톤 뿐만아니라 머리카락 색, 눈동자 색도 퍼스널 컬러 진단에 영향을 줄 수 있다.

Warm (웜)	Spring Warm (봄웜)	옐로우 베이스 톤 + 약간의 복숭아 빛		
	Autumn Warm (가을웜)	옐로우 베이스 톤 + 핑크색이 적음	봄웜 : 송혜교	가을웜 : 전지현
Cool (쿨)	Summer Cool (여름쿨)	회색끼 베이스 톤 + 약간의 복숭아 빛		
	Winter Cool (겨울쿨)	회색끼 베이스 톤 + 핑크색이 적음	여름쿨 : 정유미	겨울쿨 : 지수

2. Dataset 구축

1) 데이터 수집 및 전처리

가장 먼저, 각 퍼스널 컬러에서 대표되는 유명인 얼굴 이미지를 크롤링했다. 'Google Images Download' Library를 이용했는데, 이는 'Google'에 검색어를 입력했을 때, "이미지" 카테고리에 나오는 이미지가 정렬된 순서대로 다운로드되는 Library이다. 이때 우리의 목적은 정확한 피부색을 추출하는 것이기 때문에 색보정이 안되어있으며, 정면을 보고 있는 사진을 구해야했다. 따라서 어떠한 검색어로 이미지를 크롤링해야할까 고민한 결과, 기사 사진이 가장 적절하다고 생각되어 '000 기사사진'으로 이미지 크롤링을 진행했다. 예를 들어 봄웜으로 대표되는 연예인인 배우 '송혜교'의 얼굴 이미지를 크롤링하고자 하면, '송혜교 기사사진'을 검색해서 나오는 이미지의 상위 100개를 크롤링 및 분류하였다. 이렇게 유명인별로 100장씩 크롤링하였고, 퍼스널 컬러 별로 약 1700장, 총 7000장 이상 크롤링 및 퍼스널 컬러별로 분류하는 작업을 진행했다.



다음 그래프는 AlexNet, ResNet34, simple CNN 순으로 loss와 accuracy의 성능을 나타낸 그래프이다.

색 학습에 자주 쓰이는 alexnet이 가장 성능이 좋을 것이라 예상했지만 가장 성능이 좋지 않았다. 성능은 간단한 CNN 모델, ResNet34, AlexNet 순으로 좋았다. 이는 눈의 모양이나 얼굴 형태가 아닌 피부색만 존재하는 데 이터셋을 사용하다보니 무거운 모델의 과도한 은닉층 수와 뉴런의 수가 학습에 방해되었다고 판단하였다.

따라서 모델은 간단한 CNN 모델을 사용하기로 하였고, 초기 모델의 summary는 다음과 같다.

```
model = keras.Sequential()
model.add(keras.layers.Conv2D(32, 3, activation='relu', input_shape=(128,128,3)))
model.add(keras.layers.MaxPooling2D(2))
model.add(keras.layers.Conv2D(32, 3, activation='relu'))
model.add(keras.layers.MaxPooling2D(2))
model.add(keras.layers.Conv2D(32, 3, activation='relu'))
model.add(keras.layers.MaxPooling2D(2))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(128, activation='relu'))
model.add(keras.layers.Dense(4, activation='softmax'))
```

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_13 (Conv2D)	(None, 61, 61, 32)	9248
max_pooling2d_13 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_14 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_14 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten_4 (Flatten)	(None, 6272)	0
dense_8 (Dense)	(None, 128)	802944
dense_9 (Dense)	(None, 4)	516

=====
 Total params: 822,852
 Trainable params: 822,852
 Non-trainable params: 0

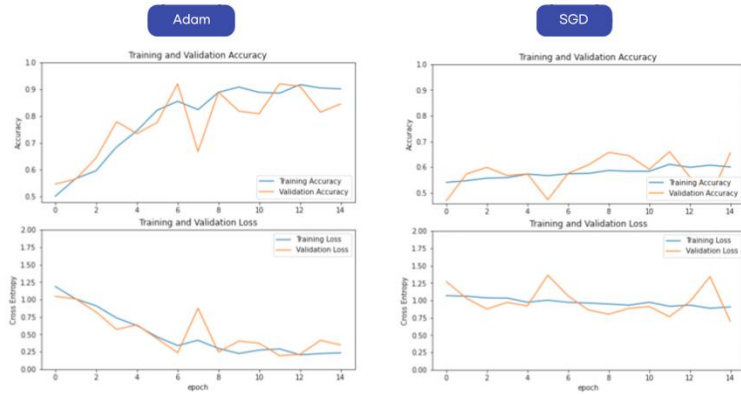
optimizer	adam (lr=0.001)
loss function	categorical_crossentropy
batch size	64
activation function	relu + softmax
epochs	15

모델의 성능은 validation accuracy는 0.84, validation loss는 0.3정도로 측정되었다.

2) Model 수정 과정

a. Optimizer

첫째로, 기존의 adam과 sgd를 비교해보았다.



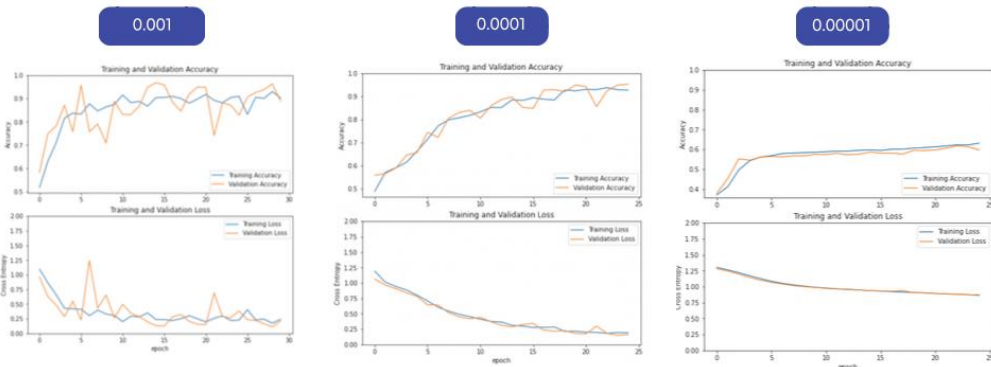
그래프를 보면 sgd는 성능이 좋지 않고, adam은 loss 값이 잘 떨어지고 높은 정확도를 보이고 있다. 따라서 같은 조건일 때 모델의 성능을 가장 좋게 만드는 Optimizer는 Adam이라고 판단하였다.

b. Data Augmentation

dataset을 부풀리고, 클래스 불균형 문제를 해소하고자 horizontal flips 기법을 사용해보았다. 예상했던 바와 같이 학습에 사용한 데이터가 피부색만 존재하는 데이터이기 때문에 뒤집어도 같아 초기 모델과 큰 성능 차이는 없었다. 따라서 Data Augmentation은 사용하지 않기로 하였다.

c. Learning rate

learning rate 값을 조절해보기 전에 epochs도 기존 15에서 25, 30으로 늘렸고, batch size는 64에서 32로 줄였다. batch size를 줄인 이유는 batch size가 크면 한 번에 많은 데이터로 학습하기 때문에 batch size를 줄이면 학습 그래프가 더 잘 수렴할 것 이라고 판단하였기 때문이다. epochs를 늘린 이유는 학습 그래프를 더 길게 보아 수렴하는지 확인해보기 위함이었다. batch size를 줄였기 때문에 learning rate를 잘 조절하여 local optimum에 빠지지 않도록 하는 것이 중요하다고 판단하였다.

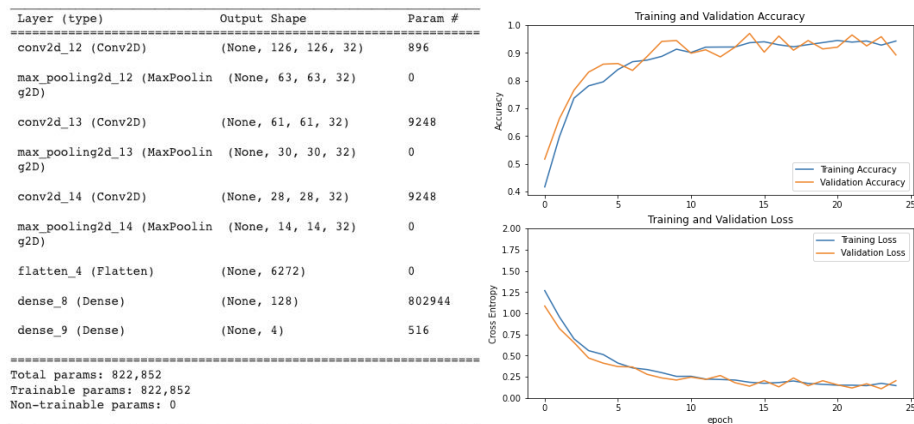


그래프를 보면 learning rate가 0.001인 경우에는 오버핏이 많이 보이는데, 0.0001의 경우에는 오버핏도 줄고 train과 valid의 차이가 크지 않아 overfitting이 되지 않는 모습을 관찰할 수 있다. 더 낮은 learning rate를 사용해본 결과 오버핏은 발생하지 않지만, 학습이 상당히 느리고 잘 수렴하지 않는 모습을 보이고 있었습니다. 따라서 learning rate는 0.0001을 사용하기로 하였다.

d. Dropout

0.5의 Dropout을 적용하니 중간중간 오버핏이 나타났다. 하지만 이미 초기 모델에 overfitting이 존재하지 않았고, 적용 후 뛰어난 성능차이도 없지만 학습시간은 늘어나 드롭아웃은 사용하지 않기로 했다.

3) 최종 Model 채택



loss	categorical_crossentropy
optimizer	adam (lr=0.001)
activation function	relu + softmax
batch size	32
epochs	25

convolution과 maxpooling을 이용한 모델을 최종 모델로 선정하였다. 하이퍼파라미터는 다음 표와 같이 설정하였고, Dropout과 Data Augmentation은 적용하지 않았다.

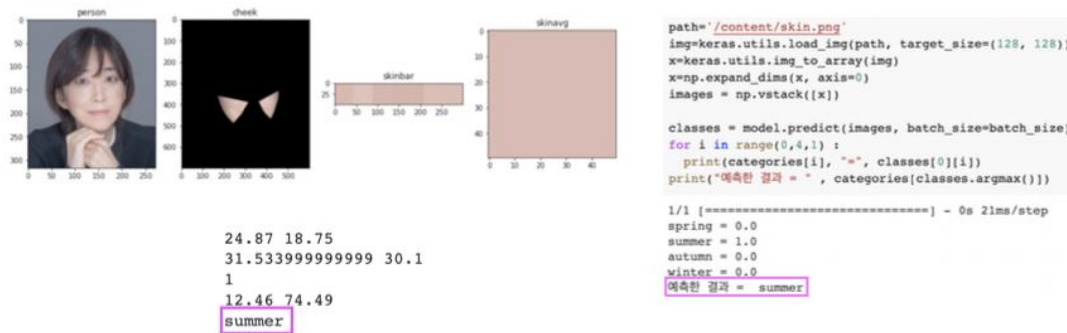
	초기 모델	최종 모델	최종 모델 - 초기 모델
train accuracy	0.9017	0.9425	+0.0408
train loss	0.2332	0.1456	-0.0876
valid accuracy	0.8457	0.8923	+0.0466
valid loss	0.3497	0.2037	-1.460

초기 모델과 최종 모델의 학습 결과를 분석한 표는 위와 같다.

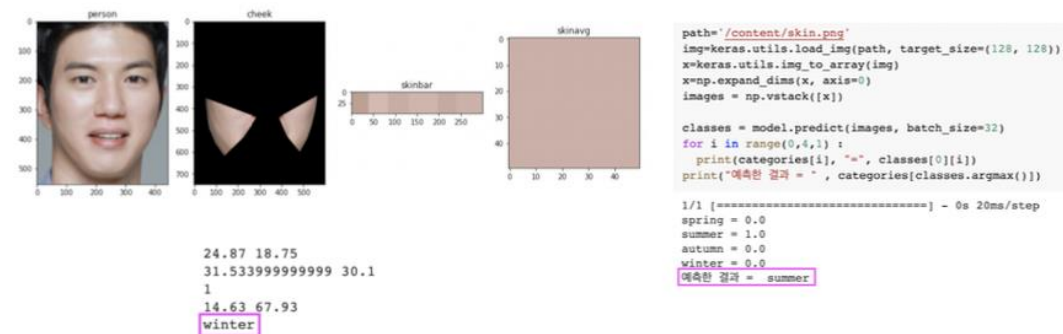
모든 영역에서 accuracy는 증가하고 loss는 줄어들어 모델 성능 개선에 성공했음을 보여주고 있다. 특히 가장 큰 차이를 보인 영역은 valid loss 였다. 이를 통해 train 에서만 학습 결과가 좋고, 예측은 잘 하지 못하는 overfitting 문제가 개선되었음을 알 수있다.

4. 학습 결과

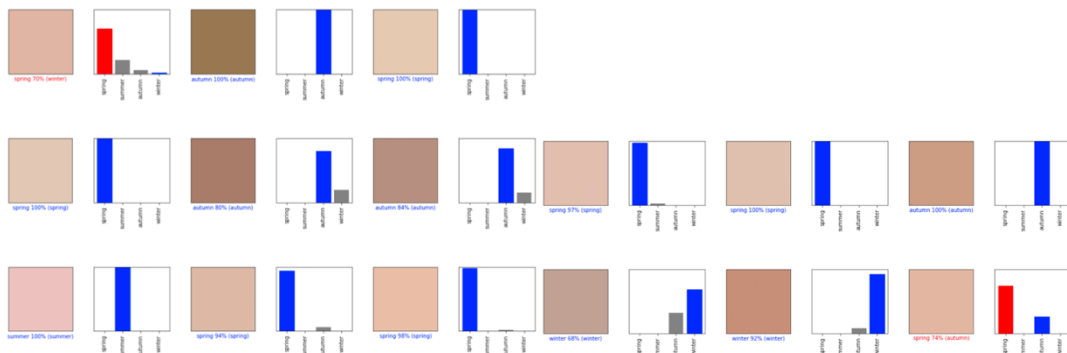
1) 테스트 데이터 예측 결과



다음은 데이터 전처리 진행 과정을 보여주고 있다. 볼 영역의 피부색을 추출해 만든 데이터를 hsv값을 통해 분류한 결과는 summer(여름 쿨톤)이 나왔고, 오른쪽과 같이 모델도 summer(여름 쿨톤)으로 예측하고 있었다.



다음과 같은 경우 hsv값으로 분류했을 때는 winter(겨울 쿨톤)이 나왔지만, 모델은 summer(여름 쿨톤)으로 예측하였다. 예측이 틀리긴 하였지만 같은 쿨톤으로 예측하였기 때문에 모델이 유의미한 예측을 하고 있다고 판단하였다.



모델의 성능을 더 평가해보기 위해 학습에 사용되지 않은 22개의 피부톤으로 퍼스널 컬러를 예측해보았다. 이 중 위와 같이 15개의 결과만 출력한 결과, 15개 중 13개의 퍼스널 컬러를 맞췄기 때문에 학습이 잘 되었음을 확인하였다.

5. 고찰

이번 프로젝트의 결론 및 고찰을 한 문장으로 표현하자면, 데이터셋 수집 및 분류에 대한 어려움과 모델 선택의 중요성 이라고 할 수 있다.

데이터셋의 경우, 처음부터 직접 얼굴 사진을 크롤링하고 피부색을 추출해야 했기 때문에 가장 많은 시간과 정성이 소요되었다. 많은 시행착오가 있었지만, 결론적으로는 모델 학습이 잘 되도록 데이터셋을 구축하였다. 하지만 유명인들의 사진으로는 이미지 특성 상 볼에는 백색광이 가장 많이 받는 곳이기 때문에 웬보다는 쿨 데이터가 더 추출되었고, 특히 봄웜톤을 데이터를 취득하는 것이 어려워 봄웜톤의 데이터가 상대적으로 부족한 상태였습니다.

이를 해결하기 위해 AI 인물 사진 생성기를 이용하려하였지만 15장 이상을 취득하려면 결제를 해야했기 때문에 사용하지 못했다. 만약 사진 데이터를 더 수집해 클래스 불균형을 해결하면 모델이 학습하는데 더 도움이 되었을 것이라고 생각한다.

또한 하이퍼 파라미터 튜닝을 통해 모델의 성능을 증진시킬 수 있음을 확인할 수 있었다. 이번 프로젝트에서 적용한 기법외에도 모델의 성능을 높일 수 있는 방법은 더 많이 존재하고 있다. 우선, learning rate에 따른 성능 개선이 제일 컸기 때문에 learning rate decay를 적용 하였으면 성능이 보다 더 좋아졌을 수도 있을 것 같다. 또한 여러 개의 개별 모델을 조합하여 최적의 모델로 일반화하는 앙상블 기법을 적용해보는 방법도 사용해 보면 좋을 것 같다.

6. 참고문헌

1. 박경미·윤가림·김영봉, 「계층화된 3차원 피부색 모델을 이용한 피부색 분할」, 『부경대학교 정보공학과』
2. 정지오, 「딥 러닝을 이용한 안면 여드름 분류 모델 연구」, 『배재대학교 컴퓨터공학전공』
3. codetorial.net/tensorflow/classifying_the_cats_and_dogs.html
4. <https://github.com/slinteruto/CAI/tree/c28f4a395cd59fbbd5009eda8fb4fd8ee93abbd8>
5. 문완목(2005), 「20대~30대 한국 여성의 베스트컬러에 관한연구」, 『홍익대 석사논문』
6. 김용현 외 2(2018), 「퍼스널 컬러 스킨 톤 유형 분류의 정량적 평가 모델 구축에 대한 연구」, 『(주)아이패밀리SC』