

데이터 분석 프로젝트

- 두피 상태 자가진단 -

과 정 : 기업 맞춤형 빅데이터
분석가 양성 과정

담 당 : 이경희

제 출 일 : 2024년 01월 11일

팀 명 : TEAM H

팀 장 : 김경수

팀 원 : 송재준, 신호섭, 안은지

목차

1. 프로젝트 및 프로그램 개요	03
1.1. 기안서	03
1.2. 기획 배경	04
1.3. 활용 데이터	05
2. 데이터 분석 모델	06
2.1. 이미지 분류 모델	06
2.2. EfficientNet	07
2.3. 분석 모델링 과정	11
2.4. 결과 분석	13
3. 시스템 모델링	16
3.1. 시스템 구성	16
3.2. 시스템 구축	17
3.3. 구현 기능	17
3.4. 결과	19
4. 결론 및 제언	21
5. 느낀 점	22
6. 참고자료	

1. 프로젝트 및 프로그램 개요

1.1. 기안서

프로젝트 개요

주제	두피건강(탈모) 자가진단 서비스		팀명	TEAM H
목적	<ul style="list-style-type: none"> - 웹에서 무료 제공되는 자가진단으로 뛰어난 접근성과 편리함 제공 - 서비스 제공자는 두피 케어와 관련된 광고를 통해 수익을 창출하는 형태의 상호이익 구조를 구축 			
목표	<ul style="list-style-type: none"> - 두피 사진을 업로드하면 두피 질환과 중증도를 자동 분류 - 성별 및 연령대별 두피 건강 통계 제공 - 보유 두피 질환과 연관된 제품 링크 및 인근 탈모병원의 위치를 제공하는 시스템을 웹으로 구축 			
주요기능	<ul style="list-style-type: none"> - 딥러닝 이미지 분석을 통한 탈모여부 진단 - 이용자에 맞는 통계차트 출력 - 두피케어 관련제품 정보, 링크 제공 - 지도 api로 인근 탈모병원 위치 제공 			
기대효과	<ul style="list-style-type: none"> - 간편한 자가진단 서비스로 집에서 무료 진단 가능 - 추천상품과 인근 병원 위치 확인으로 두피케어 방법 제공 - 실제 서비스 런칭 시, 광고로 서버비용을 충당 및 광고주는 새로운 판로 개척 			
역할분담	김경수(팀장)	모델링, 백엔드		
	송재준	모델링, 데이터 분석		
	신효섭	모델링, 프론트엔드		
	안은지	모델링, 백엔드		
데이터 출처	AI-Hub		데이터명	유형별 두피 이미지

개발 환경

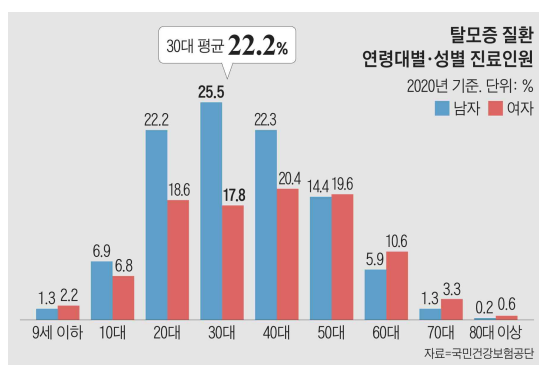
운영체제	Windows10 Pro	개발언어	python 3.10
프로세서	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz	라이브러리	torch 2.1.1 numpy 1.26.2 PIL 10.1.0 opencv 4.8.1.78
IDE	PyCharm 2023.2.5		
프레임워크	django 4.1.13		
분석모델	EfficientNet		

프로젝트 개발 일정

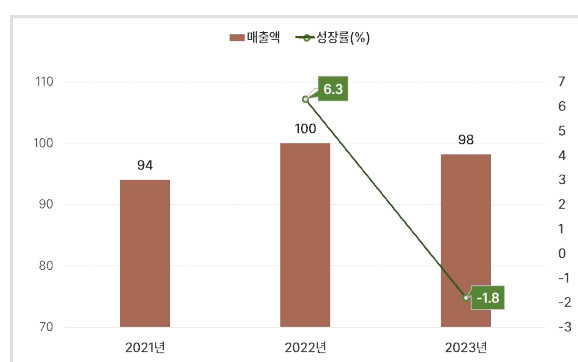
프로젝트 기간 (23/12/11 ~ 24/01/12)		1주차					2주차					3주차					4주차					5주차				
		11	12	13	14	15	18	19	20	21	22	26	27	28	29	02	03	04	05	08	09	10	11	12		
기획	프로젝트 주제 선정																									
	유사 서비스 분석																									
	계획 수립 및 기안서 작성																									
모델링	데이터 수집 및 분석																									
	데이터 모델링																									
	예측 모델 생성 및 보완																									
시스템 설계	UI 및 기능 구성 계획																									
	차트 기능 구현																									
	추천 상품 기능 구현																									
	병원 찾기 기능 구현																									
	웹 페이지 구현																									
평가	보고서 작성																									
	발표 자료 준비																									
	모의 시연																									
	발표																									
	평가																									

1.2. 기획 배경

흔히 탈모하면 떠오르는 것 중 하나가 바로 중년층에서 나타난다는 것이다. 하지만 이런 고정관념을 깨부수듯 ‘탈모증¹⁾’ 질환의 전체 진료 인원은 꾸준히 증가해 2020년에 23만3000명으로 집계되었는데, 30대가 22.2%(5만 2천 명)로 가장 많았고, 40대가 21.5%(5만 명), 20대가 20.7%(4만 8천 명)의 순으로 나타났다.²⁾ 또한 탈모 샴푸의 매출이 오름세를 보이며, 탈모 종류에 따라 건강보험이 적용된다는 사실은 탈모로 고통받고 있는 이들이 많다는 방증이 될 수 있다.



[그림 1] 연령대별 · 성별 진료인원



[그림 2] 오프라인 기준 탈모샴푸 매출 추이

위 자료에서는 탈모만 언급했으나, 다양한 두피 질환들이 탈모를 야기시킬 수 있다. 이런 것들까지 더해지면 환자 수는 더 증가할 것으로 보인다. 무릇 어떤 질병이든 초기에 진료를 받아야 병이 나을 확률이 높다. 그렇지만 막상 병원을 가려고 해도 야간진료 시간조차 방문하기 어려울 때가 있을 것이다. 그럴 때 '두피 자가진단 프로그램이 있다면 증상의 심각성을 조기에 간편하게 알 수 있지 않을까?'란 생각에서 출발해, 현재 시장에서 두피 자가진단이 상용화되지 않은 것을 확인하고 프로젝트명 ‘ScalpSpectra’를 시작하였다.

1) L63 - 원형 탈모증, L64 - 안드로젠성 탈모증(남성형 탈모증),

L65 - 흉터성 탈모증(반흔성 탈모증), L66 - 비흉터성 탈모증(비반흔성 탈모증)

2) 국민건강보험, "머리가 뭉텅뭉텅 빠지는 「탈모증」 질환 전체 환자 23만 명", 2021년 7월 15일, <https://www.nhis.or.kr/nhis/together/wbhaea01600m01.do?mode=view&articleNo=10809475>.

1.3. 활용 데이터

AI Hub에서 유형별 두피 이미지를 사용했다. 우리나라의 인구통계 데이터를 토대로 연령/성별 분포도를 고려하여 7개의 유형별로 이미지가 라벨링되었고, 그중 두피 관리 요구가 집중되어있는 연령대(10대~70대)를 중심으로 수집된 데이터다.³⁾

증상별 구분	건수
미세각질	17,434건
피지과다	80,416건
모낭사이홍반	67,414건
모낭홍반/농포	4,592건
비듬	40,482건
탈모	25,682건
양호	811건

[표 1] 유형별 두피 이미지

결과	출력
양호	0
경증	1
중등도	2
중증	3

[표 2] 데이터 라벨링

3) "유형별 두피 이미지," AI Hub, 2021년 09월 수정, 2024년 01월 02일 접속,
<https://aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=216>

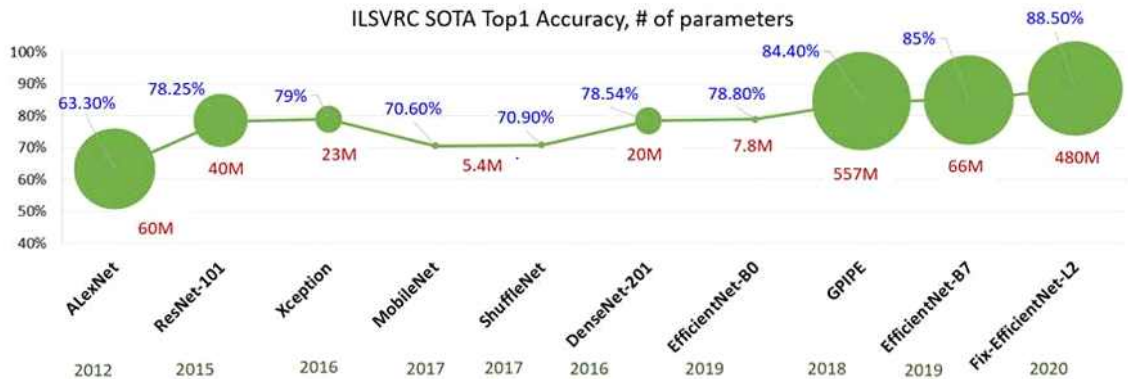
2. 데이터 분석 모델

이미지는 일반적으로 대규모 데이터이기에 머신러닝보다 딥러닝 모델로 처리한다. 그중에서도 CNN(Convolutional Neural Network)에서 파생된 모델들의 성능이 뛰어나고 일반적으로 사용된다.

2.1. 이미지 분류 모델

Inception (GoogLeNet)	
특징	여러 크기의 필터를 한 번에 사용하는 'Inception 모듈'을 사용한다.
장점	여러 크기의 필터로 다양한 특징을 잡아낸다.
단점	매개변수가 많아 메모리 사용량이 높다.
VGGNet (Visual Geometry Group Network)	
특징	작은 필터 크기(3x3)를 사용하여 깊이를 증가시킨다.
장점	간단한 구조로 이해하기 쉽다.
단점	깊은 네트워크 구조로 인해 많은 연산이 필요하다.
ResNet (Residual Network)	
특징	깊이가 깊어져도 성능이 개선되도록 잔여 학습(Residual learning)을 도입해 보완하였다.
장점	잔여 연결(Residual Connection)로 그래디언트 소실 문제를 완화한다.
단점	메모리 사용량이 많아질 수 있다.
DenseNet	
특징	Dense Block 사이에 avg pooling 연산을 수행한다.
장점	각 레이어 간의 밀접한 연결을 형성해 정보흐름을 향상시킨다.
단점	메모리 요구량과 학습 시간이 증가할 수 있다.
MobileNet	
특징	깊이별 분리 합성곱을 사용하여 모델 크기를 경량화했다.
장점	모바일처럼 자원이 제한된 환경에서 사용하기에 적합하다.
단점	더 큰 모델에 비해 정확도가 낮을 수 있다.

[표 3] CNN 계열 모델별 특징과 장단점

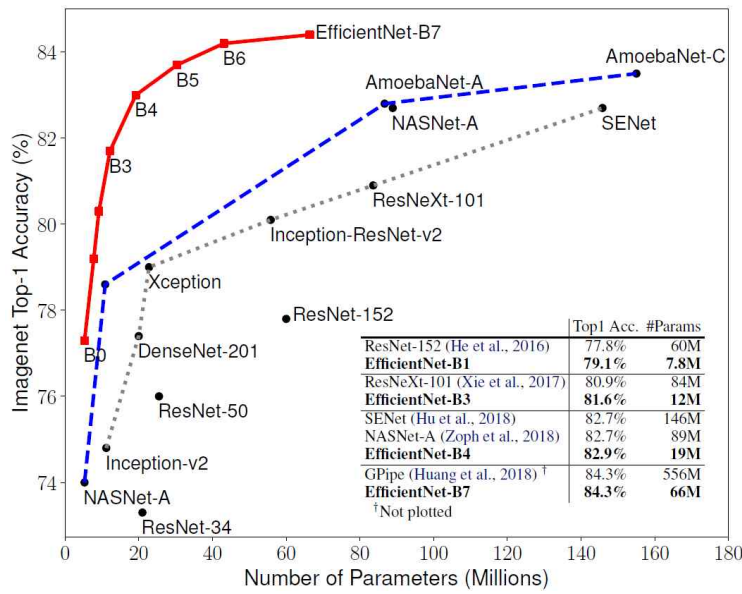


[그림 3] 2012 ~2020 ILSVRC SOTA 달성 1위 모델의 정확도와 파라미터

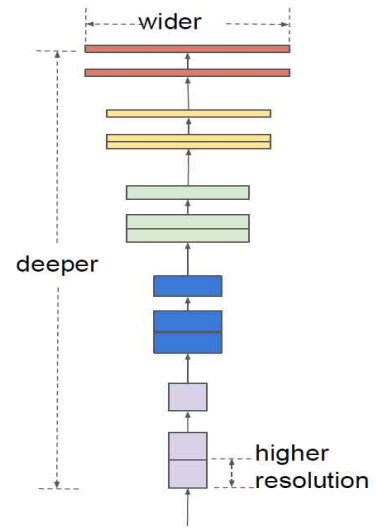
위 모델들을 탐색한 결과 네트워크의 깊이와 성능이 비례 관계가 아닌 것을 알 수 있었다. 마찬가지로 입력 이미지가 작다면 Convolution, Pooling층을 거치면서 정보 소실이 발생할 가능성이 높다. 반대로 크다면 가중치가 증가해 과적합을 야기한다. 가중치가 많다는 것은 사소한 것도 모두 정보로 인식하는 것으로, 지엽적인 정보까지 학습할 가능성이 높다. 모델의 복잡도와 연산량 증가까지 발생하기에 무조건 깊이와 넓이, 이미지 크기를 확대하는 것은 옳은 방법이 아니다. 해서 이 방법을 극복할 최적의 모델을 후술하고자 한다.

2.2. EfficientNet

최종적으로 선택한 것은 EfficientNet으로, 이름에서도 알 수 있듯이 효율적 모델이다. NAS(neural architecture search)을 사용하여 자동으로 최적화된 모델의 구조를 찾아내는 과정을 거쳐 만들어졌다. 다른 모델과 달리 깊이, 너비, 이미지 크기 3가지를 동시에 고려한 compound scaling을 적용해 파라미터 수 대비 성능이 압도적이다. 상대적으로 얇은 깊이와 파라미터가 적다는 것은, 적은 자원으로도 가능하다는 것이기에 주어진 환경에서 최적의 모델이라 여겨 선정하였다.



[그림 4] 모델별 파라미터와 정확도 관계

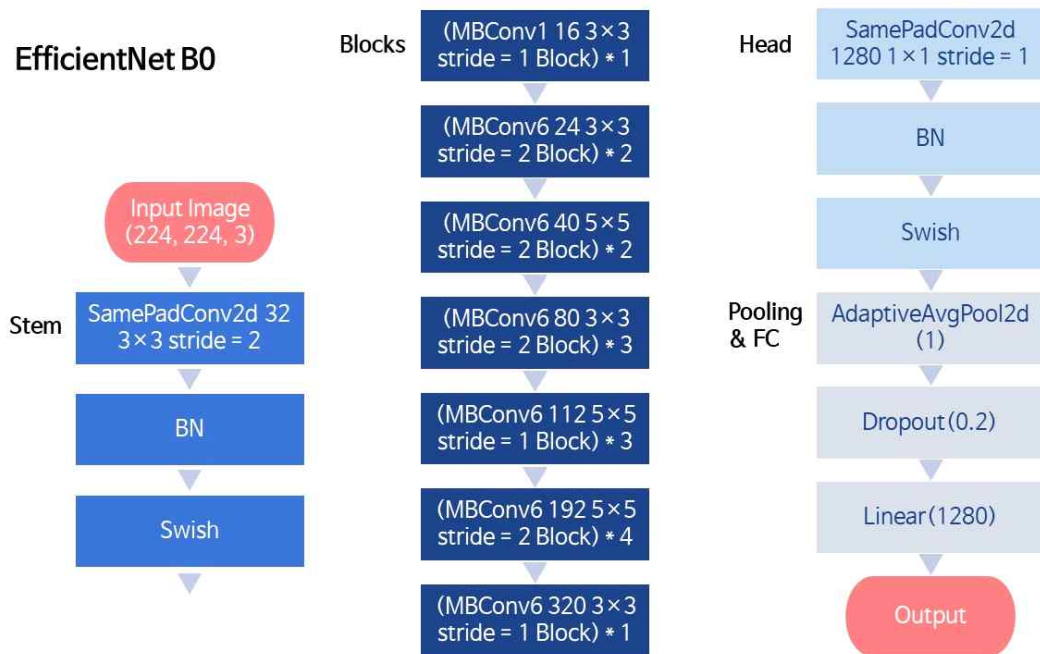


[그림 5] compound scaling

efficientnet-b0 ~ b8까지와 b7에서 크기를 더 키운 efficientnet-12가 있다. efficientnet-12이 가장 성능이 좋고 자원이 많이 필요하다. 자원의 한계가 있어 모든 미세각질, 피지과다 모델만 b3, 나머지는 b7을 사용하였다. 아래 표를 참고하면 모델이 점점 더 커지는 것을 알 수 있다.

model	width_coefficient	depth_coefficient	resolution	dropout_rate
b0	1.0	1.0	224	0.2
b1	1.0	1.1	240	0.2
b2	1.1	1.2	260	0.3
b3	1.2	1.4	300	0.3
b4	1.4	1.8	380	0.4
b5	1.6	2.2	456	0.4
b6	1.8	2.6	528	0.5
b7	2.0	3.1	600	0.5
b8	2.2	3.6	672	0.5
12	4.3	5.3	800	0.5

[표 4] efficientnet 모델별 파라미터 값



[그림 6] EfficientNet B0 모델 구조

EfficientNet의 구조는 보다시피 반복적인 블록으로 간단한 구성이다. Stem에선 3x3 컨볼루션 레이어와 이 모델에 더 효율적으로 계산하도록 변형된 Swish 활성화 함수로 구성되었다. 입력 이미지를 받고, 초기 특징 추출을 수행하는 부분이다.

Blocks에선 MBConv(Mobile Inverted Residual Block) 블록이 반복되는데 가장 핵심적인 요소다. 깊이별 분리 합성곱으로 효율적인 컨볼루션을 수행한다. Inverted Residual Block과 SE(Squeeze-and-Excitation)모듈로 확장, 압축 비율을 조절해 모델을 향상시킨다. 또한, ResNet에 쓰이는 잔여 연결(Residual Connection)로 기울기 소실 문제를 완화해 정보가 더 원활하게 흐르게 한다.

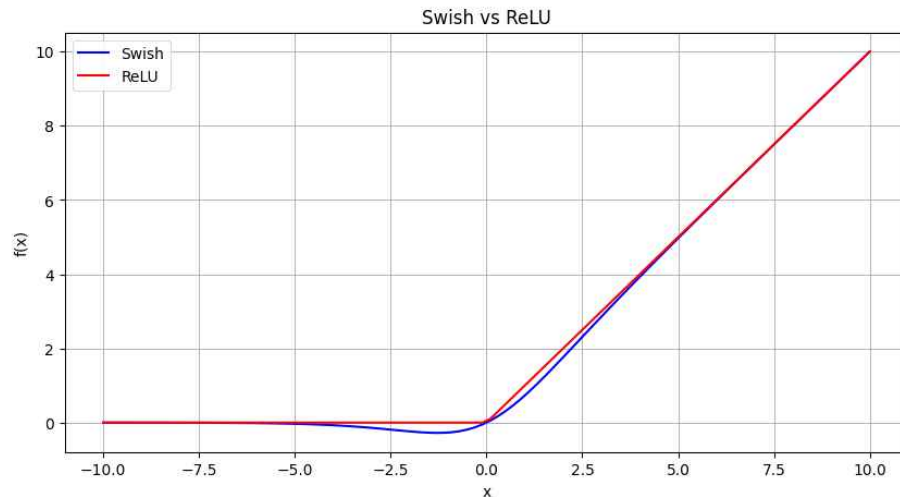
Head는 모델의 마지막 단계다. 특성 맵을 처리하고, Global Average Pooling으로 최종적인 특성을 추출한다. 그 후에 과적합을 방지하고, 마지막으로 완전 연결 레이어로 최종 예측을 수행한다.

- Swish

EfficientNet은 기존 Swish 함수와 모양은 비슷하지만, 변형된 Swish를 활성화 함수로 사용한다. 기존 Swish 식은 $f(x) = x \cdot \sigma(x)$, 변형된 식은

$f(x) = x \cdot \sigma(\beta x)$ 로 β 는 학습이 가능한 매개변수로 1보다 크면 기울기가 가파르게 변하고, 1보다 작으면 기울기가 완만하게 변한다.

Swish는 아래 그래프와 같이 약간의 음수 값을 허용하고 있기에 ReLU와 달리 기울기가 매끄러운 것을 알 수 있다. 미분 계산량이 많아지지만, 낮은 값에 대해서 Relu보다 정보의 손실(죽은 뉴런 현상)이 적다.



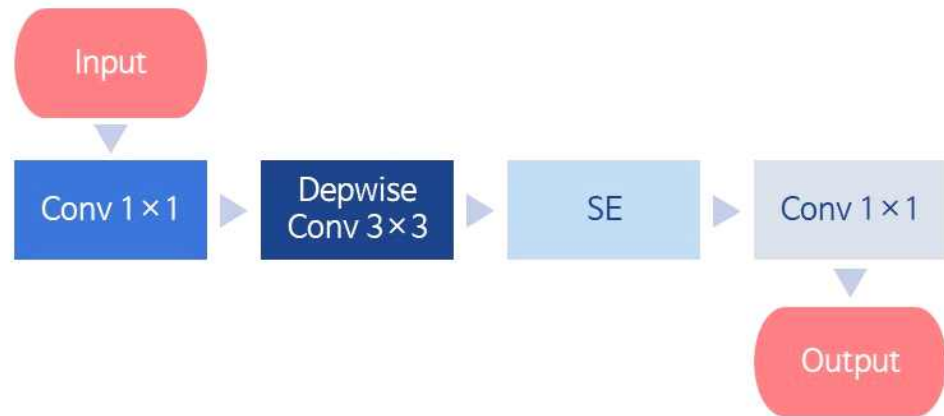
[그림 7] Swish vs. ReLU

- MBConv 구조

MBConv 구조는 딥러닝 모델의 경량화를 위해 개발된 MobileNetV2의 Inverted Residual Block에서 기인했다. Inverted Residual Block은 ResNet과 반대로 첫 번째 bottleneck에서 expansion factor를 곱해서 채널을 늘리고 두 번째 bottleneck에서 원래대로 돌려놓는다. 입력의 차원을 확장한 후 줄이는 형태로, 확장과 축소를 동시에 하지 않기에 연산량이 적어져 경량화된 구조가 되는 것이다.

여기에 Pooling, FC(Fully Connected Layer), 활성화 함수인 시그모이드로 구성된 SE(Squeeze-and-Excitation) 모듈로 채널 간의 중요도에 가중치를 부여하여 특성 맵의 중요한 정보를 높이고, 중요도가 낮은 정보를 낮추어 모델의 성능을 향상시킨다.

MBConv



[그림 8] MBConv 구조

2.3. 분석 모델링 과정

훈련 데이터와 검증 데이터를 약 7:3 비율로 나눠 각각 전처리를 한다. 데이터 전처리 기준은 훈련 데이터 기준으로 서술하였다.

- 데이터 전처리

`torchvision.transforms.Compose()`로 전처리에 쓸 함수들을 리스트로 묶었다. 이미지 입력 크기를 맞추고, 이미지를 보다 매끄럽게 변환하고, 비교적 적은 이미지로도 훈련 효과를 높이기 위해 사용하였다.

함수	매개변수	설명
Resize()	size = 300	이미지의 높이와 너비를 튜플로도 줄 수 있다.
	interpolation = InterpolationMode.BILINEAR	InterpolationMode.BILINEAR이 기본값으로 양산형 보간법이다. 실수 좌표를 둘러싸고 있는 네 개의 픽셀 값을 이용해 각 픽셀 값과의 거리의 반 비례값을 가중치로 계산한다. 최근방 이웃 보간법에 비해 느리지만 상대적으로 빠르며 계단 현상이 감소된다.
	antialias = 'True'	기본값이 'True'로 마찬가지로 계단 현상을 감소시킨다.
RandomHorizontalFlip()	p = 0.5	50%의 확률로 수평으로 뒤집는다.
RandomVerticalFlip()	p = 0.5	50%의 확률로 수직으로 뒤집는다.
Lambda()	lambda x: x.rotate(90)	시계방향으로 90도를 회전시킨다.
RandomRotation()	10	무작위로 ± 10 까지 회전시킨다.
RandomAffine()	degrees = 0	RandomRotation()로 회전 각도를 지정했기에 0을 주었다.
	shear = 10	이미지 원근을 10만큼 왜곡한다.
	scale = (0.8, 1.2)	이미지의 크기가 80 ~ 120% 범위에서 무작위로 조절된다.
ToTensor()		이미지를 tensor로 변환한다.
Normalize()	mean = [0.485, 0.456, 0.406]	ImageNet에서 널리 사용되는 평균 값이다.
	std = [0.229, 0.224, 0.225]	ImageNet에서 널리 사용되는 표준편차 값이다.

[표 5] 데이터 전처리 함수

- 모델 훈련

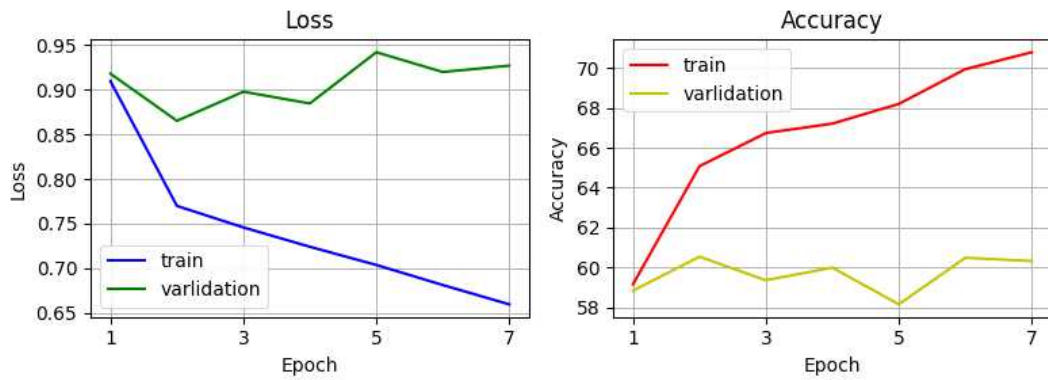
파라미터 값을 정의하고, `train_model()`로 모델을 훈련시킨다. 지정한 에포크 수에 도달해도 Trainset의 성능이 개선되지 않으면 조기종료로 한정된 자원 속에서 최적의 모델을 찾을 수 있도록 하였다.

매개변수	값	설명
model	EfficientNet. from_pretrained()	사전학습된 가중치를 포함한 모델을 가져온다.
criterion	nn.CrossEntropyLoss()	손실 계산 함수로 모델의 예측 값과 실제 값의 차이를 측정하는데 사용한다. 다중 클래스 분류에 주로 사용한다.
optimizer_ft	optim.Adam(model. parameters(),lr = 1e-4)	최적화 알고리즘으로 손실을 최소화하기 위해 사용한다. 모델 활성화함수를 Adam, 학습률은 0.0001로 설정했다.
exp_lr_scheduler	optim.lr_scheduler.StepLR(optimizer_ft, step_size=7, gamma=0.1)	step_size만큼의 에포크마다 학습률에 gamma를 곱하고 갱신해 학습률을 감소시킨다.
num_epochs	num_epochs = 100	학습할 총 에포크 수
patience	5	연속된 성능 개선 없는 에포크 수가 patience이상이면 조기종료한다. 기본 값을 5로 설정하였다.

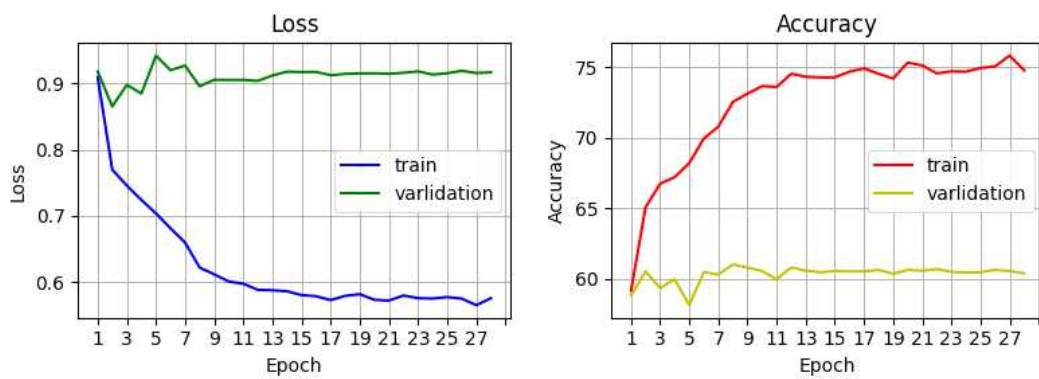
[표 6] 모델 훈련 함수

2.4. 결과 분석

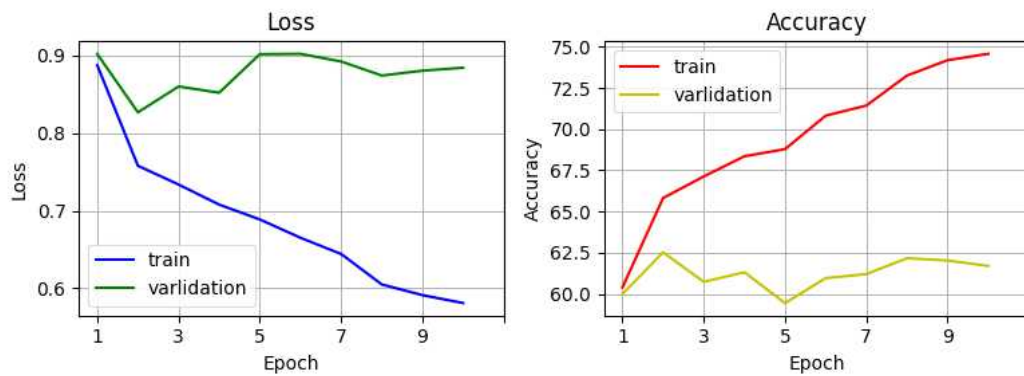
처음에 시도한 모델은 efficientnet-b0로 epoch 10, patience 5 기준 소요 시간 12분정도다. 소요시간은 짧지만 정확도가 60으로 매우 떨어지는 것을 알 수 있다. 같은 모델을 epoch 100, patience 20으로 늘렸더니 소요시간이 47분정도 걸렸으나, 정확도가 61로 매우 미미한 수준이다. 입력 크기를 300으로 바꿔서도 해보았으나 정확도는 62가 나왔다. b0는 훈련, 검증 둘다 정확도가 낮고, 검증 데이터의 손실율이 유의미한 변화가 없는 것을 확인했다.



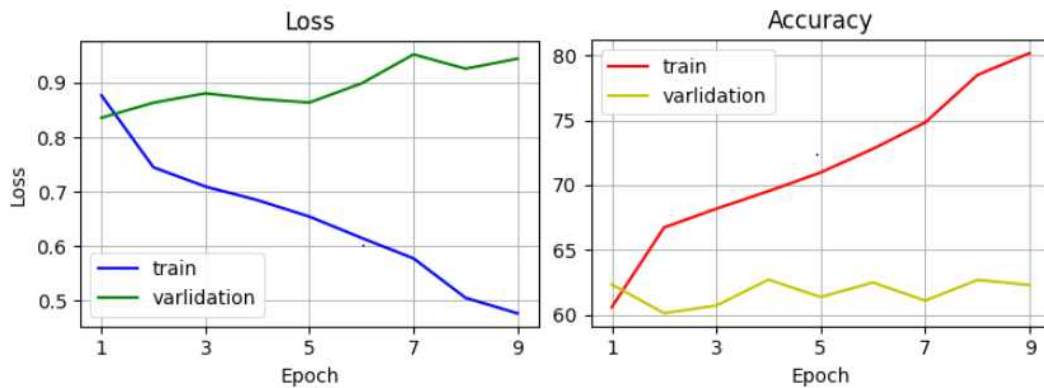
[그림 9] efficientnet-b0 epoch 10, patience 5



[그림 10] efficientnet-b0 epoch 100, patience 20

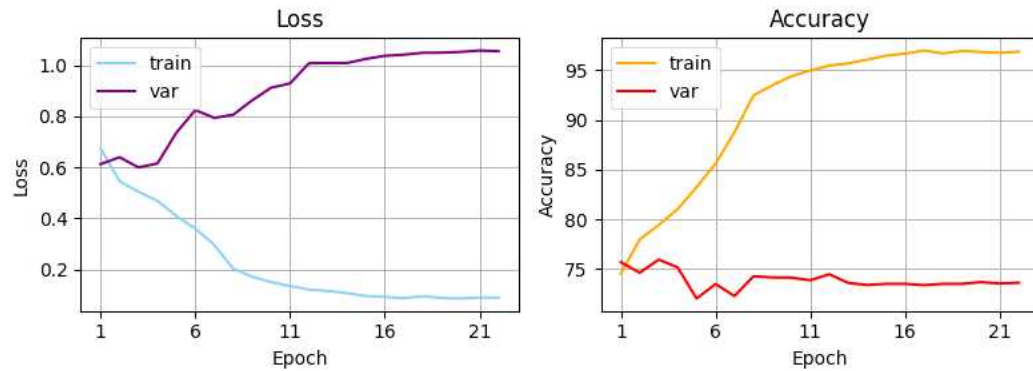


[그림 11] efficientnet-b0 epoch 10, resolution 300



[그림 12] efficientnet-b3

efficientnet-b3의 손실률과 정확도다. epoch 10번 기준 소요시간 3시간이 정확도 80으로 b0에 비해 정확도는 올랐지만 시간대비 효율이 매우 안 좋은 것을 알 수 있다.



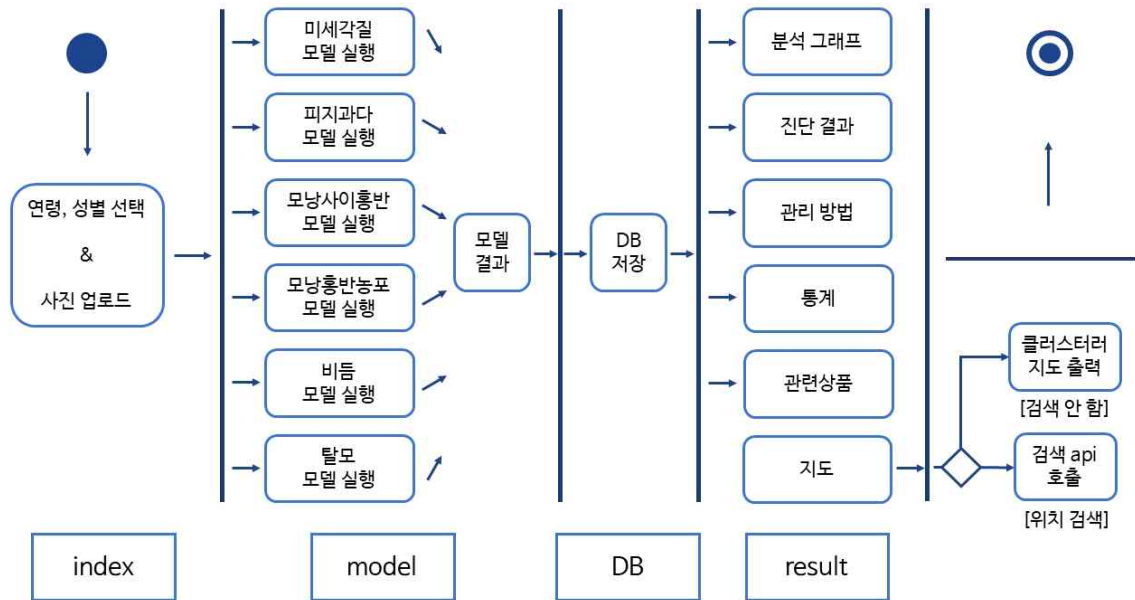
[그림 13] efficientnet-b7

efficientnet-b7 손실률과 정확도다. 에포크 20번 기준 소요시간 3시간이 걸렸으나, 정확도는 74%로 시간과 성능이 개선된 것을 알 수 있다. b7의 resolution은 600×600 이다. 하지만 이 크기로 모델을 진행할 경우, 현재 환경에서는 메모리가 부족해 제대로 동작하지 않기에 가용 메모리를 최대한 활용할 수 있게 300으로 조절하였다. 앞선 결과에서 볼 수 있듯이 똑같이 이미지 크기를 300으로 해도 b7이 더 잘나왔기에 이미지 용량이 상대적으로 적은 것들은 b7으로 모델을 돌렸다.

모델을 변경하면서 데이터의 정확도는 높아지고 손실값은 낮아져 성능은 높아졌지만, 훈련 데이터와 검증 데이터를 비교했을 때 과적합은 여전히 해결되지 않았다. 자원이 더 풍부했더라면 에포크나 다른 파라미터 조정까지 해봤을 텐데 그러지 못하였다. 비록 최적의 선택은 아니었으나, 실제 현장에서도 언제나 자원이 풍부할 것은 아니기에 나름대로 최선의 선택으로 결과를 내보았다.

3. 시스템 모델링

3.1. 시스템 구성



[그림 14] 시스템 구성

Human				
필드명	설명	Type	PK	Not Null
id	일련번호	integer	✓	✓
sex	성별	varchar		✓
age	연령	varchar		✓
pred	진행도	varchar		✓
skin	증상	varchar		✓

[그림 15] DB 테이블

3.2. 시스템 구축

아래와 같은 장점이 있어 프로젝트를 서비스하기 위해 사용하였다.

장고 프레임워크(Django Framework)

1. 파이썬기반 오픈소스 웹 애플리케이션 프레임워크라 파이썬 라이브러리를 그대로 사용가능하다.
2. MVC 패턴 기반 MTV(Model-Template-View)로 웹 개발을 보다 모듈화하고 유지보수 가능하게끔 설계되었다.
3. ORM(Object-relational mapping)기능 지원으로 SQL 질의문을 작성하지 않아도 DB와 상호작용할 수 있다.
4. 쉬운 DB관리를 위해 프로젝트를 생성하면서 관리자 기능을 제공한다.

파이토치(PyTorch)

1. 파이썬기반 오픈소스 머신러닝 라이브러리로 텐서플로보다 빠르다.
2. 메모리에서 연산하는 동시에 신경망 사이즈를 최적화하면서 동작한다.
3. Numpy를 대체하면서도 GPU로 연산이 가능하며 유연하고 빠르다.

파이썬 애니웨어(PythonAnywhere)

1. 웹 기반 Python 개발 플랫폼으로 별도의 웹 서버를 구성하거나 유지 관리할 필요 없이 간단히 애플리케이션을 작성할 수 있다.
2. 파이썬에 특화된 환경을 제공하기 때문에 주피터 노트북이나 파이썬 등을 설치하지 않아도 사용할 수 있다.
3. 서버 기반의 파이썬과 Bash 명령 줄 인터페이스에 대한 브라우저 내 접근을 제공한다.

[표 7] 시스템 구축 툴

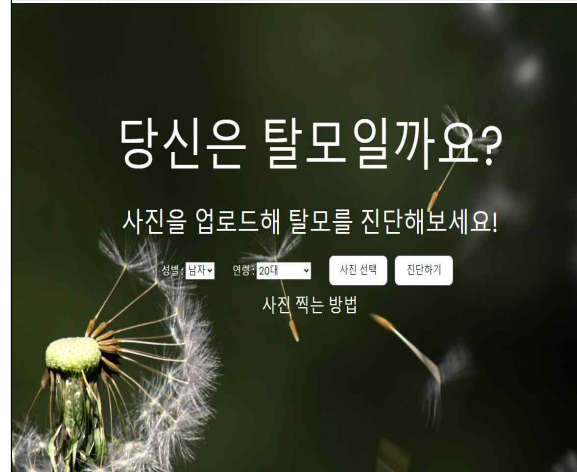
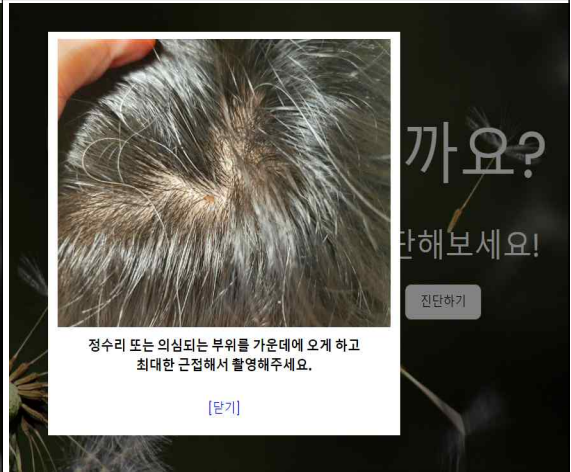

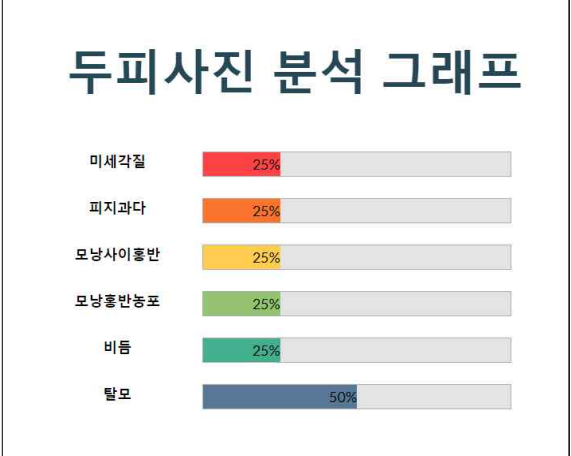
3.3. 구현 기능

기능	라이브러리 및 함수	설명
분석 그래프	def upform()	views.py에서 모델별 진단 결과(i), 증상 유형(j), 배경색(k)를 받고 for문으로 증상별 그래프를 표현한다.

진단 결과	def upform()	views.py에서 두피형{{headskin}}, 탈모 진행도{{presresult}}를 받아 출력한다.
	SQLite	index에서 버튼 선택과 모델 결과를 받아 DB에 저장한다.
관리 방법	def upform()	두피 유형별 이미지를 저장해 views.py에서 전달받은 {{headskin}}을 이미지 경로에 넣어 유형별 이미지를 출력한다.
통계	Tableau Public	Tableau에 업로드한 데이터셋과 차트 형식을 임베딩 한 뒤, 사용자가 입력한 성별과 연령에만 해당하는 자료들의 통계 차트를 제공한다.
관련 상품	bs.py-table()	oliveyoung 사이트에 있는 ‘두피케어’ 검색어로 검색된 상품들의 이미지, 이름, 가격을 가져와서 Table로 저장 후 이 중 랜덤으로 표본을 뽑아서 표시한다. 버튼 클릭 시 views.py에서 두피형을 검색어에 맞게 수정한 {{skintype}}을 링크에 넣어 두피형에 맞는 제품을 색한다.
지도	def mapmarker()	병원 list에서 마커의 위치를 나타낼 좌표와 인포윈도우에 나타낼 병원이름, 주소를 json으로 가져온다.
	Kakao 지도 Web API: libraries = services, clusterer	위치 검색과 클러스터러 마커를 표시를 지원한다.

[표 8] 웹 구현 기능

3.4. 결과

<p>인덱스 페이지</p> 	<p>팝업 - 사진 찍는 방법</p> 
<p>사진 선택</p> 	<p>결과페이지 - 두피사진 분석 그래프</p> 
<p>결과페이지 - 진단결과</p> <p>진단 결과</p> <p>당신은 '탈모성' 두피이며, 진행도는 '중증' 입니다.</p>	<p>결과페이지 - 두피 특징/관리법</p> <p>탈모성두피 특징/관리법</p> <p>특징</p> <ul style="list-style-type: none"> 두피에 열룩 현상이 보임 피지 분비와 민감성이 높고 모발이 얇아짐 <p>관리방법</p> <ul style="list-style-type: none"> 전문의가의 진단을 통해 탈모원인 파악 6개월~1년정도 성장기 모발의 사이클 개선 탈모샴푸, 탈모앰플 사용

<div> <div>결과페이지 - 통계</div> <div> <div>통계</div> <div>20대 남성들의 증상별 통계그래프를 보여드립니다.</div> <div> <div>미세각질</div> <div> <div> <div>영호</div> <div>77.0%</div> </div> <div> <div>경중</div> <div>19.9%</div> </div> <div> <div>중등도</div> <div>7.7%</div> </div> <div> <div>중증</div> <div>1.4%</div> </div> </div> <div> <div>미세각질</div> <div>영호</div> <div>77.0%</div> <div>경중</div> <div>19.9%</div> <div>중등도</div> <div>7.7%</div> <div>중증</div> <div>1.4%</div> </div> </div> </div> </div>	<div> <div>결과페이지 - 관련 상품</div> <div> <div>관련 상품</div> <div> <div> <div>이름:스케일링 샴푸</div> <div>가격:16800</div> </div> <div> <div>이름:여성두피용 샴푸</div> <div>가격:31000</div> </div> </div> <div> <div>관련상품 더보기</div> </div> </div> </div>
<div> <div>관련상품 - 상품 더 보기</div> <div> <div>'탈모 두피' 검색결과 (전체 137개의 상품)</div> <div> <div> <div>카테고리</div> <div>헤어케어</div> <div>한조케어</div> <div>건강식품</div> </div> <div> <div>브랜드</div> <div> <div>+</div> <div> <div>닥터로제</div> <div>라보에이치</div> <div>달리프</div> <div>루몬</div> </div> </div> </div> <div> <div>제품특징</div> <div> <div>+</div> <div> <div>탈모증상완화가능성</div> <div>약산성</div> <div>저자극</div> <div>Non-실리콘</div> </div> </div> </div> <div> <div>혜택</div> <div> <div>+</div> <div> <div>전체</div> <div>루몬샴푸</div> <div>세일상품</div> <div>1+1</div> </div> </div> </div> <div> <div>가격대</div> <div> <div>최저가</div> <div>~</div> <div>최고가</div> <div>가격대 적용</div> </div> </div> </div> <div> <div>전체137개의 상품이 등록되어 있습니다.</div> <div> <div>인기순</div> <div>신상순</div> <div>판매순</div> <div>낮은가격순</div> <div>할인율순</div> </div> <div> <div> <div>베스트</div> <div>가이 밀거두온</div> </div> <div> <div>ONLY</div> <div>LABO-H</div> </div> <div> <div>5중</div> <div>인상</div> </div> </div> </div> </div></div>	<div> <div>결과페이지 - 근처 병원 찾기</div> <div> <div>근처 병원 찾기</div> <div> <div>현재 위치를 입력해 근처 병원을 검색해보세요!</div> <div> <div> <div>근처구</div> <div>주소 검색</div> </div> <div> <div> <div>강북구</div> <div>서울특별시 강북구 시흥대로 228, 2층 (시흥)</div> </div> </div> </div> </div> </div></div>
<div> <div>장고 관리자 페이지</div> <div> <div>Select human to change</div> <div> <div>Action: <div></div> Go 0 of 100 selected</div> <div> <div> <div>HUMAN</div> <div>30대, 여, 중등도, 복합성</div> <div>10대, 남, 중등도, 복합성</div> <div>10대, 남, 중등도, 복합성</div> <div>10대, 남, 중등도, 복합성</div> <div>10대, 남, 중등도, 복합성</div> <div>10대, 남, 중등도, 복합성</div> <div>50대, 여, 중등도, 복합성</div> <div>50대, 여, 중등도, 복합성</div> <div>50대, 여, 중등도, 복합성</div> <div>50대, 여, 중등도, 복합성</div> <div>50대, 여, 중등도, 복합성</div> </div> </div> </div> </div> </div>	<div> <div>SQLite</div> <div> <div>SQLite Expert Personal 5.5 (x64)</div> <div> <div>File View Database Object SQL Transaction Tools Help</div> <div> <div>Database: db Table: bald_human File: C:\Final Project\Final\db\sqlite3</div> <div> <div>Database Data DDL Design SQL</div> <div> <div>rowid id sex age pred skin</div> <div> <div>1 1 남 10대 중등도 탈모성</div> <div>2 2 남 10대 중등도 탈모성</div> <div>3 3 남 40대 중등도 탈모성</div> <div>4 4 남 10대 중등도 복합성</div> <div>5 5 남 10대 중등도 복합성</div> <div>6 6 남 10대 중등도 복합성</div> <div>7 7 남 10대 중등도 복합성</div> <div>8 8 남 10대 중등도 복합성</div> <div>9 9 남 10대 중등도 복합성</div> <div>10 10 남 10대 중등도 복합성</div> <div>11 11 남 10대 중등도 복합성</div> <div>12 12 남 10대 중등도 복합성</div> <div>13 13 남 10대 중등도 복합성</div> <div>14 14 남 10대 중등도 복합성</div> <div>15 15 남 70대 중등도 복합성</div> <div>16 16 남 10대 중등도 복합성</div> <div>17 17 남 10대 중등도 복합성</div> <div>18 18 남 10대 중등도 복합성</div> <div>19 19 남 10대 중등도 복합성</div> <div>20 20 남 10대 중등도 복합성</div> </div> </div> </div> </div> </div> </div></div>

4. 결론 및 제언

한정된 자원에서 정확도를 높이는 것은 어려운 일이다. 분석 결과 훈련 데이터를 기준으로 각 모델들의 정확도는 95% 수준을 보여준다. 다만, 검증 데이터를 기준으로 하면 약 20%p 정도 더 낮게 나오는 과적합이 나타났다. 두피 이미지는 비정형 데이터지만, 두피 유형별로 생성되는 이미지는 큰 틀에서 변동성이 크지 않기 때문에 데이터를 증강해 모델 훈련을 더 시키는 방법도 한 가지 방안이라고 사료된다. 보다 성능을 개선한 efficientnet-v2로 모델을 돌리거나, 다른 모델과 앙상블하면 더 나은 결과를 얻을 거라 추측된다.

다소 걱정될 수 있는 점은 일반 사용자들이 의료 기기로 사진을 찍을 수 없다는 사실이다. 스마트폰에 탑재하는 렌즈와 의료용 렌즈의 성능은 당연히 차이가 클 수밖에 없기에 심각한 증상이 아닌 이상 경증은 발견하기 어려울 수 있다. 그러나 의료용 렌즈 수준이 아니더라도, 테스트 이미지로 성능을 확인해봤을 때 해당 질병으로 예측했기에 간이진단용으로는 충분하다고 판단했다.

이 프로젝트는 광고주에게 광고를 수주받을 것을 가정하고 진행하였다. 그래서 현재 기능은 랜덤으로 제품을 노출시키는 방식을 택했고, 후에 광고주에 맞게 수정할 예정이다. 첫 페이지에서 사용자는 연령과 성별을 입력하면, 이 정보와 모델 결과가 DB에 저장된다. 이 데이터로 각 상품별로 두피 유형뿐만이 아닌 성별과 연령에 맞게 추천하는 시스템으로 확장시킬 것이다. 더해서 해당 상품의 클릭 수까지 DB에 저장한다면 상품 판매에 도움이 될 것이라고 본다.

상품 광고뿐만 아니라 똑같은 앱과 연계해 지도로 주변 병원을 찾은 후, 바로 병원을 예약할 수 있는 기능을 추가한다면 시너지가 날 것으로 예상된다. 더하여 두피 질환 관련 질문을 받아줄 챗봇까지 만든다면 자가진단부터 두피케어까지 보다 편리하게 두피를 관리할 수 있을 것이다.

현재 데이터는 한국인을 대상으로 한 두피 이미지만 있어서 모발의 색은 검정, 갈색, 드물게 흰색으로 구성되어있다. 만약 시장을 확장한다면 각 인종별로 두피 이미지를 넣고 모델을 만들면 좋을 것이다.

5. 느낀 점

- 김경수

일반적인 머신러닝과 다르게 이미지를 이용한 딥러닝은 컴퓨터의 성능이 중요하다는 것을 느꼈습니다. 무언가를 해보려 하면 out of memory 떠버리니까 적당한 크기와 batch size를 찾는 데에 제한되어있는 컴퓨팅 단위를 많이 소모한 것 같습니다. 그리고 Django를 이용하여 웹 서비스를 하는데, 외부에서 하나 하나 틀을 만들어놓고 웹에 장착하는 것이 재미있었습니다.

- 송재준

주로 사용하던 프로그램에서 제공되지 않는 기능이 자료 검색, 구현에 있어서 얼마나 쉽지 않은지를 알 수 있었다. 완성은 할 수 있었지만 그 과정에서 틀 자체의 한계로 어느정도 타협이 필요했고, 다른 더 좋은 방식이 있는지 찾아보고 싶어졌다.

- 신호섭

웹 뼈대를 만들어 놓고 완성된 부분을 받아서 집어넣었는데 퍼즐 맞추는것 같았고 잘못된 부분을 눈으로 확인해가며 고칠 수 있어서 재밌었다. 사용자에게 효율적으로 보여줄 수 있는 방법을 많이 고민했고 많이 배운것 같다.

- 안은지

프로젝트로 얻은 것도 많지만 얼마나 무지한지 적나라하게 알게 된 소중한 시간이었다. 모델을 이해하기 위해 원래 코드와 해당 논문을 보면 선행 연구들까지 봐야 하는데 계속 모르는 개념들이 나와 시간이 오래 걸렸다. 아는 만큼 보인다라는 말을 백번 실감했다. 일례로 뒤늦게 EfficientNet-v2를 알게 되었는데 이미 자원을 다 써버린 터라 모델을 실험해보지 못한 것이 가장 아쉬웠다.

비록 웹에서 지도 기능만 담당했지만, api도 공식문서를 숙지해야 응용이 가능함을 되새기는 계기가 되었다. 또 웹은 상호작용으로 결과물이 눈에 보이는 부분이라 코드가 논리상 맞아도 무언가 작은 것 하나라도 잘못되었다면 제대로 동작하지 않는다는 것을 며칠동안 매달리며 알게 되었다.

6. 참고 자료

- 보도자료

국민건강보험, "머리가 뭉텅뭉텅 빠지는 「탈모증」 질환 전체 환자 23만 명", 2021년 7월 15일,
<https://www.nhis.or.kr/nhis/together/wbhaea01600m01.do?mode=view&articleNo=10809475>

- CNN

<https://cs231n.github.io/convolutional-networks/>
<https://junklee.tistory.com/111>
<https://inhovation97.tistory.com/25>

- Inception

<https://velog.io/@ss-hj/Inception%EC%9D%B4%EB%9E%80>
<https://gaussian37.github.io/dl-concept-inception/>
<https://inhovation97.tistory.com/45>

- Resnet

<https://blog.creation.net/mxnet-part-5-vgc16-resnet152>

- DenseNet

<https://inhovation97.tistory.com/47>

- MobileNet

<https://n1094.tistory.com/29>
<https://m.blog.naver.com/laonple/222599873338>

- EfficientNet

<https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet>
<https://hoya012.github.io/blog/EfficientNet-review/>
<https://bellzero.tistory.com/17>
<https://lynnshin.tistory.com/53>
<https://m.blog.naver.com/koreadeep/222648217764>
<https://jsc5565.tistory.com/9>
<https://blog.naver.com/PostView.naver?blogId=qwopqwop200&logNo=222053989967>
<https://magicode.tistory.com/67>
<https://self-deeplearning.tistory.com/entry/PyTorch%EB%A1%9C-EfficientNet-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0>

- 모델 관련

<https://dotiromooook.tistory.com/21>

<https://velog.io/@wsh7787/CS231n-Lecture-9#vgg>

<https://blog.naver.com/worb1605/221386398035>

<https://medium.com/the-artificial-impostor/more-memory-efficient-swish-activation-function-e07c22c12a76>

- 시스템 모델링

<https://howudong.tistory.com/79>

<https://www.guru99.com/ko/best-python-hosting-services.html>

<https://www.pythonanywhere.com/>

- html 템플릿

index.html https://startbootstrap.com/theme/creative#google_vignette

result.html <https://startbootstrap.com/theme/freelancer>

민들레 이미지 <https://www.pexels.com/ko-kr/photo/289323/>

- tableau

https://help.tableau.com/current/api/embedding_api/en-us/docs/embedding_api_filter.html