

# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

**두피Dr**은 문제성 두피로 인해 고민하는 사람들을 위해 탄생했습니다.

디지털 현미경만 있으면 집에서 간편하게 6가지 유형별(미세각질, 피지과다, 모낭사이홍반, 모낭홍반농포, 비듬, 탈모)로 두피상태를 분석해주고, 이에 따른 효과적인 두피 관리방법을 제시합니다.



# Warning

이 후 ppt의 슬라이드에 두피를 현미경으로 찍은 다소 징그럽고, 자극적인 사진이 담겨있을 수 있음을 미리 알려드립니다. 심신미약자이거나 환 공포증 등이 있으신 분은 중간부터 소리만 들으실 것을 권장드립니다.

# 두피 상태 분석 서비스

6TEAM\_Project\_1 2021.08.13 ~ 2021.09.10  
**Dupi.Dr**

# [ CONTENTS ]

## 1. 두피케어의 현재

- 문제성 두피
- 소비자 분석

## 2. Dupi.Dr 프로젝트소개

- 조직 구성
- 업무분장
- 요구사항정의
- 메뉴구성도
- 서비스 구상도/ 서비스 흐름도
- 화면설계서
- Usecase/ ERD/ DFD
- 추천 두피 관리방법 제시
- Basemodel\_ EfficientNetB4
- 개발환경
- 예산

## 3. Dupi.Dr 기대효과

## 4. Dupi.Dr 실사

## 5. Reference

- DB테이블명세
- 핵심소스코드
- 프로젝트 관리

# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 1. 두피케어의 현재



머리가 너무 많이 빠져서  
스트레스 받아본 적 있으신가요?



두피에 뽀루지가 나서  
스트레스 받아본 적 있으신가

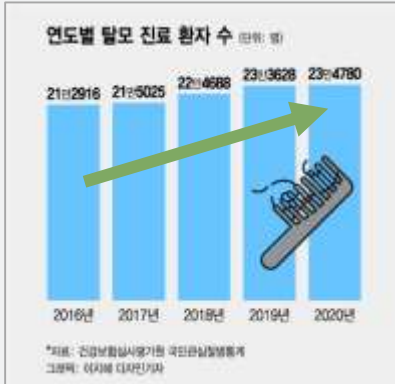




**비듬** 때문에  
스트레스 받아본 적 있으신가요?



## 탈모 환자 수 증가 추세



최근 5년간 탈모증으로 병원을 방문해 진료를 받은 국민은 **109만명**으로 나타났다.

2015년 20만 8601명이던 탈모 환자는 2016년 21만 2141명, 2017년 21만 4217명, 2018년 22만 4743명, 2019년 23만 2167명으로 **5년간 11.3% 증가**했다.

진료비는 지난 5년간 1486억원이 지출됐으며, 2015년 246억원 수준이던 탈모 진료비는 지난해 **361억원**으로 **46.5% 급증**했다.

또한, 제약업계와 유비스트 데이터는 국내 잠재적 **탈모 인구가 1000만명**에 육박할 것으로 추정한다.

<출처 : 메디칼업저버(<http://www.monews.co.kr>)>

## 한 해, 두부 지루 환자 수 249,000명

국민건강보험공단의 진료 통계 자료를 보면, 지난해 '두부 지루(질병코드 L21.0)' 환자는 모두 **24만9천명**으로, 인구 10만명당 501명 정도가 이 질병으로 병원을 찾았습니다.

두부 지루는 머리 피부의 피지샘 활동이 늘어 피지(기름)가 지나치게 많이 분비되고 이 때문에 비듬이나 염증이 생긴 경우를 말합니다.

<출처:2013년 국민건강보험진료통계  
<https://news.kbs.co.kr/news/view.do?ncd=2792533> >

## 한 해, 비듬 환자 수 365,381명

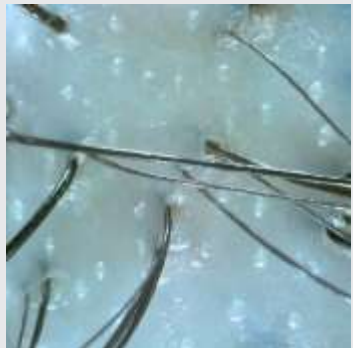
2012년 국민건강보험 진료통계를 보면 비듬환자는 연중 발생하지만 특히 겨울철(12~2월)이 9만9586명으로 가장 많았다. 봄(3~5월)이 9만5000명으로 뒤를 이었다. 여름(6~8월, 8만7466명)과 가을(9~11월, 8만3329명)보다 더 많이 발생한다.

<출처:2012년 국민건강보험진료통계  
<https://m.khan.co.kr/life/health/article/201403272114195> >

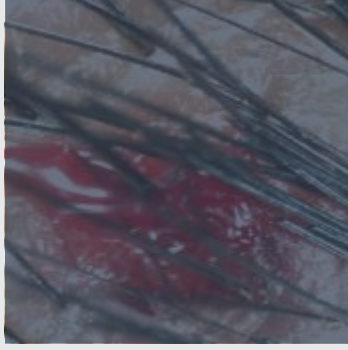
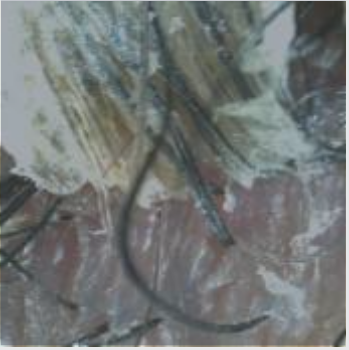
많은 사람들이  
**문제성 두피**를 경험하고 있다

# 다양한 두피 질환과 바람직한 해결책은?

## 다양한 두피 질환들



모낭사이홍반  
모낭홍반농포  
미세각질  
피지과다  
비듬  
탈모 ...



“두피 질환은 방치하면 탈모로 이어질 가능성이 높은 만큼 비듬과 가려움증, 과다 피지 분비 증상 외에도 두피 통증과 두피에서 열이 나는 증상, 머리카락이 가늘어지고 힘이 없어지는 증상이 나타나면 두피 건강을 진단받는 것이 좋다”

[https://m.health.chosun.com/svc/news\\_view.html?contid=2008021100767](https://m.health.chosun.com/svc/news_view.html?contid=2008021100767)  
강한피부과 강진수 원장 日

즉.

두피에 문제가 생긴다면  
정확한 병명과 현상을 파악하고  
그에 맞는 적절한 관리를 해야 한다

## 탈모에 관한 자가인식집단에 따른 두피관리행위의 차이

전문가와의 상담여부 두피 · 모발건강을 위해 전문가와 상담을 하는지에 대한 여부를 살펴보면

상담을 전혀 안한다는 48.1%로 응답자의 절반에 가까웠고,  
거의 안한다가 26.7%,  
보통이다가 17.6% 순서로 나타났다.

출처: 성인 남녀 탈모 자가인식에 의한 두피관리실태 연구  
<김미정, 정숙희, 심선녀 경남정보대학교 미용계열, 경동대학교 의료뷰티학과, 한국융합학회논문지 제3권 제3호, p24>

## 비듬 여부와 치료실태에 대한 설문조사

응답자(성인 1000명)의 약 40%가 비듬이 있거나  
지난 1년간 비듬 증상을 겪은 적이 있었다.

그들이 시도한 비듬 관리방법은

- 마트에서 판매하는 일반(비듬방지용) 두피샴푸 사용이 53.8%,
- 약국에서 판매하는 비듬치료 약용샴푸 사용이 24%,
- 병원에서 처방 받은 약용샴푸 사용이 16% 였다.

출처 : 의약뉴스(<http://www.newsmp.com>)  
한국존슨앤드존슨, 닐슨 리서치 성인 1000명을 대상으로 비듬 여부와 치료실태에 대한 설문조사

문제성 두피를 인식하더라도  
대부분 집에서 직접 진단, 관리 하고 있다

# 소비자들은 문제성 두피를 어떻게 관리하고 있을까?\_

## FGI



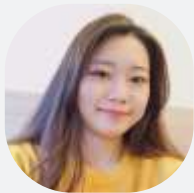
**김소현 양**  
경기 고양 거주 (26세, 여)

**Q:** 두피 문제로 고민해본 적 있으신가요?

**A:** 네, 작년에 집안일로 스트레스를 받아 원형탈모가 생긴 경험이 있어요.

**Q:** 당시 관리는 어떻게 하셨나요?병원은?

**A:** 원형탈모가 생겼을 때, 제가 외국에 있을 때라서 병원에 가지도 못하고 혼자서 어떻게든 해결해보려고 했던 것 같아요. 온갖 유튜브랑, 블로그를 찾아보고 했던 것 같아요.



**노재희 양**  
캐나다 토론토 거주 (30세, 여)

**Q:** 두피 문제로 고민해본 적 있으신가요?

**A:** 음... 머리에서 각질인지 비듬인지 자꾸 떨어져서 스트레스 받았던 적 있어요. 자꾸 어깨 위를 확인하게 되고 검은 색 옷은 기피하게 되었던 것 같아요.

**Q:** 당시 관리는 어떻게 하셨나요?병원은?

**A:** 당연히 비듬인 줄 알고 샴푸부터 비듬 샴푸로 바꾸고 머리 감고 잘 말리려고 노력했던 것 같아요. 단순히 비듬으로 병원을 가긴 좀 그렇죠.



**이준석 씨**  
경남 울산 거주 (34세, 남)

**Q:** 두피 문제로 고민해본 적 있으신가요?

**A:** 역시 탈모가 가장 고민이죠. 아버지가 탈모셨고, 요즘 머리가 많이 빠지는 느낌이 들어서 불안해요. 남자는 머리 스타일이 외모의 절반인데...

**Q:** 관리는 어떻게 하고 계신가요?병원은?

**A:** 파마는 생각도 못하고, 탈모 샴푸를 쓰는 정도 예요. 지금 일과 육아에 바빠서 병원까지는 생각 못하고 있어요.

**일로 바쁘거나,  
두피 문제를 가볍게 보거나,  
병원 진단을 받기 어려운 상황**

**때문에 샴푸, 민간요법 등으로 상태 호전을 기대하고 있었다**

일로 바쁘거나,  
두피 문제를 가볍게 보거나,  
병원 진단을 받기 어려운 상황 때문에...  
제대로 된 진단과 해결책을 받지 못하고  
문제성 두피 문제로 계속 고민하는 사람들을  
위해

+

집에서도 간편하게 할 수  
있는  
두피 상태 분석 서비스

Dupi.

Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 2. Dupi.Dr 프로젝트 소개





## 김민교

**Project Manager**

기획 및 프로젝트 프로세스 관리



## 노상희

**Data Analyst**

데이터 및 사용자 요구 분석



## 박기정

**Deep Learning Engineer**

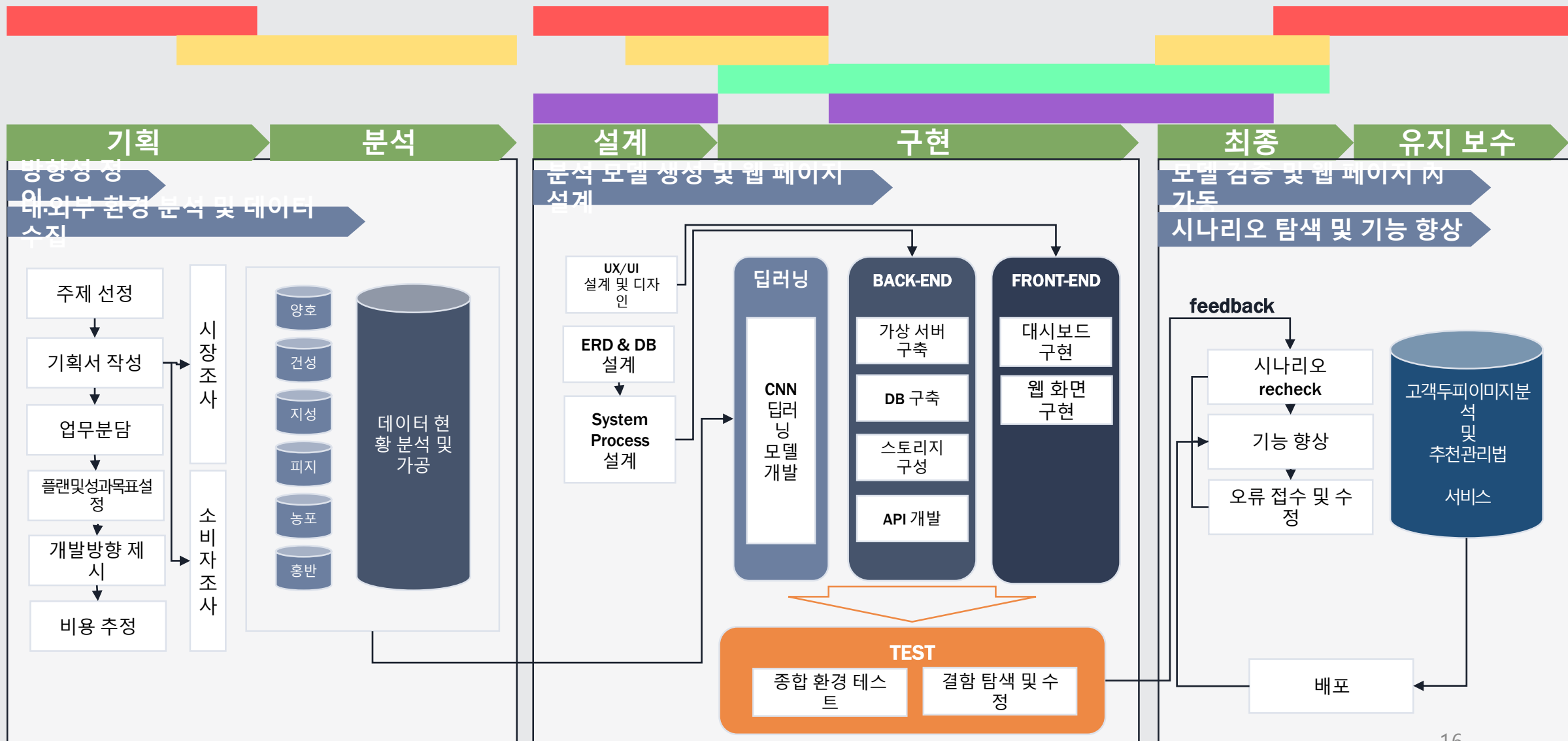
CNN 모델 구축 및 엔지니어링



## 권한준

**Web Developer**

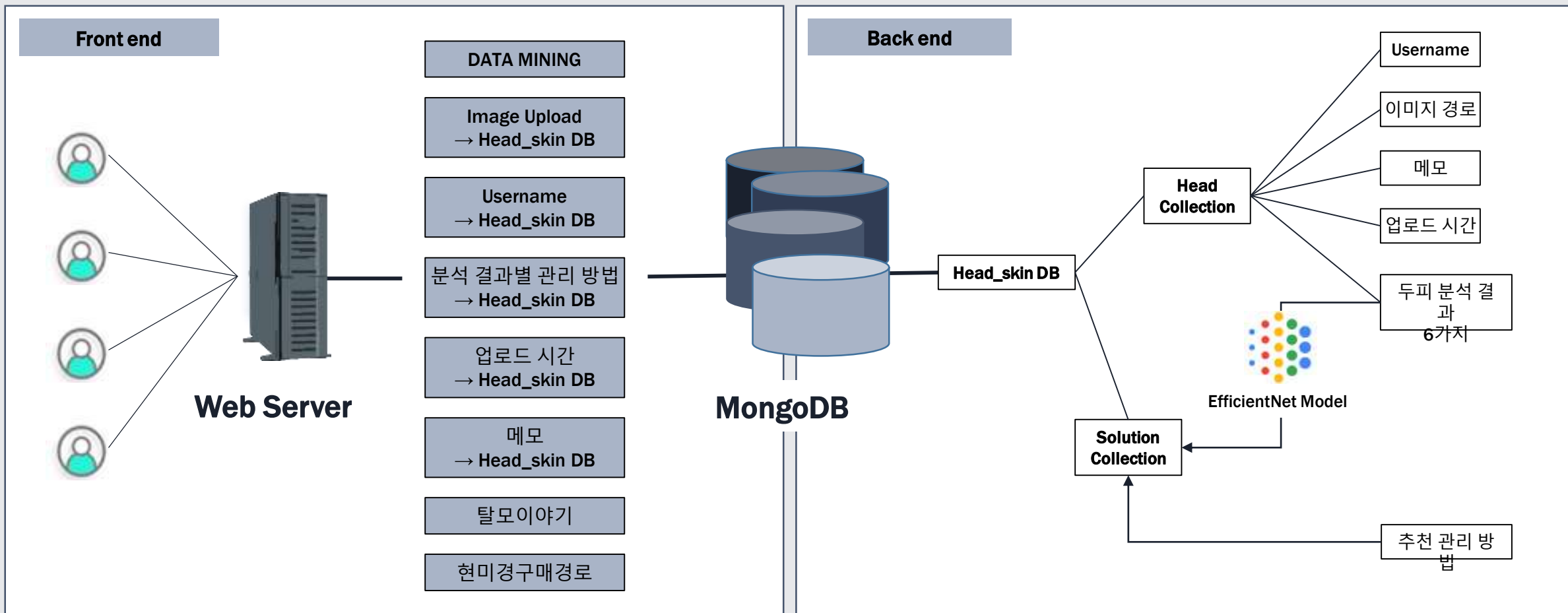
프론트엔드, 백엔드

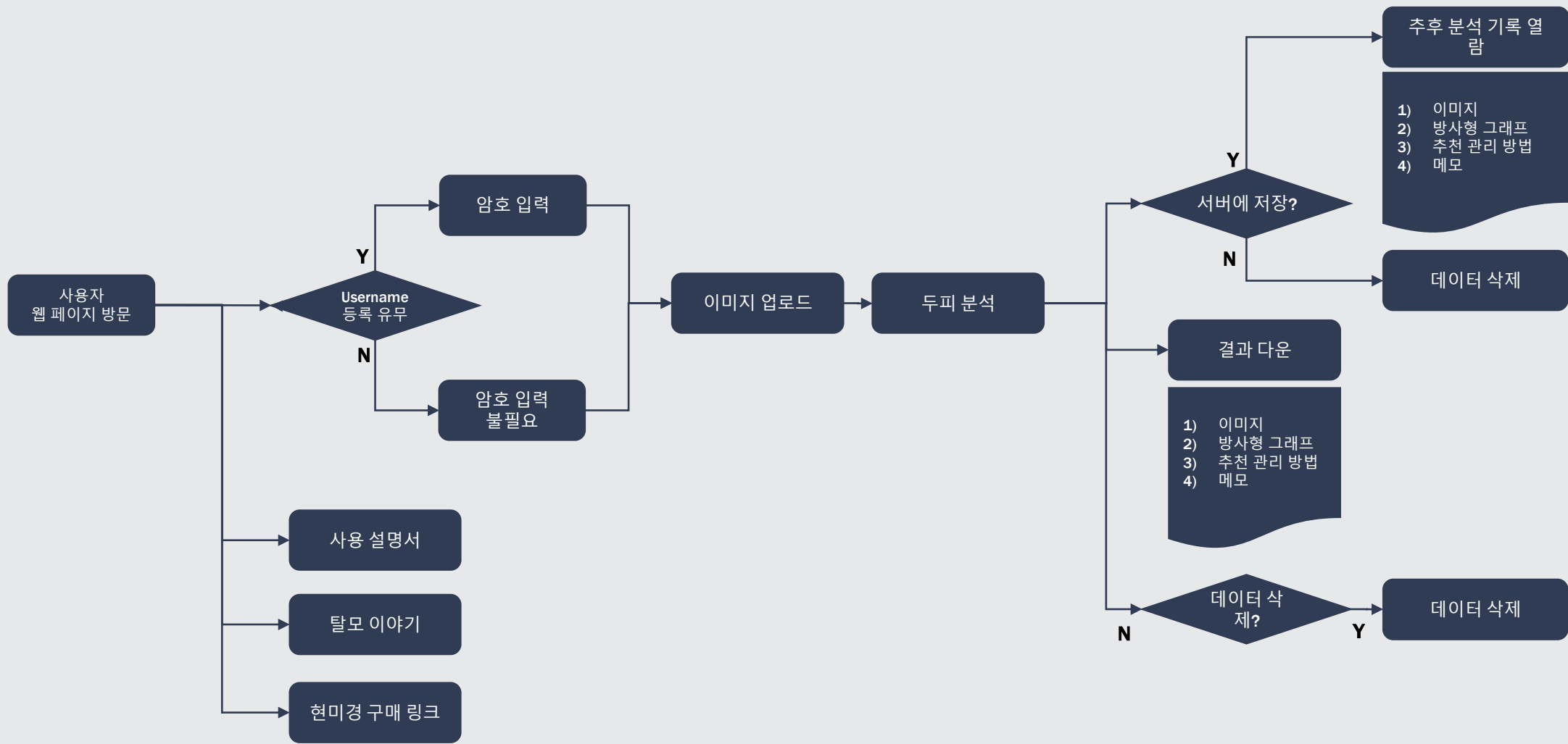


요구사항ID	요구사항 명	기능ID	기능명	세부사항	비고
A01	두피분석 서비스	A01-B01	두피 상태 분석	유형별 두피 이미지 확보	
				이미지 전처리	
				두피 이미지 분류에 가장 적합한 CNN모델 구축	
				서비스를 위한 웹 구축	
				서버에서 학습된 모델 실행	
		A01-B02	두피 분석 기록 열람	분석 결과 저장을 위한 DB구축	
		A01-B03	두피분석결과 인쇄	분석 결과 로컬에 스크린샷 저장으로 구현	
		A01-B04	두피 분석 웹 사용법	웹페이지 탭에 사용법 추가	



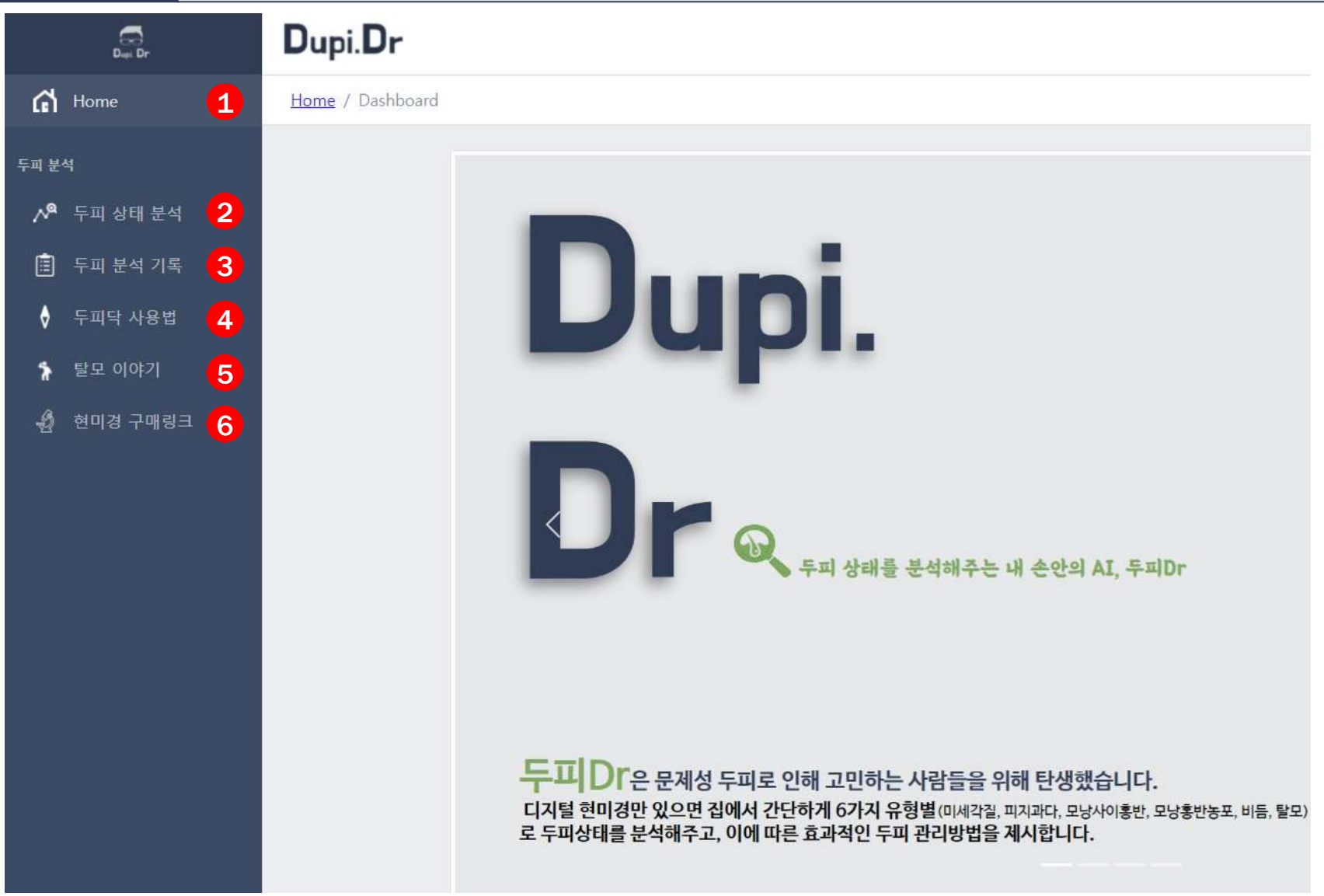
# 서비스 구상도







## 화면이름 Home



Description	
1	Main Page로 두피닥에 대한 간략한 소개 페이지
2	두피사진을 업로드하여 상태를 분석 받는 페이지
3	두피 사진을 통해 진단 기록이 있다면 기록 열람이 가능한 페이지
4	두피닥을 사용하기 위한 설명서 페이지
5	탈모에 관련한 속설들을 정리해 놓은 페이지
6	두피닥을 이용하기 위한 현미경 구매 링크
7	
8	
9	
10	

## 화면이름 두피 상태 분석 페이지 두피 상태 분석

### 두피 분석

사진은 60배율 현미경으로 밝고 선명한 사진을 넣었을 때 가장 정확한 결과를 얻을 수 있습니다. 결과가 예상과 다르게 나오는 경우 초점이나 밝기 해상도를 확인해주세요



1

test

2

4

Password

☐ 암호 사용 

3

암호 사용시에만 비공개가 됩니다.

5

파일 선택

0013\_A2LEBJJDE0006...6550773142\_6\_BH.jpg

6

7

33%


Description	
1	기록 열람에 사용할 사용자 이름 입력 칸
2	기록에 있는 사용자 아이디 중복 확인 버튼
3	체크 유무에 따라 비밀번호 사용 미사용 선택
4	비밀번호 입력 칸
5	사용자가 찍은 두피 이미지 업로드 창 열기 버튼
6	두피 분석을 진행하기 위해 분석 실행 버튼
7	6가지 유형별 분석 진행사항 확인을 위한 프로세스 바
8	
9	
10	

## 화면이름 두피 상태 분석 결과 페이지

test의 결과 삭제 서버에 저장 결과 다운

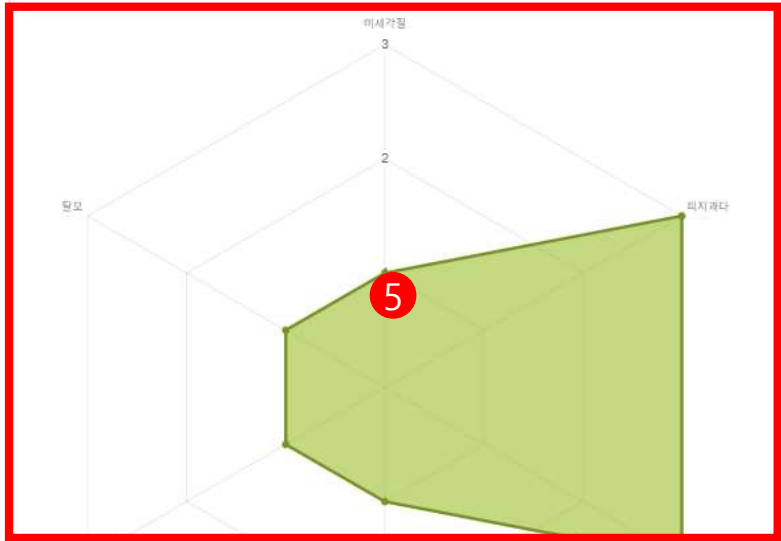
1 2 3

저장에 저장하지 않으면 기록이 남지 않습니다.



4

test의 두피



### 유형별 관리 방법

#### 미세각질

머리 두피 각질 원인은 나쁜 화학용품의 접촉이나, 신체의 열, 노화 등으로 인해 시작된다. 머리 두피 각질을 계속 방지하게 될 경우 두피의 모근과 모낭의 성장을 방해하고 심한 경우 탈모증상까지 초래할 수 있다. 따라서 꾸준한 두피관리로 머리 각질제거에 신경을 써야 한다.

머리 두피 각질을 위한 두피관리 방법으로는 천연샴푸 또는 두피 각질제거 샴푸를 사용하여 머리를 감는 방법이 있다. 또한 두피의 수분을 증가시켜 줄 수 있는 습관을 갖는 것도 필요하다.

좋은 두피관리 방법으로는 평소 스트레스를 해소시킬 수 있는 생활습관을 갖는 것과 두피에 좋은 음식으로 올바른 식습관을 유지하는 것이 있다. 두 가지 모두 두피각질 예방에 좋다.

출처: <https://www.hankyung.com/news/article/201311152777q>

#### 피지과다

피지가 많은 사람은 모발의 청결을 유지하기 위해 머리를 감을 때 비누보다 샴푸를 사용하

### 추천 관리 방법

#### [비듬성]

\* 정의  
비듬균의 이상증식 및 두피 각질층, 내모근초의 탈락 등에 의해 발생하는 두피유형으로 크게 비듬증상의 원인에 따라 자연적이나 생리현상에 의해 발생하는 비듬유형과 외적, 내적 이상현상에 의해 비듬균이 이상 증식하여 발생하는 '이상비듬'으로 구분되어 진다. 또한 비듬의 형태에 따른 분류는 노화된 각질이 정상적으로 각화작용을 하지 않고 저조한 신진대사장애로 수분결여로 생기는 건성비듬, 피지선의 과다로 표피의 각질층의 박리에 의해 생기는 지성비듬이 있다.

증상/상태

- 건성비듬 : 피지분비 장애가 심하게 나타나며, 호르몬, 영양 불균형 등 내적요인의 작용으로 피지 분비량이 감소하여 전체적으로 두피의 피지량이 정상두피의 절반정도 밖에 미치지 못하는 상태를 나타내고 있다. 또한 분비되는 피지의 상태에 있어서도 알칼리성 화합제품, 드라이, 외부 온도변화, 모발 및 두피에 대한 잦은 시술 등의 작용으로 인하여 제거되거나 혹은 건조되는 상태를 나타낸다.
- 지성비듬 : 피지선의 왕성한 분비작용으로 인해 두피에 끈적임이 시만 피지가 형성되어있는 상태로 두피 전체에서 산화된 피지물과 새로이 분비되는 피지를 쉽게 확인할 수 있다. 산화된 피지물이 두피자극을 시켜서 두피와 모근 주변에서 산화 및 발진 현상을 볼 수 있으며

Description	
1	분석한 기록을 저장하지 않고 삭제하는 버튼
2	분석한 내용과 두피 이미지를 추후에 열람하기 위해 서버에 저장하는 버튼
3	분석 내용을 로컬에 스크린 샷 형태로 저장하는 버튼
4	사용자가 분석을 위해 업로드한 사진
5	6가지 유형별 증상의 정도를 방사형 그래프로 시각화
6	6가지 유형별 증상 유무에 따라 나타나는 관리 방법
7	6가지 유형의 증상 유무에 따라 추천 관리방법 제시
8	
9	
10	

화면이름 두피 분석 기록 열람 페이지

두피 분석 기록

## 두피 분석 기록 검색

분석시 사용한 이름을 입력해주세요

test11

1

검색2

test11의 결과

뒤로4

1. 2021-08-31 10:08:38.543

▽

Description	
1	사용자가 두피 분석시 사용했던 이름 입력
2	사용자에게 해당하는 기록 검색 실행 버튼 (패스워드를 설정했다면 입력 팝업이 뜬다.)
3	검색을 누른 후 나오는 일자 별 분석결과기록
4	다시 두피 분석 기록 검색 창으로 가는 뒤로 버튼
5	
6	
7	
8	
9	
10	

1

## Dupi.Dr 사용설명서

### ◆ HOME

#### I 두피 상태 분석

- 사용자 등록
- 이미지 업로드
- 분석 기록 저장
- 분석 기록 다운로드

#### II 두피 분석 기록

- 분석 기록 열람

#### III 탈모 이야기

#### IV 현미경 구매 안내

### ◆ HOME

Dupi.Dr 메인 화면 구경하기



## Description

1 두피닥 사용 설명서

2

3

4

5

6

7

8

9

10

1

대머리는 유전이 된다?

^

2

대머리가 되는 유전인자라고 하는 것은 크게 두 가지로 나누어 볼 수 있는데, 그 첫 번째가 대머리와 직접적으로 연관있는 인자 (5알파-리닥타제의 활성도 및 남성 호르몬에 대한 감수성 등을 결정짓는 인자)가 유전이 되는 것과 대머리가 되기 쉬운 체질 및 성격, Life Style 등이 유전이 되는 것이다.

두 번째 요인으로 들 수 있는 대머리가 되기 쉬운 체질 및 Life Style 등은 후천적인 원인으로 본인의 의지에 따라 변화가 가능한 부분이며, 탈모 및 대머리를 가속화시키는 보조 역할의 작용을 한다.

3

찾은 샴푸는 탈모를 촉진시킨다?

두피를 자주 두드리면 머리가 난다?

모발(베넷모)을 한번 식발하면 다시 자라는 머리카락이 증가한다?

원형탈모는 관리를 하지 않아도 회복된다?

머리카락이 길면 탈모되기가 쉽다?

검은깨, 미역, 다시마, 콩 등의 식품이 모발성장에 좋다?

대머리는 정력이 세다?

모자나 가발을 착용하면 대머리가 된다?

모발의 굵기가 가는 사람은 탈모(대머리)가 되기 쉽다?

두피의 움직임이(혈행 원활) 좋으면 탈모가 되지 않는다?

육모제, 양모제를 사용하면 모발이 자란다?

Description	
1	궁금한 질문을 클릭
2	해당 질문에 대한 답
3	
4	
5	
6	
7	
8	
9	
10	



## 화면이름 현미경 구매링크 페이지 현미경 구매링크

다목적 60 배 현미경 보석 및 옥 식별 휴대용 미니 고출력 카메라 전화 돋보기

**29,200 원**

최대 1,460원 적립

무료배송  
10/16 도착 예정

판매자: 영보사물 [판매자 상품 보러가기](#)  
택배사: 롯데택배

**1** 배송지|배송|색상: CHINA|60X|은

캐시적립 혜택 ① | 최대 1,460원 적립

쿠팡이머니 결제 시 1% 적립

[로켓와우 + 쿠팡이 계좌이체] 결제 시 2% 적립

[로켓와우 + 쿠팡이머니] 결제 시 4% 추가적립

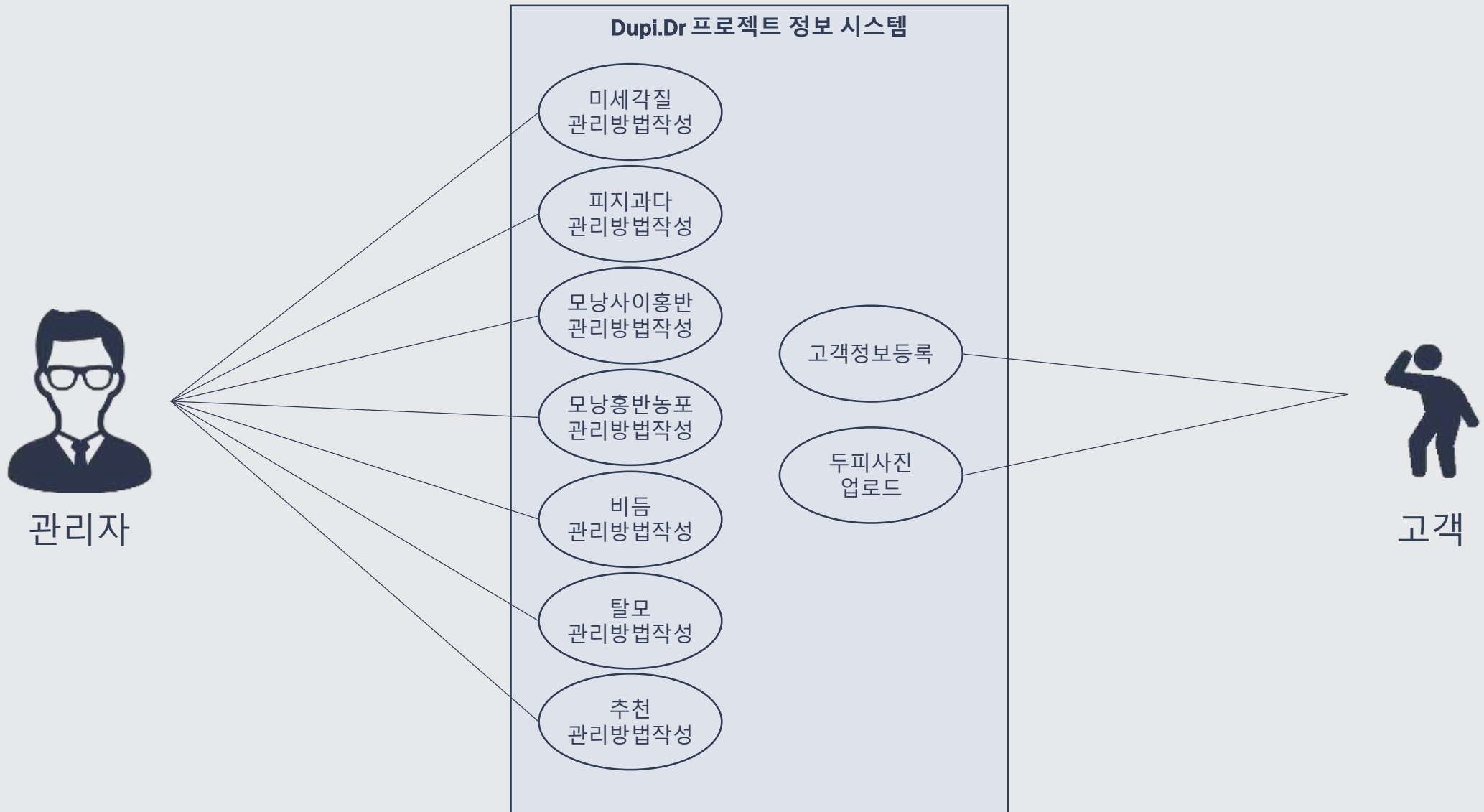
[로켓와우 무료체험 신청하기](#)

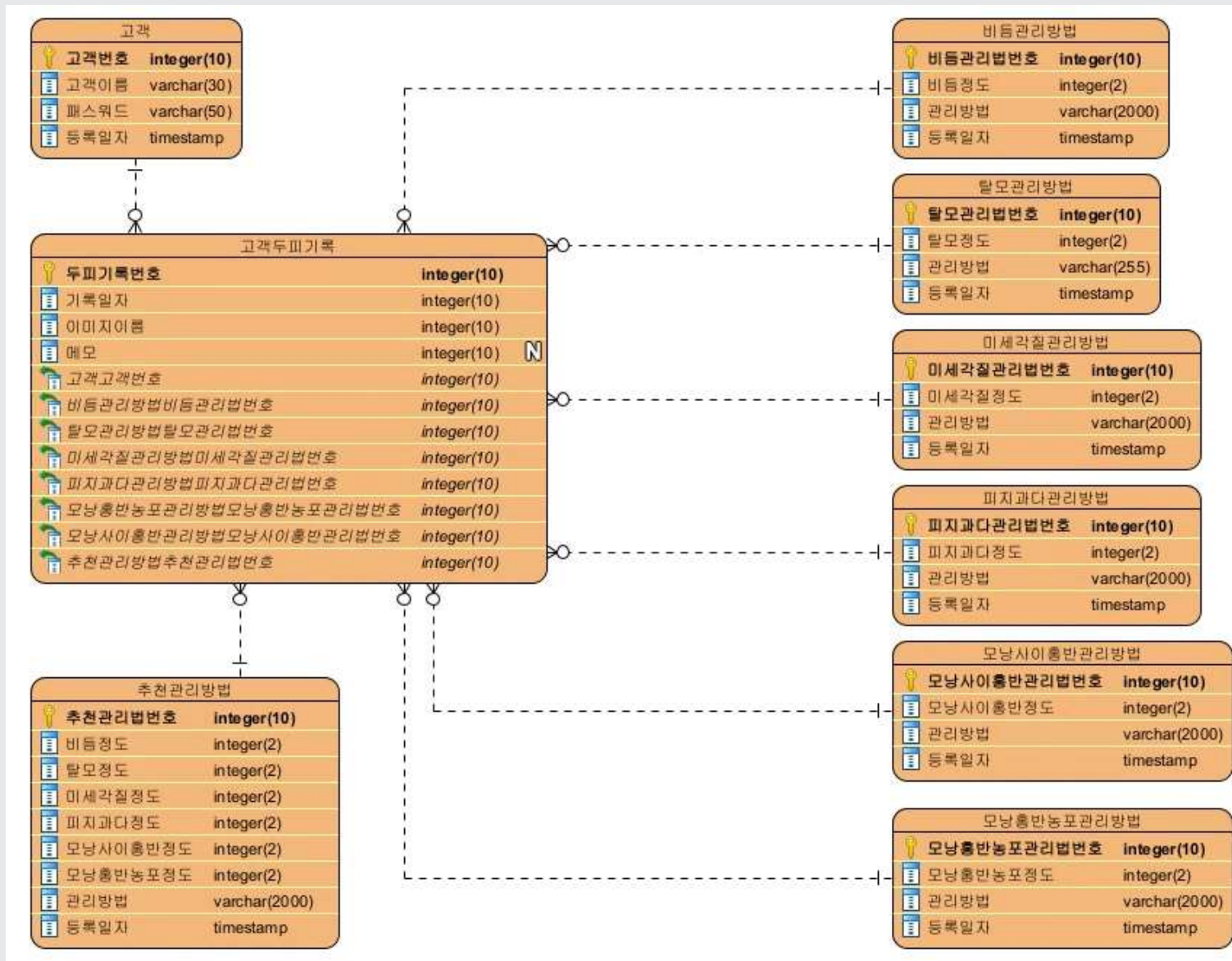
1

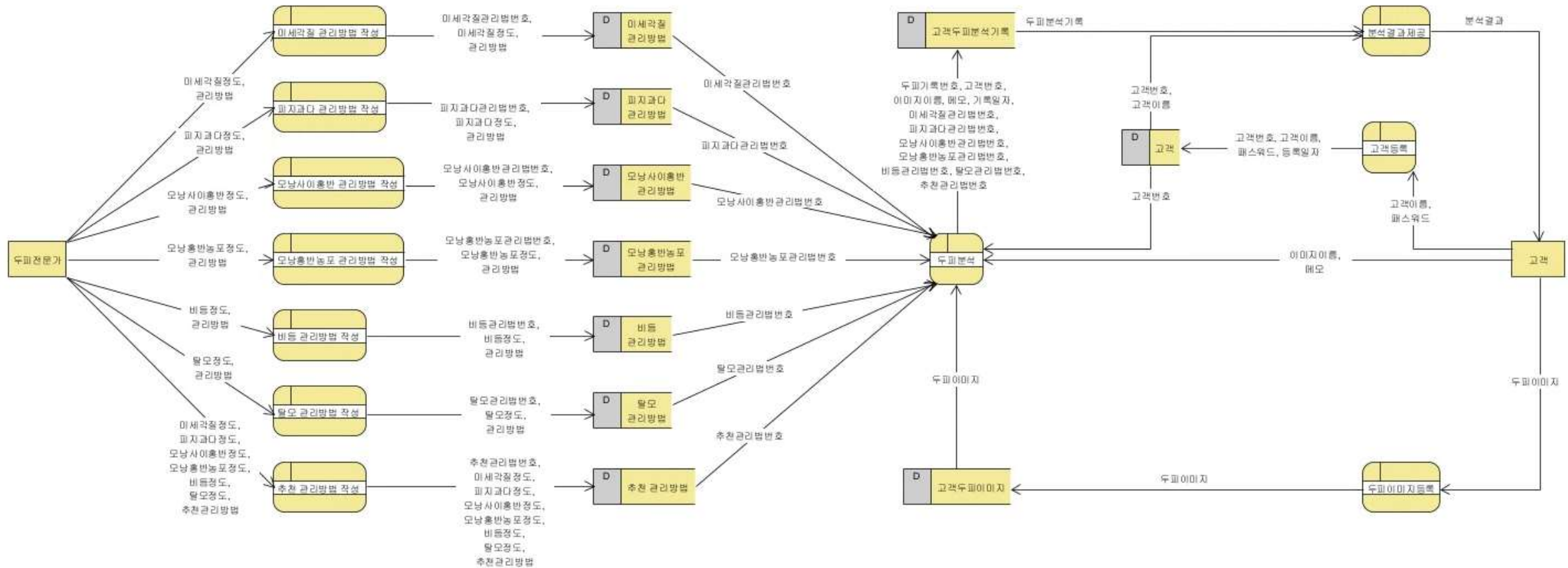
[장바구니 담기](#)

[바로구매 >](#)

Description	
1	두피 이미지를 찍기 위한 현미경 구매 링크 결과 쇼핑몰 페이지
2	
3	
4	
5	
6	
7	
8	
9	
10	







<https://aihub.or.kr/aidata/30758>

구분	이미지	미세각질	피지과다	모낭사이 홍반	모낭홍반/ 농포	비듬	탈모
양호 (정상)		-	-	-	-	-	-
건성		+	-	-	-	-	-
지성		-	+	-	-	-	-
민감성		+ -	-	+	-	-	-
지루성		+ -	+	+	-	+ -	-
염증성		+ -	+ -	-	+	+ -	-
피염성		+ -	+ -	-	-	+	-
탈모성		-	-	-	-	-	+

※ "+": 해당 증상이 있음 / "-": 해당 증상이 없음 / "+-": 해당 증상이 있거나 없을 수 있음

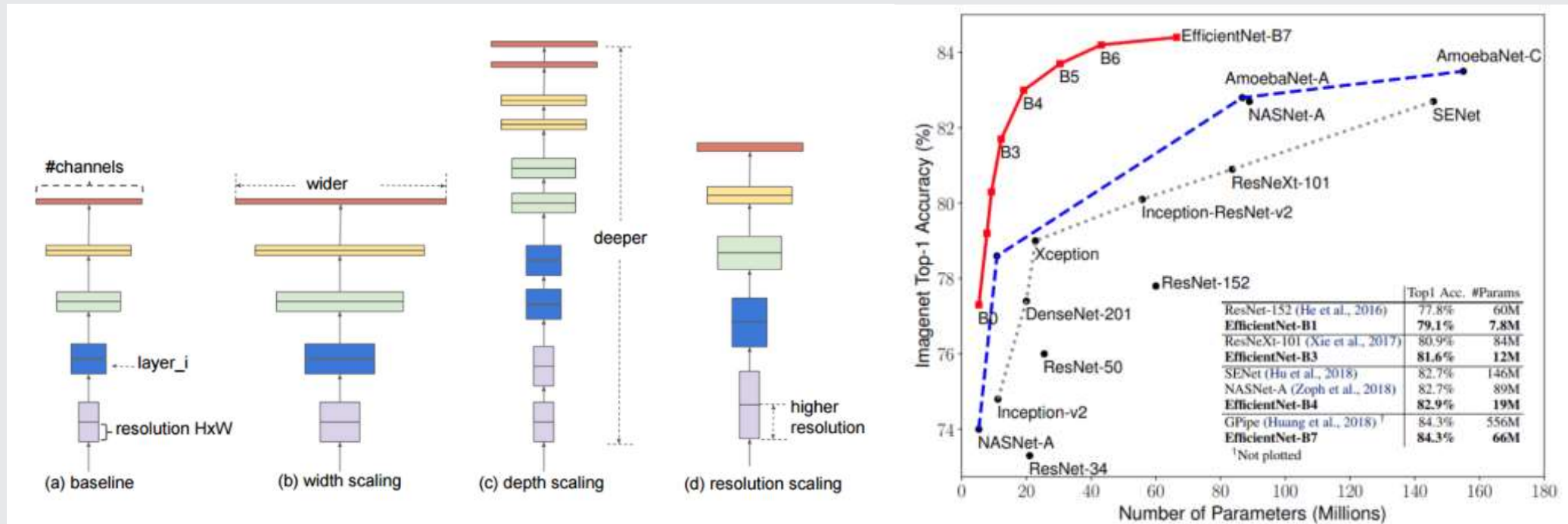
8가지 유형(양호, 건성, 지성, 민감성, 지루성, 염증성, 비듬성, 탈모성)의 두피 상태 이미지는 데이터의 이미지 특성 측면에서

두피 질환 6가지 유형(미세각질, 피지과다, 모낭사이홍반, 모낭홍반농포, 비듬, 탈모)의  
주요 특징 조합으로 도출해낼 수 있다.

<https://aihub.or.kr/aidata/30758>

Dupi.Dr의 두피 분석 서비스를 통해 각 두피 질환 유형별 정도(양호, 경증, 중등도, 중증)를 분석하고, 왼쪽 조합표에 따라 각 **유형별 맞춤 관리방법을 제시**한다.





## 1. 추론이 작고 빨라 현재 개발환경에 적합하다.

당사의 **EfficientNet-B7**은 ImageNet에서 **84.3%**의 최첨단 상위 **1위** 정확도를 달성하며 기존 **ConvNet**보다 추론이 **8.4배** 작고 **6.1배** 빠르다. 또한 **EfficientNets**는 전송이 잘되며 **CIFAR-100(91.7%)**, **꽃(98.8%)**, 기타 **3가지** 전송 학습 데이터 세트에서 최첨단 정확도를 달성하며 매개 변수는 훨씬 적다.

## 2. 정확도가 높다.

널리 사용되는 **ResNet-50**과 비교했을 때, **EfficientNet-B4**는 유사한 **FLOPS**를 사용하는 반면, 상위 **1위** 정확도는 **ResNet-50**의 **76.3%**에서 **82.6%(+6.3%)**로 향상되었습니다.



## 서버 사양

- Ubuntu18.04LST
- Architecture: ARMv8
- Node: 14.17.2 NPM: 6.14.13
- MongoDB: 3.6.3
- Node module list
- multer 1.4.3 (image upload)
- bcrypt 5.0.1 (암호화)
- mongoose 5.13.3 (mongoDB 연결)
- axios 0.21.1(웹에 post / get 요청 보낼 때 쓰는 모듈)
- express 4.17.1 (서버 구성하는 모듈)

## React module List

- coreui 4.0.0 (템플릿 버전)
- react 17.0.2 (react 자체 버전)

## Language

- Python
- mongoDB
- React
- node.js

## Library

- Tensorflow 2.3.0
- Keras 2.4.0
- Numpy 1.19.0
- Matplotlib 3.4.2
- opencv-python 4.5.3
- Pillow 8.3.1

## DataBase

- Pymongo

## Device

- Computer 3대
  - 한준 GPU\_ RTX2060s
  - 기정 GPU\_ RTX3070
  - 민교 GPU\_ GTX1660s
- Corab Pro 계정 3개

## 두피Dr 프로젝트 추정 예산

인건비	직책	인원수	시간당 임금	일 평균 근로시간	근로 일 수	투입율	총 근로 시간	임금	
	IT 기획자	1	48,591	8	20	100%	160	₩7,774,560	
	데이터분석가	1	43,459	8	20	100%	160	₩6,953,440	
	시스템SW 개발자	1	31,631	8	20	100%	160	₩5,060,960	
	응용 SW 개발자	1	40,397	8	20	100%	160	₩6,463,520	
	총합								₩26,252,480
간접경비	항목								비용
	숙박비								₩1,200,000
	교통비								₩390,400
	간식비								₩600,000
	총합								₩2,190,400
	H/W 구매비용	항목							
디지털 현미경								₩29,200	
서버								₩100,000	
노트북 4대								₩4,800,000	
총합								₩4,929,200	
S/W 구매비용		항목							
	colab pro								₩39,033
	총합								₩39,033
	총 프로젝트 비용								₩33,411,113

# Dupi. Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 3. Dupi.Dr 기대효과

## • 교육적 기대효과

두피Dr은 사용자에게 객관적인 두피 분석을 제시하고, 분석 결과에 따른 적절한 관리 방법을 제공하며, 분석 결과를 기록할 수 있다. 이러한 서비스는 사용자에게 두피 상태별 관리 방법 학습을 기대할 수 있으며, 과거 기록으로부터 호전, 악화 등의 가시적인 피드백을 받아 지속적인 관리를 기대할 수 있다.

## • 사회적 기대효과

기존에 시장에 나와 있는 두피 분석 서비스는 한 회사의 서비스를 광고하기 위해 만들어져 다양한 사용자가 두피 분석 서비스를 경험하지 못하는 한계점이 존재했다. 또한 분석 결과가 탈모에 치중하거나 현미경 사진이 아닌 '헤어라인'만을 수집하여 정확한 분석 결과를 기대하기 어려웠다.

본 프로젝트는 인터넷에서 쉽게 구매 가능한 디지털 현미경으로 누구나 자신의 두피 분석 결과에 접근할 수 있다. 이는 일반인의 의학적 지식에 대한 접근성 편의를 기대할 수 있으며, 두피 건강으로 인해 고통받는 국민의 국민건강에 기여할 수 있을 것으로 기대된다.

## • 산업적 기대효과

두피 상태를 6가지로 구분하여 분석한 후, 맞춤 제품을 추천하여 소비자 구매를 유도할 수 있다. 이는 탈모에만 초점이 맞춰져 있는 타 서비스와 차별점으로, 두피 질환에서도 특히 염증 관련 제품 산업군의 성장을 도모할 수 있다.

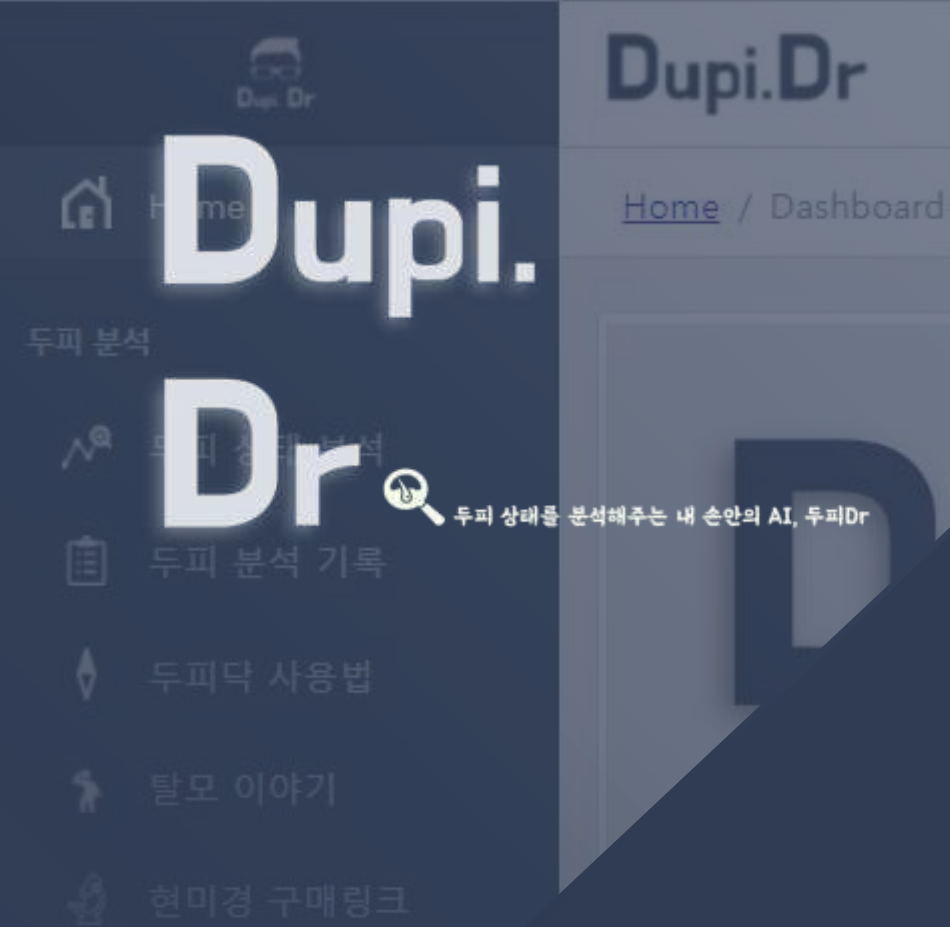
### -8가지 두피 유형으로 분류

소비자가 집에서 간편하게 자신의 두피 유형을 분석 받고, 그에 맞는 샴푸, 린스 등의 두피케어 상품 판매에 효과가 있을 것으로 기대한다.

## • 경제적 기대효과

### -소비자의 두피 치료를 위한 의료비 거액 지출 예방 효과

분석 결과를 참고하여 피부과 방문의 필요성을 미리 인지하고, 두피 관련 큰 질환이 발병하기 전에 가정에서 조치를 취할 수 있다. 따라서, 의료비의 전체 지출액을 아낄 수 있을 것으로 기대하고 있다.



## 4. Dupi.Dr 실사

Dupi.Dr

백메이커 교육 6 TEAM

Home

[Home](#) / [Search](#) / [Analysis](#)

두피 분석

두피 상태 분석

두피 분석 기록

두피당 사용법

탈모 이야기

변미광 구매링크

## 두피 분석

사진은 60배를 현미경으로 밝고 선명한 사진을 놓았을 때 가장 정확한 결과를 얻을 수 있습니다. 결과가 예상과 다르게 나오는 경우 초점이나 밝기 해상도를 확인해주세요.




Username

Password

☐ 암호 사용  
암호 사용시에만 비공개가 됩니다.

파일 선택  선택된 파일 없음



Home

두피 분석


- 두피 상태 분석
- 두피 분석 기록
- 두피당 사용법
- 탈모 이야기
- 현미경 구매링크

## Dupi.Dr

[Home](#) / [Search](#) / [Analysis](#)

### 두피 분석

사진은 60배율 현미경으로 밝고 선명한 사진을 놓았을 때 가장 정확한 결과를 얻을 수 있습니다. 결과가 예상과 다르게 나오는 경우 초점이나 밝기 해상도를 확인해주세요.



☐ 암호 사용  
 암호 사용시에만 비공개가 됩니다.

## 데이터셋 제공자의 모델 정확도

다만, 제한된 데이터 수량으로 인해 정확도 제고 및 활용에 한계성을 지니고 있었다.



그 결과, 진단 정확도가 70%를 하회하는 수준이 머물러 두피 이미지에 대한 AI 진단 정확도를 제고하고 경쟁력 있는 제품으로 업그레이드하는 과업을 계속 진행하고 있다.

[https://aihub.or.kr/sites/default/files/Sample\\_data/%EA%B5%AC%EC%B6%95%ED%99%9C%EC%9A%A9%EA%B0%80%EC%9D%B4%EB%93%9C%EB%B6%81\\_2020-02/148.%EC%9C%A0%ED%98%95%EB%B3%84\\_%EB%91%90%ED%94%BC\\_%EC%9D%B4%EB%AF%B8%EC%A7%80\\_%EB%8D%B0%EC%9D%B4%ED%84%B0\\_%EA%B5%AC%EC%B6%95\\_%EA%B0%80%EC%9D%B4%EB%93%9C%EB%9D%BC%EC%9D%B8.pdf](https://aihub.or.kr/sites/default/files/Sample_data/%EA%B5%AC%EC%B6%95%ED%99%9C%EC%9A%A9%EA%B0%80%EC%9D%B4%EB%93%9C%EB%B6%81_2020-02/148.%EC%9C%A0%ED%98%95%EB%B3%84_%EB%91%90%ED%94%BC_%EC%9D%B4%EB%AF%B8%EC%A7%80_%EB%8D%B0%EC%9D%B4%ED%84%B0_%EA%B5%AC%EC%B6%95_%EA%B0%80%EC%9D%B4%EB%93%9C%EB%9D%BC%EC%9D%B8.pdf)

## 두피Dr의 testset 모델 정확도

```
Found 8248 images belonging to 4 classes.

===== bideum_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.69352
=====

Found 13635 images belonging to 4 classes.

===== hongban_effB4.hdf5 Evaluate =====
acc: 80.00%
loss: 0.52033
=====

Found 3639 images belonging to 4 classes.

===== misae_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.63593
=====

Found 1070 images belonging to 4 classes.

===== nongpo_effB4.hdf5 Evaluate =====
acc: 70.00%
loss: 0.91290
=====

Found 16236 images belonging to 4 classes.

===== pijji_effB4.hdf5 Evaluate =====
acc: 66.67%
loss: 0.80835
=====

Found 5288 images belonging to 4 classes.

===== talmo_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.96061
=====
```





# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 5. Reference

## Table 목록

NO	한글명	영문명	개요	비고
1	고객두피기록	user_dupi_record	고객두피 분석 기록	
2	고객	gogek	고객정보	
3	미세각질관리방법	mise_manage	미세각질 관리방법 정보	
4	피지과다관리방법	pigi_manage	피지과다 관리방법 정보	
5	모낭홍반농포관리방법	nongpo_manage	모낭홍반농포 관리방법 정보	
6	모낭사이홍반관리방법	hongban_manage	모낭사이홍반 관리방법 정보	
7	비듬관리방법	bidum_manage	비듬 관리방법 정보	
8	탈모관리방법	talmo_manage	탈모 관리방법 정보	
9	추천관리방법	recommend_manage	두피 유형별 추천 관리방법 정보	

## Primary Key 목록

NO	PK	Table	컬럼명	비고
1	record_no	user_dupi_record	record_no	
2	gogek_no	gogek	gogek_no	
3	mise_manage_no	mise_manage	mise_manage_no	
4	pigi_manage_no	pigi_manage	pigi_manage_no	
5	nongpo_manage_no	nongpo_manage	nongpo_manage_no	
6	hongban_manage_no	hongban_manage	hongban_manage_no	
7	bidum_manage_no	bidum_manage	bidum_manage_no	
8	talmo_manage_no	talmo_manage	talmo_manage_no	
9	recommend_manage_no	recommend_manage	recommand_manage_no	

## Foreign Key 목록

NO	FK	관련 Table	컬럼명	비고
1	gogek_no	user_dupi_record	gogek_no	
2	mise_manage_no	user_dupi_record	mise_manage_no	
3	pigi_manage_no	user_dupi_record	pigi_manage_no	
4	nongpo_manage_no	user_dupi_record	nongpo_manage_no	
5	hongban_manage_no	user_dupi_record	hongban_manage_no	
6	bidum_manage_no	user_dupi_record	bidum_manage_no	
7	talmo_manage_no	user_dupi_record	talmo_manage_no	
8	recommend_manage_no	user_dupi_record	recommend_manage_no	

테이블한글명	고객두피기록		테이블영문명		user_dupi_record		
개요	고객두피 분석 기록						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
record_no	두피기록번호	Yes	No	Yes	Intiger	10	
image_name	이미지이름		No	No	Varchar	255	
memo	메모		Yes	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	record_no						
FK	mise_manage_no						
	pigi_manage_no						
	nongpo_manage_no						
	hongban_manage_no						
	bidum_manage_no						
	talmo_manage_no						
Trigger							
참고사항							

테이블한글명	추천관리방법	테이블영문명			recommend_manage		
개요	두피 유형별 추천 관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
recommend_manage_no	탈모관리법 번호	Yes	No	Yes	Intiger	10	
mise_degree	미세각질정도		No	No	Varchar	2	
pigi_degree	피지과다정도		No	No	Varchar	2	
nongpo_degree	모낭홍반농포정도		No	No	Varchar	2	
hongban_degree	모낭사이홍반정도		No	No	Varchar	2	
bidum_degree	비듬정도		No	No	Varchar	2	
talmo_degree	탈모정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	recommend_manage_no						
FK							
Trigger							
참고사항							

테이블한글명	고객			테이블영문명		gogek	
개요	고객 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
gogek_no	고객번호	Yes	No	Yes	Intiger	10	
gogek_name	고객이름		No	No	Varchar	30	
pw	패스워드		No	No	Varchar	50	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	gogek_no						
FK							
Trigger							
참고사항							

테이블한글명	미세각질관리방법			테이블영문명		mise_manage	
개요	미세각질관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
mise_manage_no	미세각질관리법번호	Yes	No	Yes	Intiger	10	
mise_degree	미세각질정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	mise_manage_no						
FK							
Trigger							
참고사항							

테이블 한글명	피지과다관리방법			테이블영문명		pigi_manage	
개요	피지과다 관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
pigi_manage_no	피지과다관리 법번호	Yes	No	Yes	Intiger	10	
pigi_degree	피지과다정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	pigi_manage_no						
FK							
Trigger							
참고사항							

테이블한글명	모낭홍반농포관리방법			테이블영문명		nongpo_manage	
개요	모낭홍반농포관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
nongpo_manage_no	모낭홍반농포관리법 번호	Yes	No	Yes	Intiger	10	
nongpo_degree	모낭홍반농포 정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	nongpo_manage_no						
FK							
Trigger							
참고사항							

테이블 한글명	모낭사이홍반관리방법			테이블영문명		hongban_manage	
개요	모낭사이홍반관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
hongban_manage_no	모낭사이홍반관리법 번호	Yes	No	Yes	Intiger	10	
hongban_degree	모낭사이홍반 정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	hongban_manage_no						
FK							
Trigger							
참고사항							

테이블한글명	비듬관리방법			테이블영문명		bidum_manage	
개요	비듬 관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
bidum_manage_no	비듬관리법 번호	Yes	No	Yes	Intiger	10	
bidum_degree	비듬정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	bidum_manage_no						
FK							
Trigger							
참고사항							



테이블한글명	탈모관리방법			테이블영문명		talmo_manage	
개요	탈모 관리방법 정보						
소유자		보관주기	영구		위치		
Row 길이		초기건수			증가량(월)		
컬럼명	설명	KEY 구분	NULL	Unique	DataType	길이	비고
talmo_manage_no	탈모관리법 번호	Yes	No	Yes	Intiger	10	
talmo_degree	탈모정도		No	No	Varchar	2	
management	관리방법		No	No	Varchar	2000	
record_day	기록일자		No	No	Timestamp		
볼륨	Initial(Kb)		64		Next(Kb)	1024	
	DBSPACE명		Edge Device Jetson Nano				
PK	talmo_manage_no						
FK							
Trigger							
참고사항							

## Description

Remove hangul from image name

## 코드

```
import glob
import os

target_name = 'misegakgeal_inspection'
root_dir = os.path.join(os.getcwd(), target_name)

for train_val in ['training', 'validation']:
    img_paths = glob.glob(os.path.join(root_dir, train_val + '\*\*.jpg'))

    for idx, img_name in enumerate(img_paths):
        file_name = ''
        if img_name.find('(') != -1:
            file_name = '{}.jpg'.format(img_name[:img_name.find('(') - 1])
            os.rename(img_name, file_name)
        else:
            continue

    print(f'{idx}.', file_name, "hangul is deleted!!")
```

양호데이터 수가 부족하여 양호 데이터를 크롭하는 과정에서 이미지 파일 이름에 한글이 들어가 **openCV**에서 이미지를 불러오지 못하므로 한글을 지워주었다.

## Description

## Resize Image for EfficientNet\_B4

## 코드

```
import cv2
import glob
import os
from PIL import Image

classes = ['pi1', 'pi2', 'pi3']
root_dir = r'E:\datacourse_free\team_project\pi_dataset\training'
# 456 by 456 size is for efficientNetB5
# should be change depending on what efficientNet model u choose
TARGET_IMAGE_SIZE = (380, 380)

# B4모델은 380x380로 정해져있음

for class_name in classes:
    image_paths = glob.glob(root_dir + f'\{class_name}\*.jpg')

    for index, image_path in enumerate(image_paths):
        # resize image
        img = cv2.imread(image_path, cv2.IMREAD_COLOR)
        resized_img = cv2.resize(img, dsizes=TARGET_IMAGE_SIZE, interpolation=cv2.INTER_AREA)

        # get class_root_path, image_name
        class_path = image_path.split('\\')[-2]
        image_name = image_path.split('\\')[-1]

        # image save
        save_image_path = r'E:\datacourse_free\team_project\misegakgeal_inspection\training' + '\{0}\{1}'.format(class_path, image_name)
        cv2.imwrite(save_image_path, resized_img)
        print(f"{index}... -> {class_path} : {image_name} is saved!!!")
```

이미지 축소로 인한 해상도문제를 보간법으로 해결

## Description

## Change directory constructor to use ImageGenerator

## 코드

```
import os
import numpy as np
import shutil

def split_data(all_files, ratio=0.8):...

def train_validation_config(all_file_names, train_dataset, validation_dataset):...

def create_dir(train_path, validation_path, classes_list):...

RATIO = 0.8
root_dir = 'E:/datacourse_free/team_project'
dataset_name = 'pi_dataset'
dataset_path = os.path.join(root_dir, dataset_name)
classes_list = ['pi0', 'pi1', 'pi2', 'pi3']
paths0 = 'E:/datacourse_free/team_project/pi_dataset/pi3'
# 이 때 train, validation 폴더를 만들기 전에 Image폴더를 만들어준다.

train_path = dataset_path + r'\training'
validation_path = dataset_path + r'\validation'

create_dir(train_path, validation_path, classes_list)

for class_name in classes_list:
    images_path = dataset_path + f'\{class_name}'
    all_file_names = os.listdir(images_path)

    # split train, validation data
    train_dataset, validation_dataset = split_data(all_files=all_file_names, ratio=RATIO, show=True)

    # show ratio of total, train, validation
    train_validation_config(all_file_names, train_dataset, validation_dataset)

    # move image to another train, validation directory
    for idx, filename in enumerate(train_dataset):
        shutil.move(images_path + '/' + filename, train_path + f'/{class_name}' + f'/{filename}')
        print(idx, filename, 'moved!!!')
    for idx, filename in enumerate(validation_dataset):
        shutil.move(images_path + '/' + filename, validation_path + f'/{class_name}' + f'/{filename}')
        print(idx, filename, 'completed!!!')
```

```
def split_data(all_files, ratio=0.8):
    # shuffle the data
    np.random.shuffle(all_files)

    train_file_names = all_files[:int(len(all_files)*ratio)]
    validation_file_names = all_files[int(len(all_files)*ratio):]

    return train_file_names, validation_file_names

def create_dir(train_path, validation_path, classes_list):
    # if training, validation dir is not exist then create dir
    if not os.path.exists(train_path):
        os.mkdir(train_path)

    # create target dir
    for target in classes_list:
        if not os.path.exists(train_path + f'\{target}'):
            os.mkdir(train_path + f'\{target}')

    if not os.path.exists(validation_path):
        os.mkdir(validation_path)

    # create target dir
    for target in classes_list:
        if not os.path.exists(validation_path + f'\{target}'):
            os.mkdir(validation_path + f'\{target}')
```



## Description

Cropping the image data for increasing the number of image data

## 코드

```
import cv2
import glob
import os
import random

# define crop_image function
def crop_image(img):...

# efficient model B4 size 380
CROP_SIZE = 380
NUM_IMAGES_LIMIT = 2800
# root_dir = os.path.join(os.getcwd(), r'misegakgeal_inspection\training\mi0')
root_dir = os.path.join(os.getcwd(), r'misegakgeal_inspection\training\mi0')
images = glob.glob(root_dir + r'\*.jpg')
random.shuffle(images)

count = 0

for image in images:
    if len(os.listdir(root_dir)) > NUM_IMAGES_LIMIT:
        break
    count += 1
    img = cv2.imread(image, cv2.IMREAD_COLOR)
    result_cropped = crop_image(img)
    img_name = image.split('\\')[-1]
    class_name = image.split('\\')[-2]

    for idx, save_image in enumerate(result_cropped):
        cv2.imwrite(os.getcwd() + f'\\misegakgeal_inspection\\training\\crop\\{idx}_cropped_{img_name}.jpg', save_image)
        print(f'{idx}...{img_name}' + 'completed!!!')
```

암호 데이터셋이 부족하여 overlap하여 이미지를 잘라  
Image 개수를 4배 증가시켰다.

```
def crop_image(img):
    if not img.any(): # 2차원 배열같은 경우는 any() 또는 all() 로 True와 False를 판별한다.
        print("there is no image to crop!!!")
    else:
        cropped_img1 = img[:CROP_SIZE, :CROP_SIZE]
        cropped_img2 = img[img.shape[0] - CROP_SIZE:, :CROP_SIZE]
        cropped_img3 = img[img.shape[0] - CROP_SIZE:, img.shape[1] - CROP_SIZE:]
        cropped_img4 = img[:CROP_SIZE, img.shape[1] - CROP_SIZE:]

        return [cropped_img1, cropped_img2, cropped_img3, cropped_img4]
```



## Description

Under sampling image data set for solving imbalanced data problem

## 코드

```
import cv2
import numpy as np
import os
import glob

NUM_FILES = 2000

root_dir = r'E:\datacourse_free\team_project\pi_dataset'
print(root_dir)
target_classes = ['pi1', 'pi2', 'pi3']

for target_class in target_classes:
    images_path = root_dir + f'\{target_class}'
    images = glob.glob(images_path + r'\*.jpg')
    np.random.shuffle(images)

    for idx, image in enumerate(images):
        if len(os.listdir(images_path)) <= NUM_FILES:
            break

    os.remove(image)
    print(f'{idx}... {image} is removed!!!')

# ===== test all size is 380x380 =====
root_dir = r'D:\datacourse_free\team_project\bi_dataset\validation\pi3\*.jpg'
imgs = glob.glob(root_dir)

for i in imgs:
    img = cv2.imread(i, cv2.IMREAD_COLOR)

    if img.shape != (380, 380, 3):
        print('False')

print('All shape is matched!!!')
```

경증, 중등도, 중증 데이터가 양호데이터에 비해 월등히 많아 **imbalance**한 문제를 경증, 중등도, 중증 데이터를 **undersampling**했다.

## Description

## EfficientNet\_B4 Model learning code 1

## 코드

```
import os
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Flatten, Dropout, Dense, BatchNormalization
from tensorflow.keras.callbacks import ModelCheckpoint

from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.efficientnet import EfficientNetB4

# set train, test path to training
train_path = "E:/datacourse_free/team_project/misegakgeal_inspection/training"
validation_path = "E:/datacourse_free/team_project/misegakgeal_inspection/validation"

IMAGE_SIZE = (380, 380)

# set generator
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    # validation_split=0.2, # 밑에 training_datagen과 validation_datagen에 subset을 넣어주어야 한다.
    rotation_range=90, # UP
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2, # UP # 이미지 굴절
    zoom_range=0.2, # UP -> epoch을 늘려야 한다.
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='wrap' # reflect
)

validation_datagen = ImageDataGenerator(
    rescale=1. / 255,
)
```



## Description

## EfficientNet\_B4 Model learning code 2

### 코드

```

training_datagen = train_datagen.flow_from_directory(
    train_path,
    shuffle=True,
    batch_size=6,
    target_size=IMAGE_SIZE,
    class_mode='categorical'
)

validation_datagen = validation_datagen.flow_from_directory(
    validation_path,
    shuffle=True,
    batch_size=6,
    target_size=IMAGE_SIZE,
    class_mode='categorical',
)

NUM_CLASSES = 4

# include_top() True로 4 classes 중에서 올바른 클래스를 찾아내는 것 -> 모델의 요구사항!!
# 이 코드는... 새로운 차를 사진 살펴볼
base_model = EfficientNetB4(
    weights=None,
    include_top=True,
    input_shape=(IMAGE_SIZE, 3),
    pooling='avg',
    classes=NUM_CLASSES
)

base_model.summary()

# sgd scheduler
from tensorflow.keras.optimizers import SGD
initial_learning_rate = 0.01

# sgd scheduler from keras docs
sgd_opt = SGD(
    learning_rate=initial_learning_rate,
    decay=1e-6,
    momentum=0.9,
    nesterov=True
)

```

```

# sgd scheduler
from tensorflow.keras.optimizers import SGD
initial_learning_rate = 0.01

# sgd scheduler from keras docs
sgd_opt = SGD(
    learning_rate=initial_learning_rate,
    decay=1e-6,
    momentum=0.9,
    nesterov=True
)

checkpoint = ModelCheckpoint(
    filepath='E:/datacourse_free/team_project/final_model_0905.hdf5',
    monitor='val_loss',
    mode='min',
    save_best_only=True
)

# compile model
base_model.compile(
    loss='categorical_crossentropy',
    optimizer=sgd_opt,
    metrics=['acc']
)

# model training
history = base_model.fit_generator(training_datagen,
    epochs=150,
    validation_data=validation_datagen,
    verbose = 1,
    callbacks=[checkpoint]
)

```

## Description

## Evaluate 6 Models that we made

## 코드

```

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

def trained_model_load(model_path):
    return load_model(model_path)

def create_generator(test_dataset_path):
    test_datagen = ImageDataGenerator(rescale=1./255)

    test_datagenerator = test_datagen.flow_from_directory(
        test_dataset_path,
        target_size = (380, 380),
        batch_size = 6,
        class_mode = 'categorical'
    )

    return test_datagenerator

def show_evaluate(model_name, model, test_datagenerator):
    s = "="*20 + f" {model_name} Evaluate " + "="*20
    print()
    print(s)
    scores = model.evaluate_generator(test_datagenerator, steps=5)
    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    print("%s: %.5f%%" % (model.metrics_names[0], scores[0]))
    print("="*len(s), end="\n")
    print()

models = ['bideum', 'hongban', 'misae', 'nongpo', 'piji', 'talmo']

for model in models:
    model_name = model + '_effB4.hdf5'
    model_path = 'E:/datacourse_free/team_project/final_model' + f'/{model_name}'
    test_dataset_path = 'E:/datacourse_free/team_project/test_dataset' + f'/{model}'

    loaded_model = trained_model_load(model_path)
    test_datagenerator = create_generator(test_dataset_path)

    show_evaluate(model_name, model = loaded_model, test_datagenerator = test_datagenerator)
  
```

```

Found 8248 images belonging to 4 classes.

===== bideum_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.69352
=====
Found 13635 images belonging to 4 classes.

===== hongban_effB4.hdf5 Evaluate =====
acc: 80.00%
loss: 0.52033
=====
Found 3639 images belonging to 4 classes.

===== misae_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.63593
=====
Found 1070 images belonging to 4 classes.

===== nongpo_effB4.hdf5 Evaluate =====
acc: 70.00%
loss: 0.91290
=====
Found 16236 images belonging to 4 classes.

===== piji_effB4.hdf5 Evaluate =====
acc: 66.67%
loss: 0.80835
=====
Found 5288 images belonging to 4 classes.

===== talmo_effB4.hdf5 Evaluate =====
acc: 73.33%
loss: 0.96061
=====
  
```

## Description

Vgg16 Model → acc: about 50%

## 코드

```
import numpy as np
import glob
from tensorflow.keras.applications import VGG16
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Dropout, Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split

paths = glob.glob('E:/datacourse_free/team_project/head_skin_images4/미세각질/*/*.jpg')
paths = np.random.permutation(paths)

# plt.imread() -> image를 수치로 바꿔준다.
x_data = np.array([plt.imread(paths[i]) for i in range(len(paths))]) # 한 번에 다 불러와서
# keras에서 custom dataset 확인해보기
y_data = np.array([paths[i].split("\\")[-2] for i in range(len(paths))]) # Label만 뽑기 위해

label_dict = {
    '양호' : 0,
    '경증' : 1,
    '중증도' : 2,
    '중증' : 3
}

for i in range(len(y_data)):
    y_data[i] = label_dict[y_data[i]]

x_data = x_data / 255.
y_data = to_categorical(y_data)

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, stratify=y_data, test_size=0.2)
```

```
# imagenet -> 이미지 거동치를 가져온다.
transfer_model = VGG16(weights='imagenet', include_top=False, input_shape=(128, 128, 3))
transfer_model.trainable = False # True로 해보기

# concat additional model at the end of vgg16(preModel)
additional_model = Sequential()
additional_model.add(transfer_model)
additional_model.add(Flatten())

additional_model.add(Dense(2048, activation='relu'))
additional_model.add(Dense(1024, activation='relu'))
additional_model.add(Dense(4, activation='softmax'))

additional_model.summary()

checkpoint = ModelCheckpoint(filepath='test_batch_64.hdf5',
                             monitor='loss',
                             mode='min',
                             save_best_only=True)

additional_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])

history = additional_model.fit(x_train, y_train,
                               epochs=100,
                               validation_data=(x_test, y_test),
                               batch_size=64,
                               verbose=1,
                               callbacks=[checkpoint])
```

## Description

MobileNet Model → acc: about 50%

## 코드

```
from keras.applications import mobilenet
from keras.models import Sequential, Model
from keras.layers import Dense, Dropout, Activation, Flatten, GlobalAveragePooling2D
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D
from keras.layers.normalization import BatchNormalization
from keras.preprocessing.image import ImageDataGenerator

img_rows, img_cols = 224, 224
batch_size = 16
MobileNet = mobilenet.MobileNet(weights='imagenet', include_top=False, input_shape=(img_rows, img_cols, 3))

for layer in MobileNet.layers:
    layer.trainable = True

for (i, layer) in enumerate(MobileNet.layers):
    print(str(i), layer.__class__.__name__, layer.trainable)

def add_top_model(MobileNet, num_classes):
    # Add top model
    FC_Hoed = add_top_model(MobileNet, num_classes)
    model = Model(inputs=MobileNet.input, outputs=FC_Hoed)
    print(model.summary())

# Train, validation data split
train_paths = '/content/drive/MyDrive/scalp_images/Training/modeling_92/wisakajil_resize224'
train_paths

training_datagen = ImageDataGenerator(
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    fill_mode='nearest'
)

validation_generator = training_datagen.flow_from_directory(
    validation_paths,
    target_size=(img_rows, img_cols),
    batch_size=batch_size,
    class_mode='categorical'
)

train_generator = training_datagen.flow_from_directory(
    train_paths,
    target_size=(img_rows, img_cols),
    batch_size=batch_size,
    class_mode='categorical'
)

from tensorflow.keras.optimizers import RMSprop, Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

checkpoint = ModelCheckpoint(
    'model_{epoch:02d}.h5',
    monitor='val_acc',
    save_best_only=True,
    save_weights_only=True
)

earlystop = EarlyStopping(
    monitor='val_acc',
    min_delta=0.001,
    patience=10
)

learning_rate_reduction = ReduceLROnPlateau(monitor='val_acc',
                                             patience=5,
                                             verbose=1,
                                             factor=0.5)

callbacks = [earlystop, checkpoint, learning_rate_reduction]

model.compile(loss='categorical_crossentropy',
              optimizer=Adam,
              metrics=['accuracy'])

nb_train_samples = 34176
nb_validation_samples = 3000

epochs = 25

history = model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples // batch_size,
    epochs=epochs,
    callbacks=callbacks
)
```

## First test EfficientNet\_B0

## 코드

## References

**Keywords:** *workplace spirituality, organizational commitment, organizational citizenship behavior, organizational trust, organizational identification*

Wynn

Figure 1



## Description

## EfficientNet\_B0 + Imagedatagenerator

### 코드

```
train_paths = os.path.join(os.getcwd(), '1.미세작물_resize_224x224')
train_paths

- 'c:\\Users\\minkyu_sia\\Desktop\\무미작물프로젝트\\1.미세작물_resize_224x224'

training_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.3,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)

validation_datagen = training_datagen.flow_from_directory(
    train_paths,
    shuffle=True,
    batch_size=16,
    target_size=(224, 224),
    class_mode='categorical',
    color_mode='rgb',
    subset='validation'
)

training_datagen = training_datagen.flow_from_directory(
    train_paths,
    shuffle=True,
    batch_size=16,
    target_size=(224, 224),
    class_mode='categorical',
    color_mode='rgb',
    subset='training'
)

Found 2506 images belonging to 4 classes.
Found 1000 images belonging to 4 classes.

transfer_model = EfficientNetB0(weights=None, include_top=False, input_shape=(224, 224, 3))

additional_model = Sequential()
additional_model.add(transfer_model)
additional_model.add(Flatten())

additional_model.add(Dense(1024, activation=tf.keras.layers.LeakyReLU(alpha=0.1)))
additional_model.add(Dropout(0.5))
additional_model.add(Dense(512, activation=tf.keras.layers.LeakyReLU(alpha=0.1)))
additional_model.add(Dropout(0.5))
additional_model.add(Dense(4, activation='softmax'))

additional_model.summary()

checkpoint = ModelCheckpoint(filepath='first_eff_B0.h5',
    monitor='loss',
    mode='min',
    save_best_only=True)

additional_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])

history_record = additional_model.fit(training_datagen,
    epochs=100,
    validation_data=validation_datagen,
    callbacks=[checkpoint])
```

학습 정확도가 안올라서 중도 중단

## Description

## EfficientNet\_B0 + Imagedatagenerator값 조정 + Learning rate dacay

### 코드

```

train_paths = os.path.join(os.getcwd(), '1.미세각들_resize_224224')
train_paths

'c:\\Users\\Minkyu_Kim\\Desktop\\두리안프로젝트\\1.미세각들_resize_224224'

training_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.1,
    rotation_range=40,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)

validation_datagen = training_datagen.flow_from_directory(
    train_paths,
    shuffle=True,
    batch_size=16,
    target_size=(224, 224),
    class_mode='categorical',
    color_mode='rgb',
    subset='validation'
)

training_datagen = training_datagen.flow_from_directory(
    train_paths,
    shuffle=True,
    batch_size=16,
    target_size=(224, 224),
    class_mode='categorical',
    color_mode='rgb',
    subset='training'
)

Found 1272 images belonging to 4 classes.
Found 1167 images belonging to 4 classes.

transfer_model = EfficientNetB0(weights=None, include_top=False, input_shape=(224, 224, 3))

additional_model = keras.Sequential()
additional_model.add(transfer_model)
additional_model.add(Flatten())

additional_model.add(Dense(1024, activation=tf.keras.layers.LeakyReLU(alpha=0.1)))
additional_model.add(Dropout(0.5))
additional_model.add(Dense(512, activation=tf.keras.layers.LeakyReLU(alpha=0.1)))
additional_model.add(Dropout(0.5))
additional_model.add(Dense(4, activation='softmax'))

additional_model.summary()
# keras.utils.plot_model(additional_model, show_shapes=True, dpi=100)

# global_step = tf.Variable(0, trainable=False)

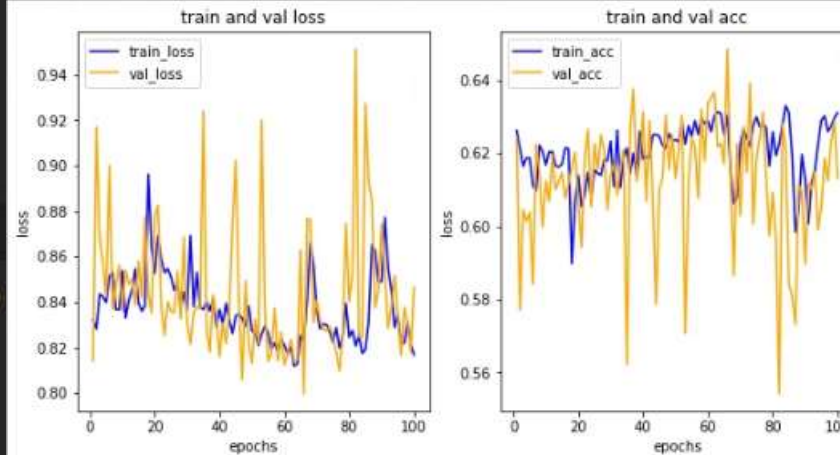
start_learning_rate = 1e-4
lr_schedule = ExponentialDecay(start_learning_rate, decay_steps=100000, decay_rate=0.96, staircase=True)

checkpoint = ModelCheckpoint(filepath='third_eff_b0.h5',
                             monitor='loss',
                             mode='min',
                             save_best_only=True)

early_stopping_cb = keras.callbacks.EarlyStopping(patience=2,
                                                  restore_best_weights=True)

additional_model.compile(loss='categorical_crossentropy', optimizer=Adam(lr_schedule), metrics=['acc'])

history_second = additional_model.fit(training_datagen,
                                     epochs=200,
                                     validation_data=validation_datagen,
                                     callbacks=[checkpoint, early_stopping_cb])
    
```



학습실패



## Description

## Kaggle skin disease learning model test

### 코드

```
train_paths = os.path.join(os.getcwd(), '미세각질_resize300_crop4', '미세각질_resize300_crop4_train')
val_paths = os.path.join(os.getcwd(), '미세각질_resize300_crop4', '미세각질_resize300_crop4_val')
print(train_paths, '\n', val_paths)

c:\Users\Winky_Kim\Desktop\두피약포도씨분\미세각질_resize300_crop4\미세각질_resize300_crop4_train
c:\Users\Winky_Kim\Desktop\두피약포도씨분\미세각질_resize300_crop4\미세각질_resize300_crop4_val

prepare validation data:

paths = glob.glob('C:/Users/Winky_Kim/Desktop/두피약포도씨분/미세각질_resize300_crop4/미세각질_resize300_crop4_val/**/*.jpg')
paths = np.random.permutation(paths)
paths[0]

'C:/Users/Winky_Kim/Desktop/두피약포도씨분/미세각질_resize300_crop4/미세각질_resize300_crop4_val\\1.검은\\0043_A211833BE00048F_1005511361632_2_TH.jpg_rb.jpg'

X_val = np.array([plt.imread(paths[i]) for i in range(len(paths))])
y_val = np.array([paths[j].split('\\')[-2] for j in range(len(paths))])
print(X_val.shape, y_val.shape)

(4800, 300, 300, 3) (4800,)

# Set the CNN model
# my CNN architecture is In -> [[Conv2D-vrelu]*2 -> MaxPool2D -> Dropout]*3 -> Flatten -> Dense*2 -> Dropout -> Out
input_shape = (300, 300, 3)
num_classes = 3

fifth_model = Sequential()
fifth_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', padding = 'same', input_shape=input_shape))
fifth_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', padding = 'same',))
fifth_model.add(MaxPool2D(pool_size = (2, 2)))
fifth_model.add(Dropout(0.16))

fifth_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', padding = 'same'))
fifth_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', padding = 'same',))
fifth_model.add(MaxPool2D(pool_size = (2, 2)))
fifth_model.add(Dropout(0.20))

fifth_model.add(Conv2D(64, (3, 3), activation='relu', padding = 'same'))
fifth_model.add(Conv2D(64, (3, 3), activation='relu', padding = 'same'))
fifth_model.add(MaxPool2D(pool_size=(2, 2)))
fifth_model.add(Dropout(0.25))

fifth_model.add(Flatten())
fifth_model.add(Dense(256, activation='relu'))
fifth_model.add(Dense(128, activation='relu'))
fifth_model.add(Dropout(0.4))
fifth_model.add(Dense(num_classes, activation='softmax'))
fifth_model.summary()
```

```
# Define the optimizer
optimizer = Adam(lr=0.0001, beta_1=0.0, beta_2=0.999, epsilon=1e-08, decayed=False, amsgrad=False)

# compile the model
fifth_model.compile(optimizer=optimizer, loss="categorical_crossentropy", metrics=["accuracy"])

checkpoint = ModelCheckpoint(filepath='kaggle_skin_disease_modeltest.h5',
                             monitor='loss',
                             mode = 'min',
                             save_best_only=True)

# Set a learning rate scheduler
learning_rate_reduction = ReduceLROnPlateau(monitor='val_accuracy',
                                             patience=4,
                                             verbose=1,
                                             factor=0.5,
                                             min_lr=0.000)

datagen = ImageDataGenerator(
    rescale=1./255,
    featurewise_center=False, # set input mean to 0 over the dataset
    samplewise_center=False, # set each sample mean to 0
    featurewise_std_normalization=False, # divide inputs by std of the dataset
    samplewise_std_normalization=True, # divide each input by its std
    zca_whitening=False, # apply ZCA whitening
    rotation_range=10, # randomly rotate images in the range (degrees), 0 to 180
    zoom_range = 0.1, # randomly zoom image
    width_shift_range=0.12, # randomly shift images horizontally (fraction of total width)
    height_shift_range=0.12, # randomly shift images vertically (fraction of total height)
    horizontal_flip=True, # randomly flip images
    vertical_flip=True, # randomly flip images
    fill_mode='reflect') # randomly fill images

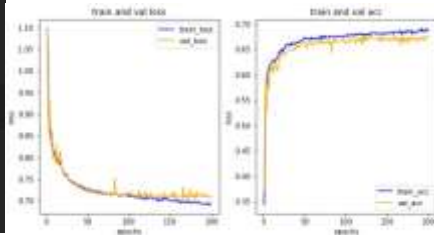
training_datagen = datagen.flow_from_directory(
    train_paths,
    shuffle=True,
    batch_size=16,
    target_size=(300, 300),
    class_mode='categorical',
    color_mode='rgb'
)

datagen_val = ImageDataGenerator(
    rescale=1./255) # randomly flip images

validation_datagen = datagen_val.flow_from_directory(
    val_paths,
    shuffle=True,
    batch_size=16,
    target_size=(300, 300),
    class_mode='categorical',
    color_mode='rgb'
)

Found 4800 images belonging to 3 classes.

history_kaggle = Fifth_model.fit_generator(training_datagen,
                                           epochs=200,
                                           validation_data=validation_datagen,
                                           callbacks=[checkpoint, learning_rate_reduction])
```



[illegible]



Microsoft Todo 활용  
실시간 작업 동기화

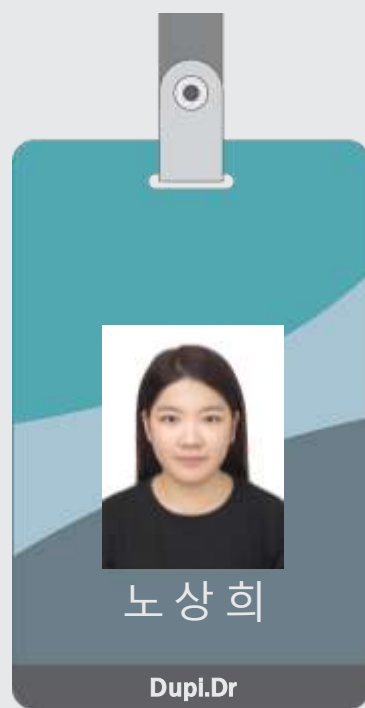
# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

# Q & A



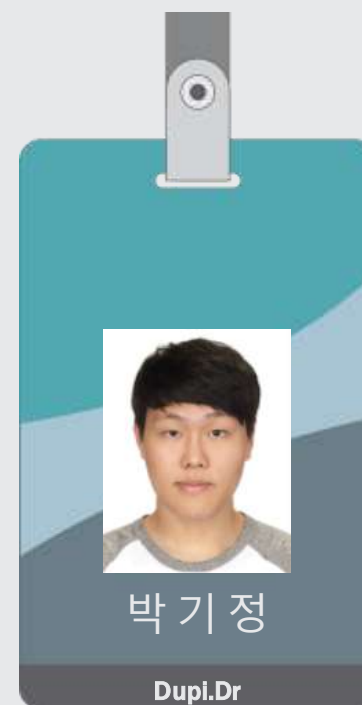
홍익대학교 빅데이터 개발자 청년인재 양성사업 프로젝트1  
개발 기간: 2021.8.13 ~ 2021.9.10  
두피닥컴퍼니  
대표번호: 010-7211-1858(김민교)



# Dupi. Dr

두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 프로젝트 제작자 소개



# Dupi.

# Dr



두피 상태를 분석해주는 내 손안의 AI, 두피Dr

## 감사합니다.

이중협 멘토님!  
오소진 교수님!  
신건철 멘토님!  
정말 감사합니다!