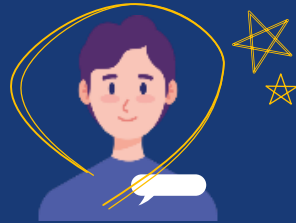
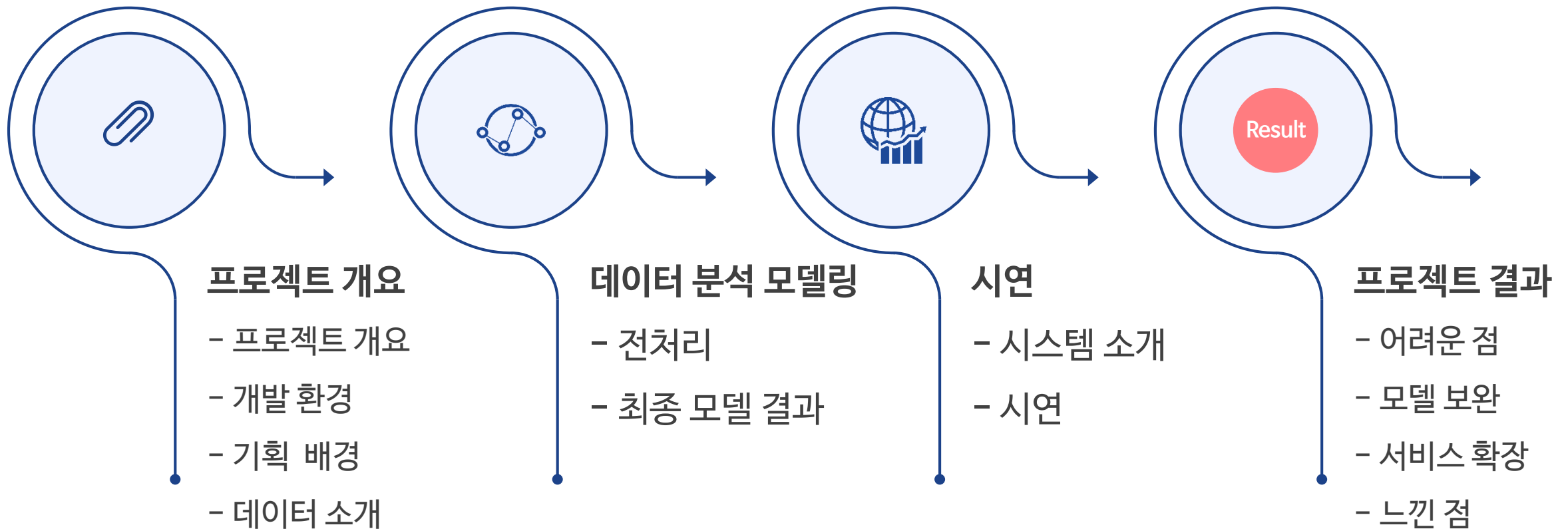


ScalpSpectra

#딥러닝을 이용한 두피 자가진단 서비스





목적 & 목표

- 웹에서 무료 제공되는 자가진단으로
뛰어난 접근성과 편리함 제공
- 두피 질환과 중증도를 자동 분류

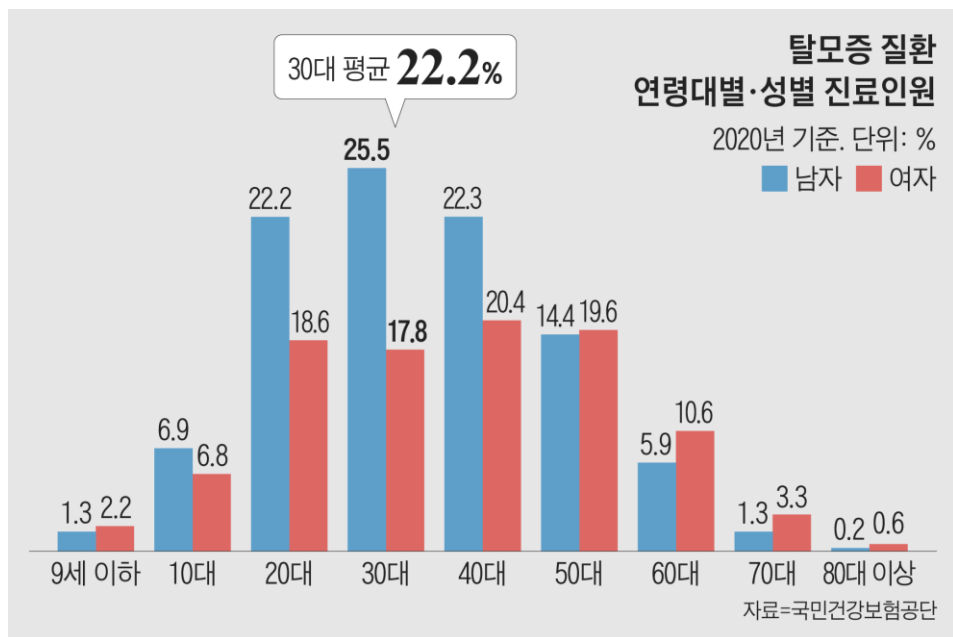
개발 환경



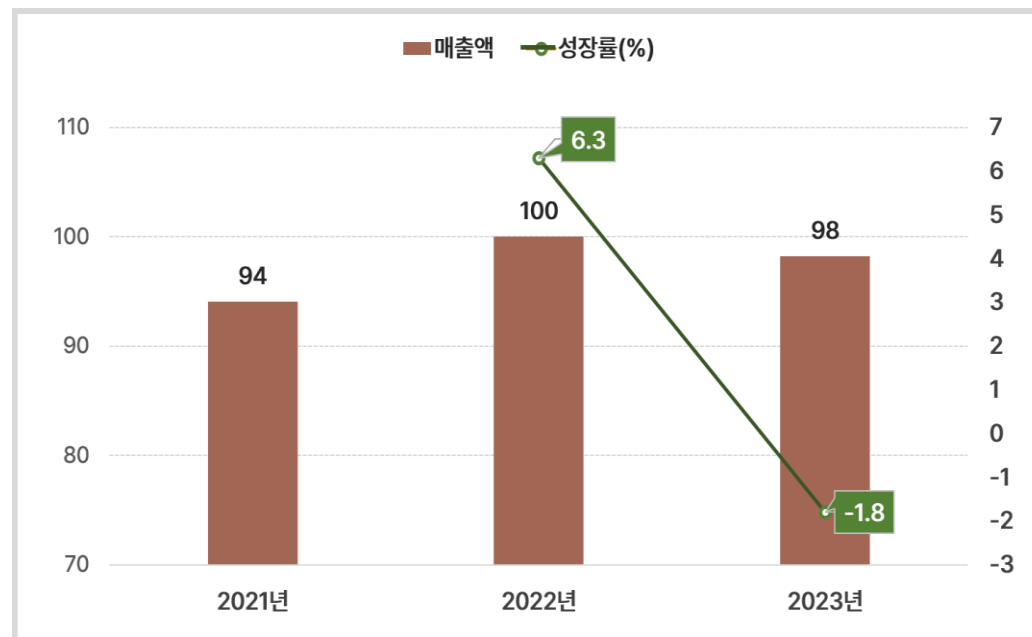


기획 배경

젊은 층에서도 발생하는 탈모



성장하는 두피 케어 시장



데이터 셋



유형별 두피 이미지

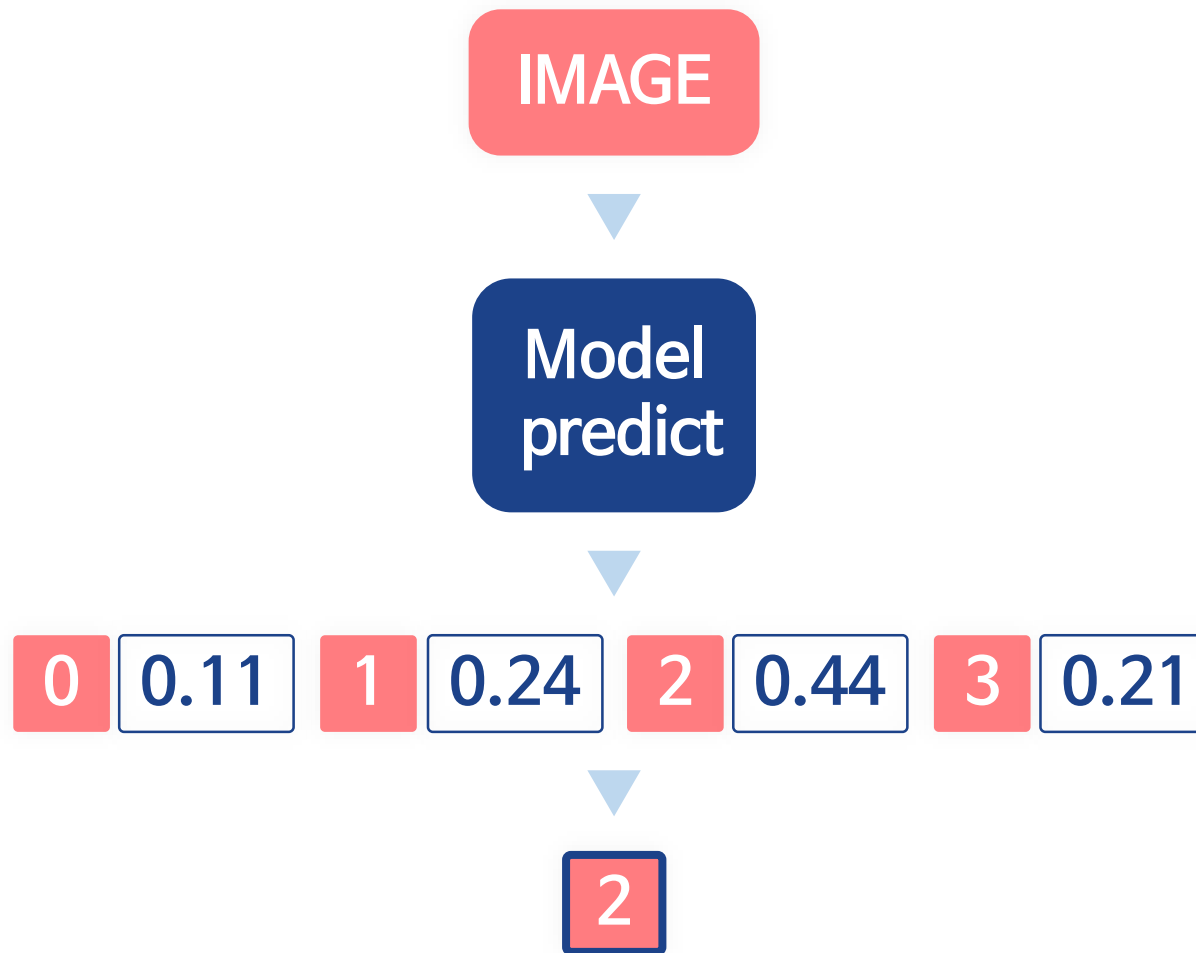
증상별 구분	건수
미세각질	17,434건
피지과다	80,416건
모낭사이홍반	67,414건
모낭홍반/농포	4,592건
비듬	40,482건
탈모	25,682건
양호	811건



데이터 라벨링

유형별 중증도

결과	출력
양호	0
경증	1
중등도	2
중증	3





EfficientNet B0

Efficient → 효율적인

Input Image (224, 224)

Conv 3x3 (색상 정보 처리)

MBConv1, k 3×3

MBConv6, k 3×3 x2

MBConv6, k 5×5 x2

MBConv6, k 3×3 x3

MBConv6, k 5×5 x3

MBConv6, k 5×5 x3

MBConv6, k 3×3 x4

Conv 1x1, Pooling, FC

Output



MBConv

- Mobile Inverted Bottleneck Convolution
- 모바일 기기에서 동작하는 것을 목표로 한 CNN 모델

EfficientNet에서는
 $A \times B \times C \rightarrow a \times b \times c$ 변환,
다시 $A \times B \times C$ 로 변환해
가중치를 적용하는 SE 추가

기존 Convolution

Input(3x3x3)



Output(1)

MBConv1

Input(3x3 + 3x3 + 3x3)



Output(1 + 1 + 1)

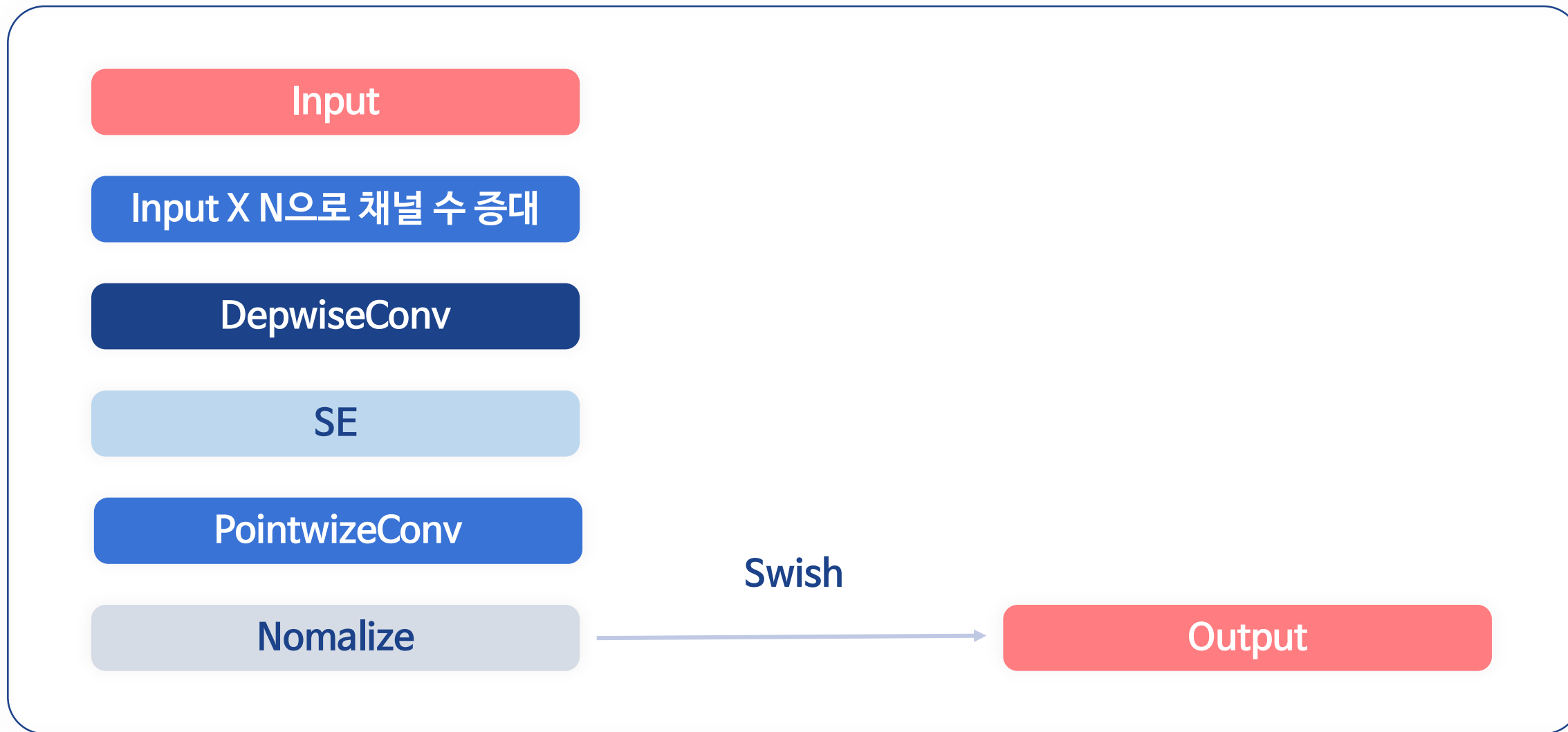


Output(1)



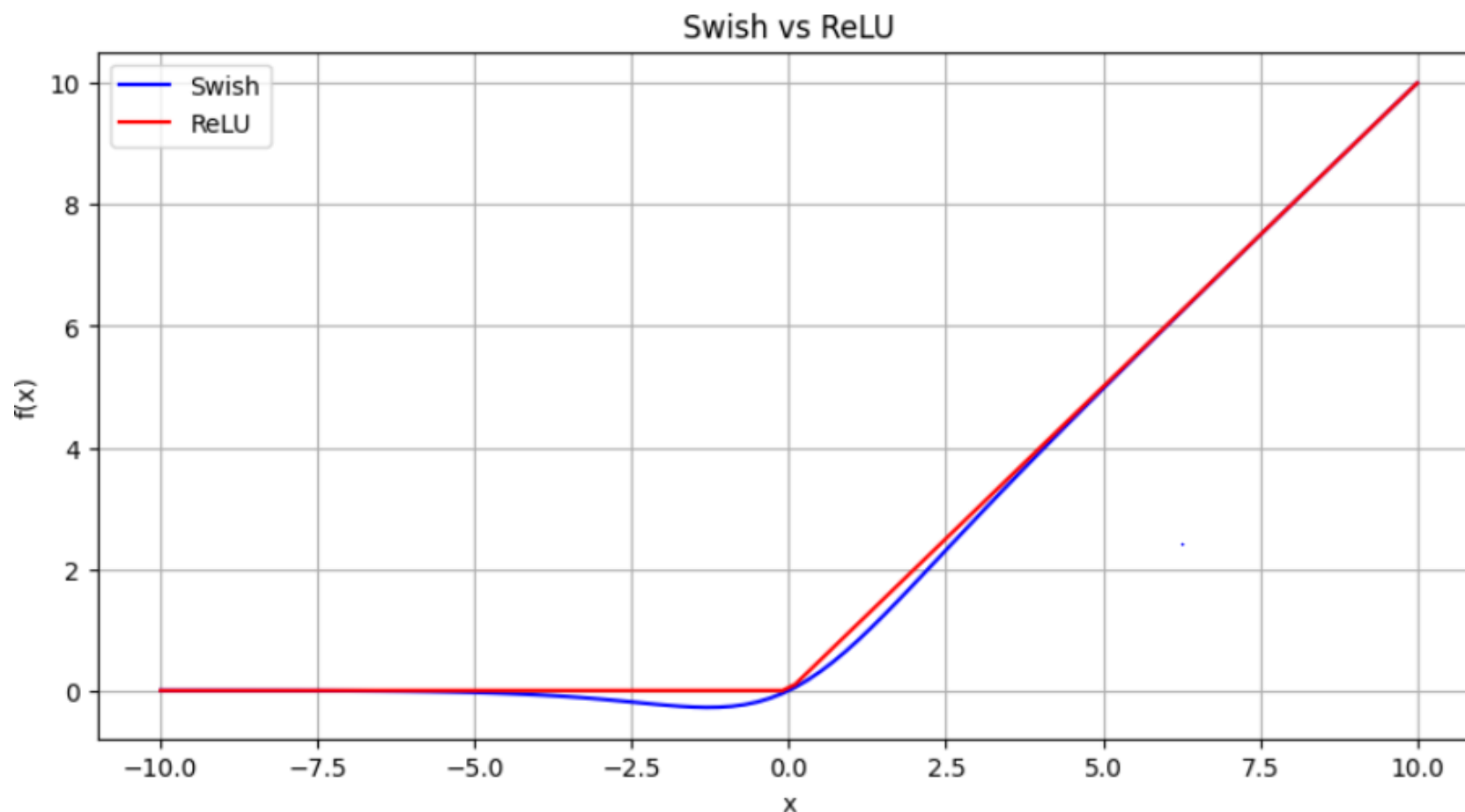


MBConv(N) 블록





Swish 함수



Swish

$$f(x) = x \cdot \sigma(x)$$

변형 Swish

$$f(x) = x \cdot \sigma(\beta x)$$



모델 학습 - 이미지 전처리

IMAGE
640 × 480



Transform.Compose()

Resize(300)

RandomHorizontalFlip(0.5)

RandomverticalFlip(0.5)

rotate(90)

RandomRotation(10)

RandomAffine(0, 10, (0.8, 1.2))

toTensor()

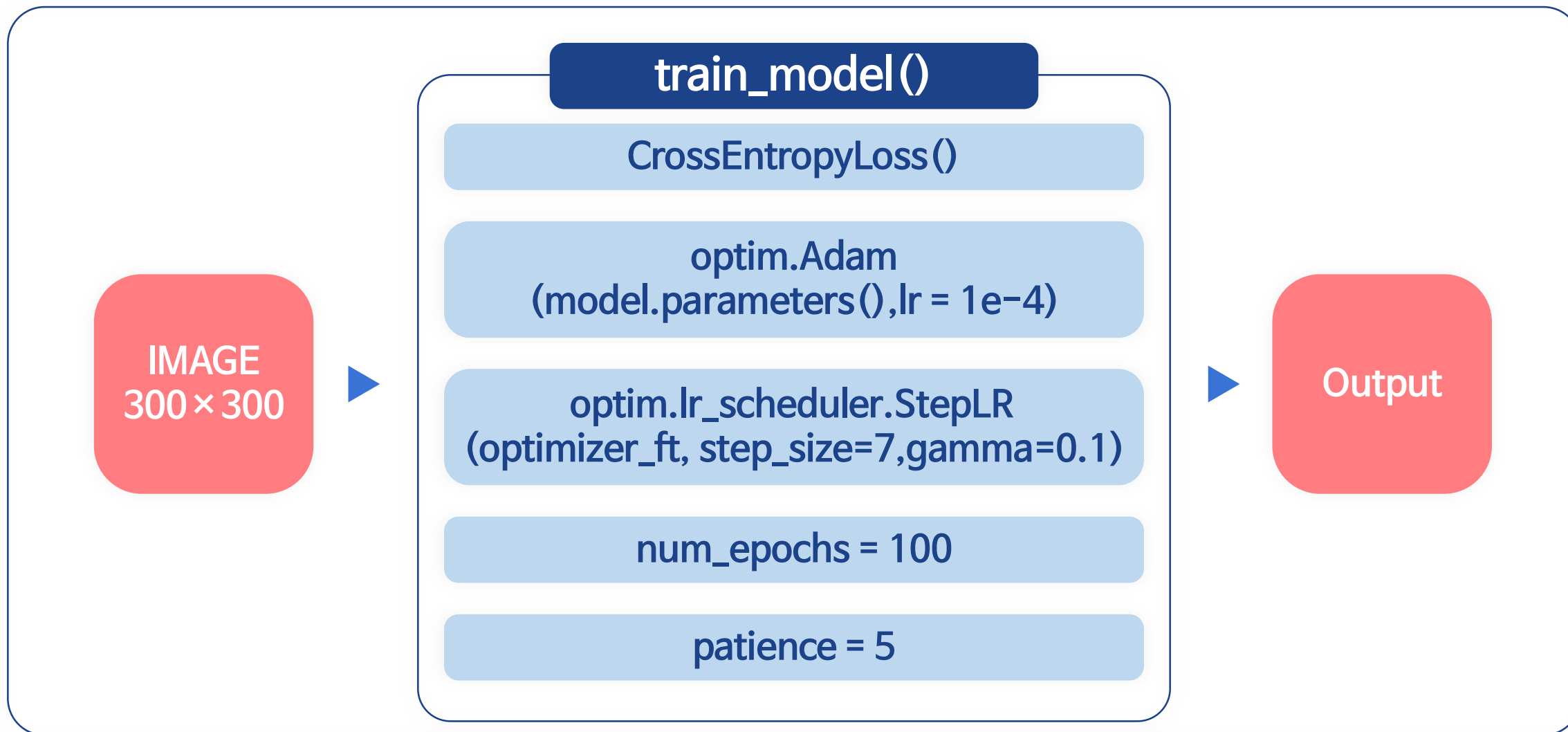
Normalize()



IMAGE
300 × 300

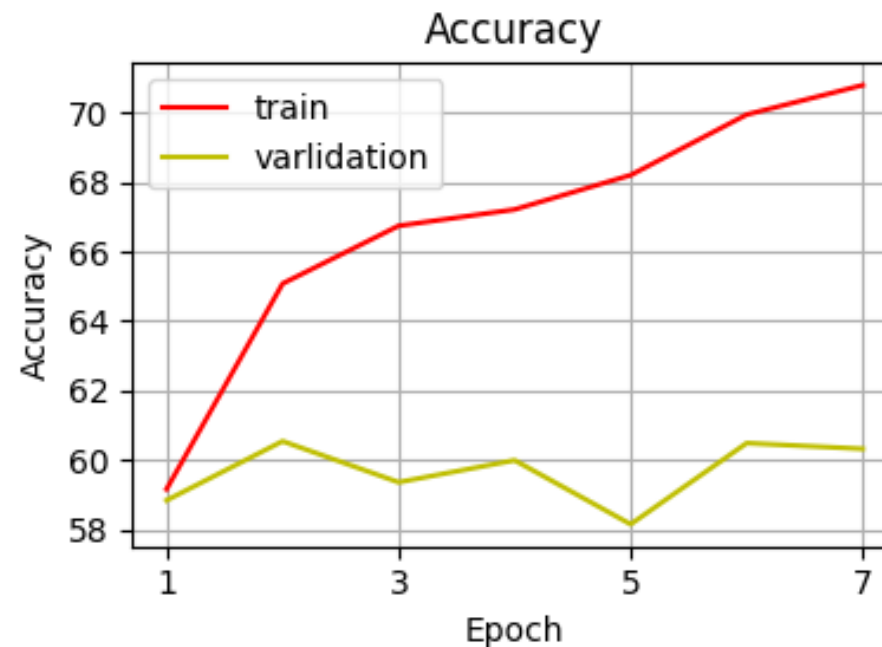
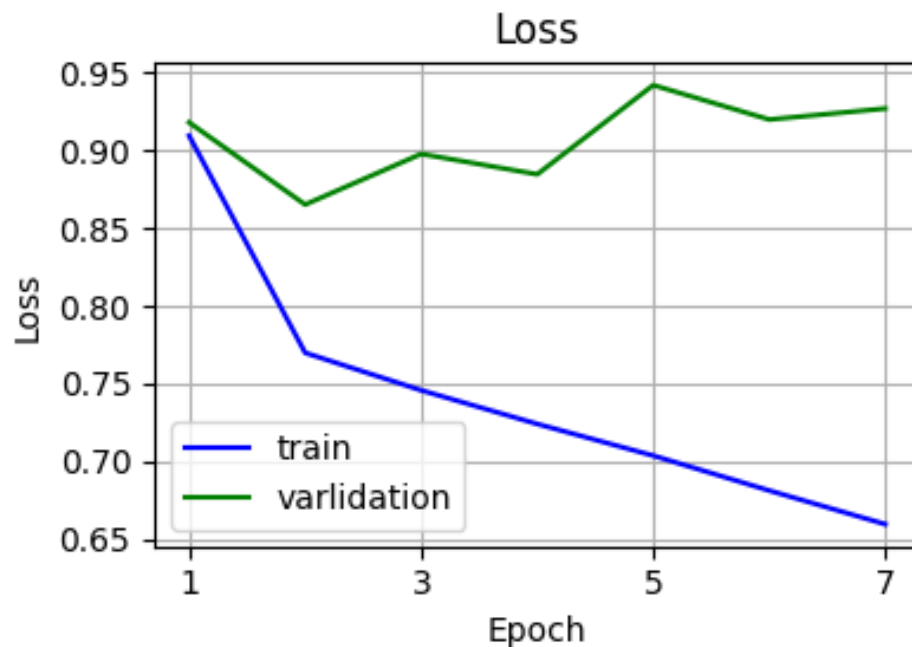


모델 학습 - 훈련



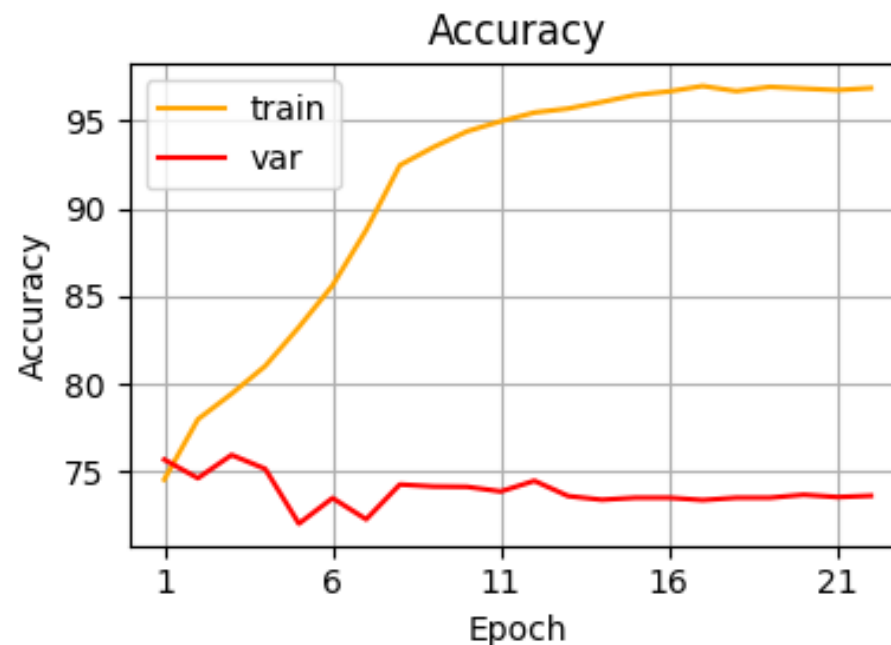
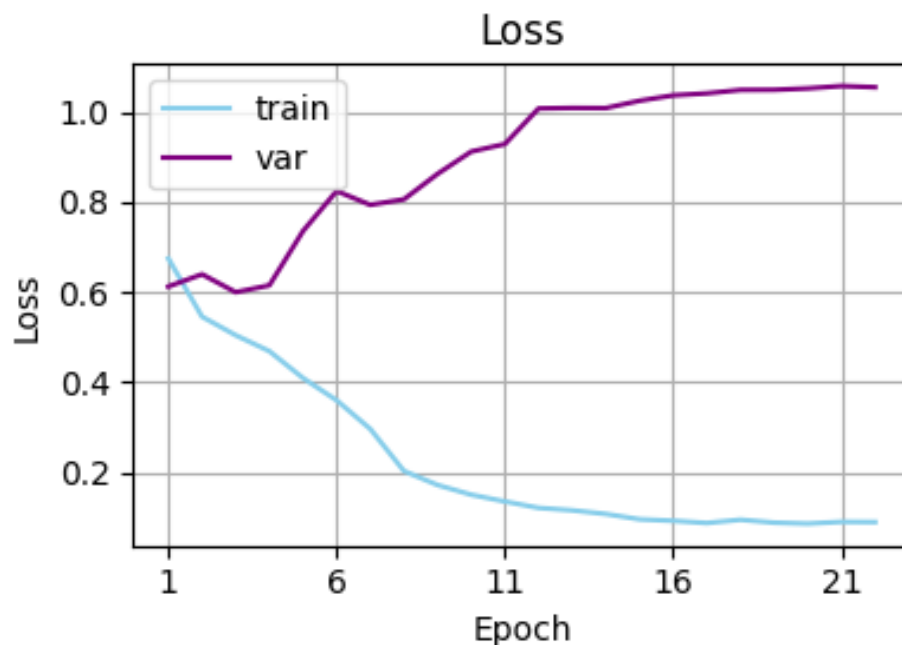
모델링 결과 - EfficientNet b0(input_size = 224×224)

- 에포크 10번 기준
- 소요시간 12분, 정확도 60 %



모델링 결과 - EfficientNet b7 (input_size = 300×300)

- 에포크 20번 기준
- 소요시간 3시간, 정확도 74%



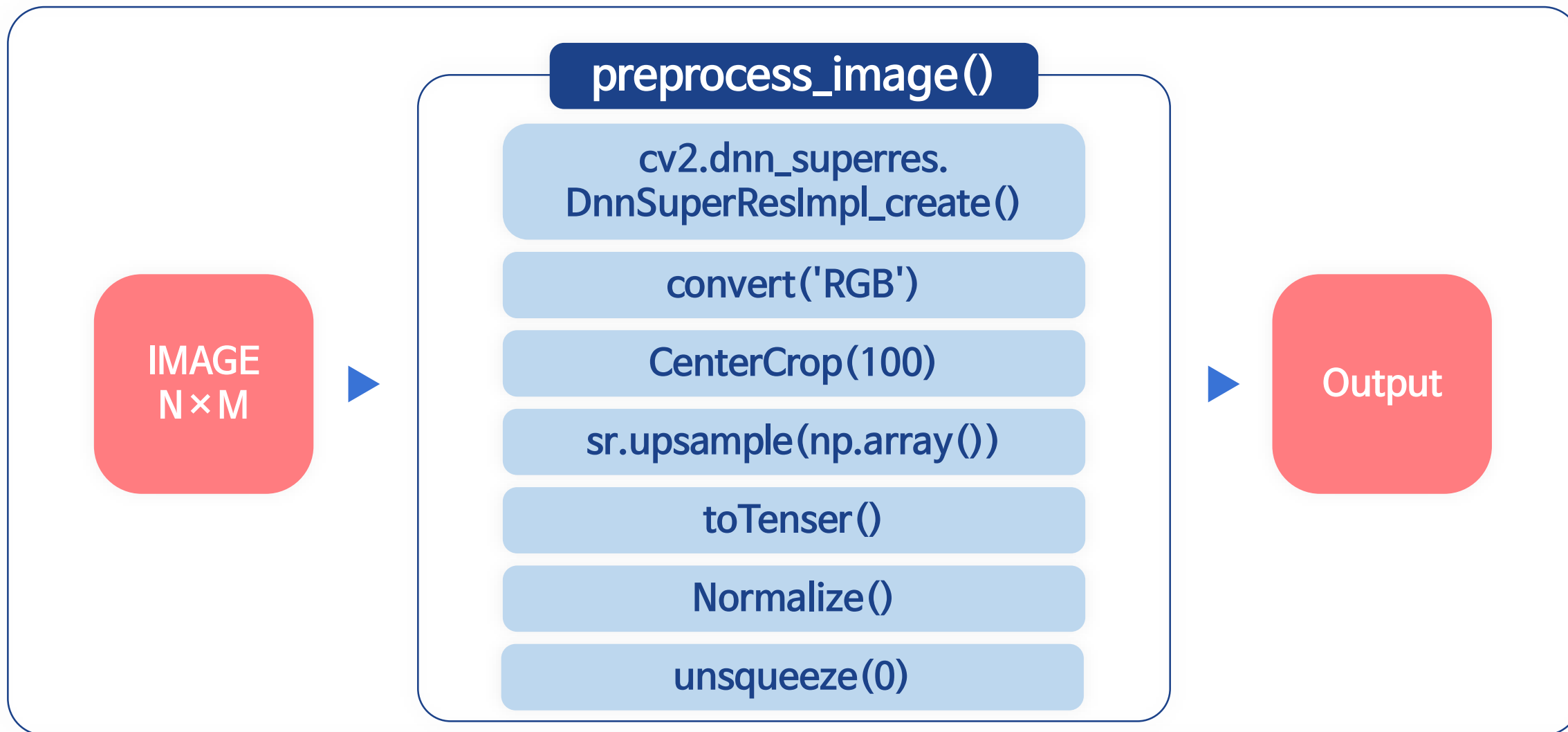


모델링 결과

- 600×600 은 메모리 부족으로 동작이 힘들
- 가용 메모리를 최대한 활용할 수 있게 크기 조절
- 300으로 해도 b7이 가장 잘 나옴
- 이미지 용량이 적은 것은 b7으로 채택

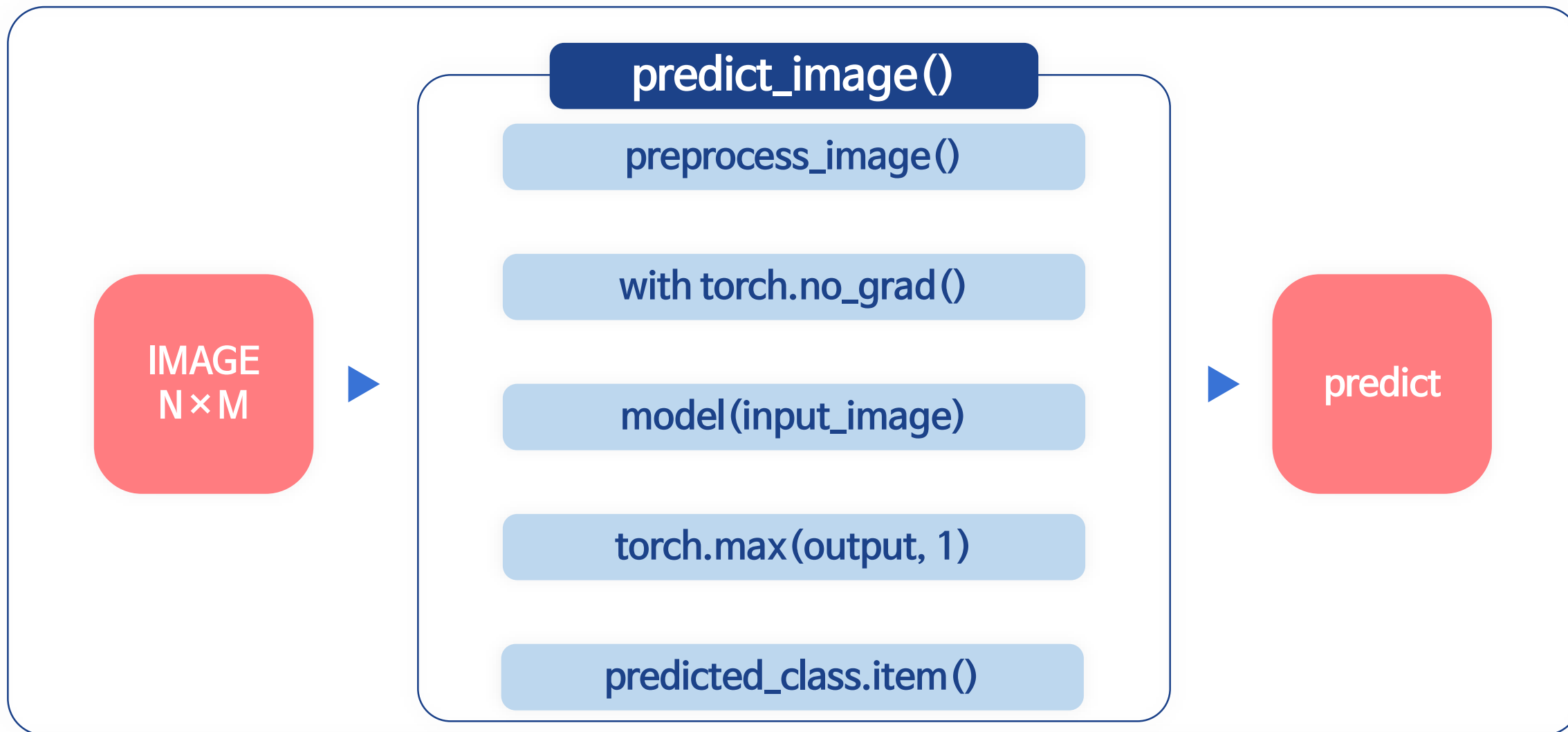


모델 적용 - 입력 이미지 전처리

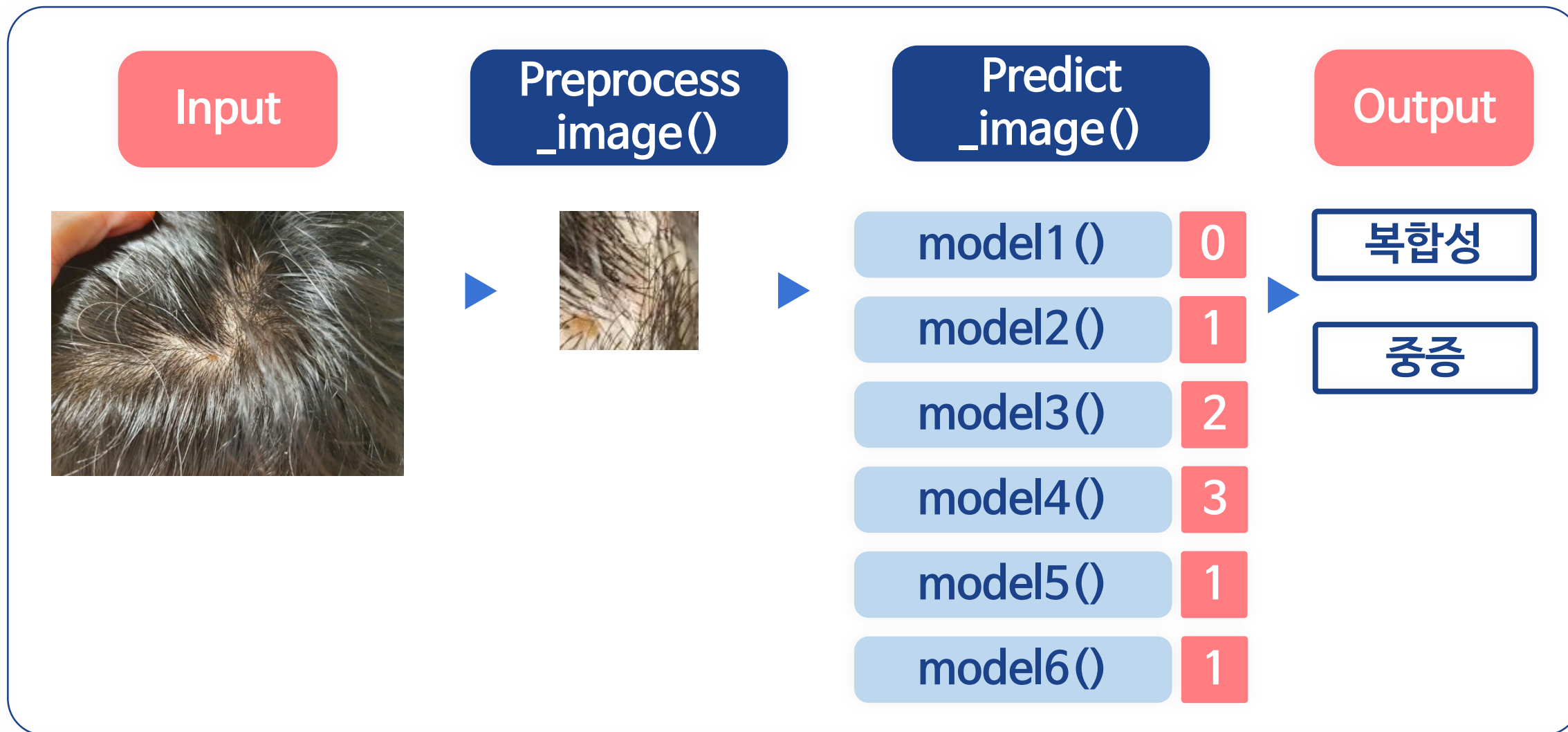




모델 적용 - 각 모델 예측

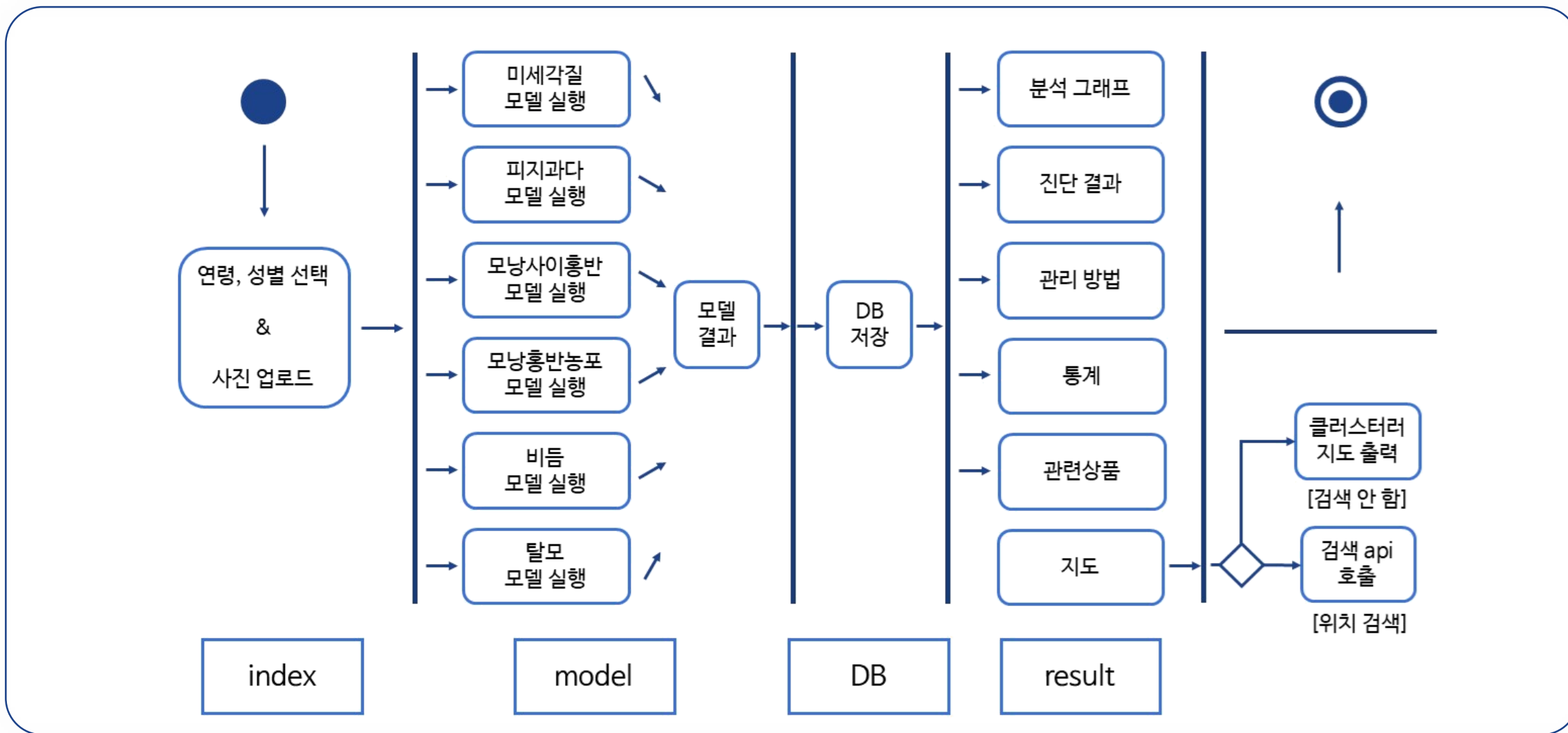


모델 적용





시스템 구성



DB

Human

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

Models.py - 모델 정의

Django-Admin

Views.py - 모델 데이터 저장

```
human = Human(sex=gender, age=age, pred=m_c[p], skin=f_class[v])
human.save()
```



프로젝트 시 어려움

문제

정확도를 높이려면 최적화된 resolution에 맞춰야 하는데, B7 은 600x600이라서 메모리에 과부하가 왔다.

```
OutOfMemoryError: CUDA out of memory. Tried to allocate 2.06 GiB. GPU 0 has a total capacity of 14.75 GiB of which 1.81 GiB is free. Process 34560 has 12.94 GiB memory in use. Of the allocated memory 12.45 GiB is allocated by PyTorch, and 366.09 MiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try setting max_split_size_mb to avoid fragmentation. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF
```

해결

Colab Pro+ 구매 후, GPU(40GB)를 가장 효율적으로 쓸 수 있게 Image Size를 300x300으로 지정

사용 메모리 : 33GB/40GB

다른 방법 - Batch Size 줄이기

현재 모델은 32인데. Batch를 더 줄이면 메모리 사용량은 줄어들어 600x600도 가능했지만, content에 넣을 수 있는 용량때문에 drive.mount를 사용했다.



```
from google.colab import drive
drive.mount('/content/gdrive')
```

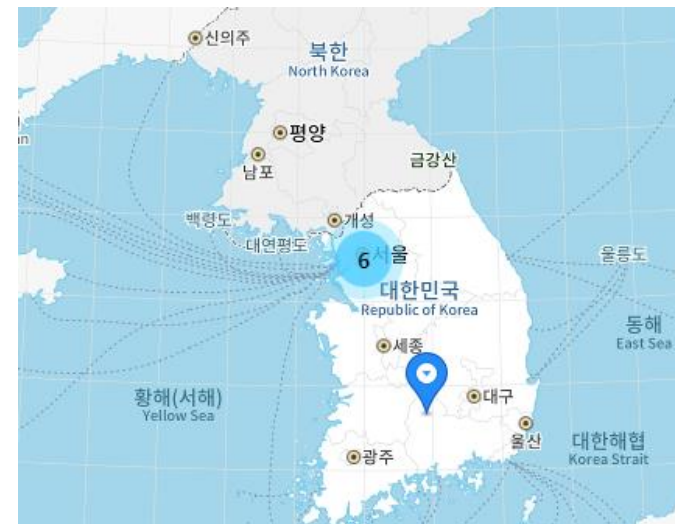

프로젝트 시 어려움

문제

지도 마커가 일부만 표시되는 것을 확인
지도 API 라이브러리 2개중 하나만 작동

해결

100개씩 잘라서, 좌표값이 든 파일을 같이 확인해
마커의 표시 유무차이를 확인하고, 좌표의 x, y값을 바꾸었다.
기존에 포트 번호를 지정하지 않은 것을 지정하여 해결했다.

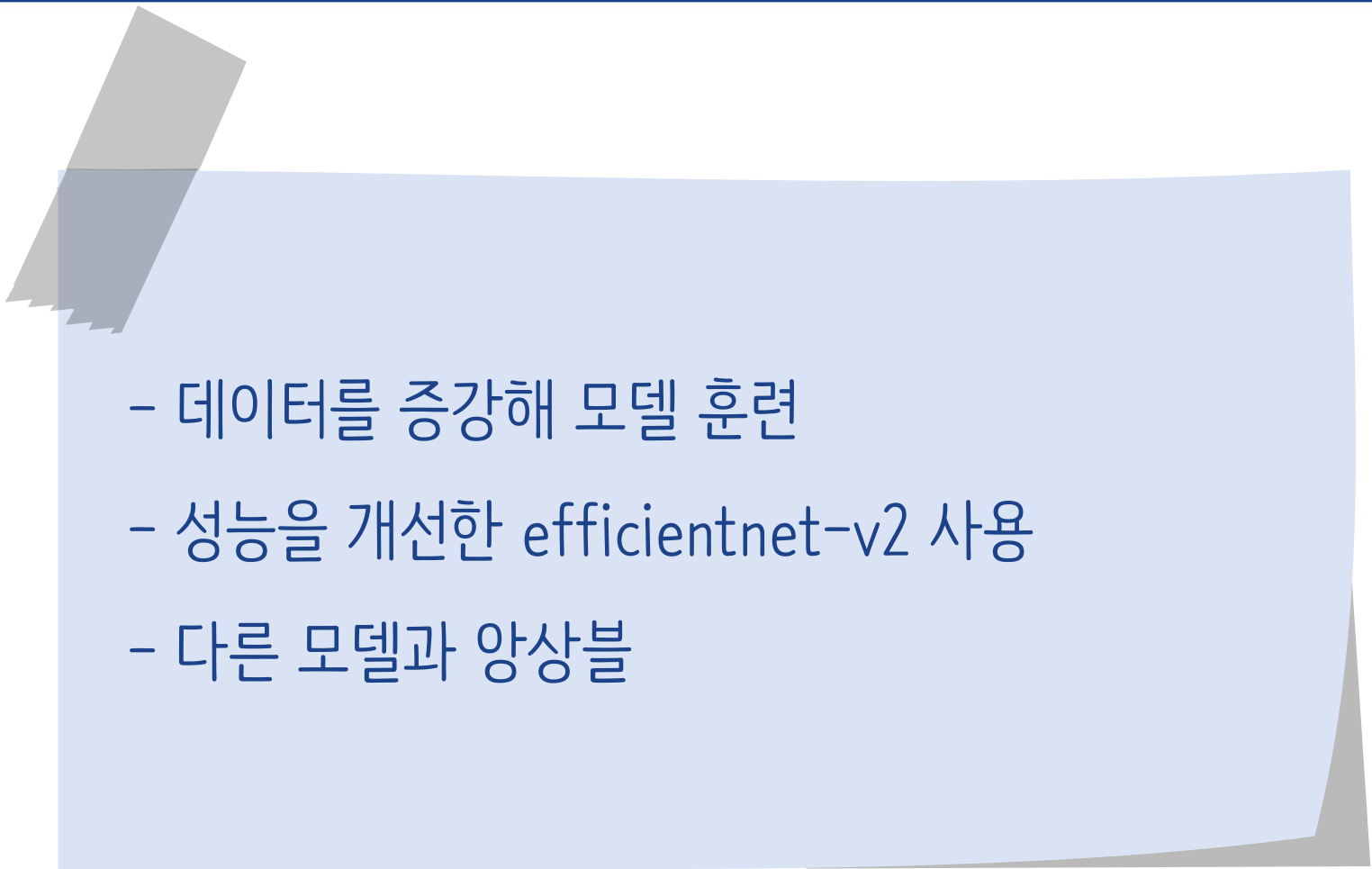


Web

사이트 도메인

<http://127.0.0.1:8000>

모델 보완

- 
- 데이터를 증강해 모델 훈련
 - 성능을 개선한 efficientnet-v2 사용
 - 다른 모델과 앙상블

서비스 확장

- 상품클릭 수도 DB에 저장해 상품추천 서비스 개발
- 똑닥같은 앱과 연계해 병원 예약 안내
- 두피 질환 관련 질문을 받아줄 챗봇 개발
- 다른 인종의 두피 모델 개발

느낀 점

- 김경수

이미지 딥러닝은 컴퓨터 성능이 중요하다는 것을 느꼈습니다. 뭔가 해보려 하면 out of memory 떠버리니까 적당한 크기와 batch size를 찾는 데에 제한된 컴퓨팅 단위를 많이 소모한 것 같습니다. 그리고 Django로 웹 서비스를 하는데, 외부에서 하나하나 틀을 만들어놓고 웹에 장착하는 것이 재미있었습니다.

- 신호섭

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

- 송재준

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

- 안은지

얻은 것도 많지만 얼마나 무지한지 적나라하게 알게 된 소중한 시간이였다. 계속 모르는 개념이 나와 시간이 오래 걸려, 아는 만큼 보인다라는 말을 백번 실감했다. 또한, 공식문서를 숙지해야 응용이 가능함을 되새기는 계기가 되었다.

Q&A

