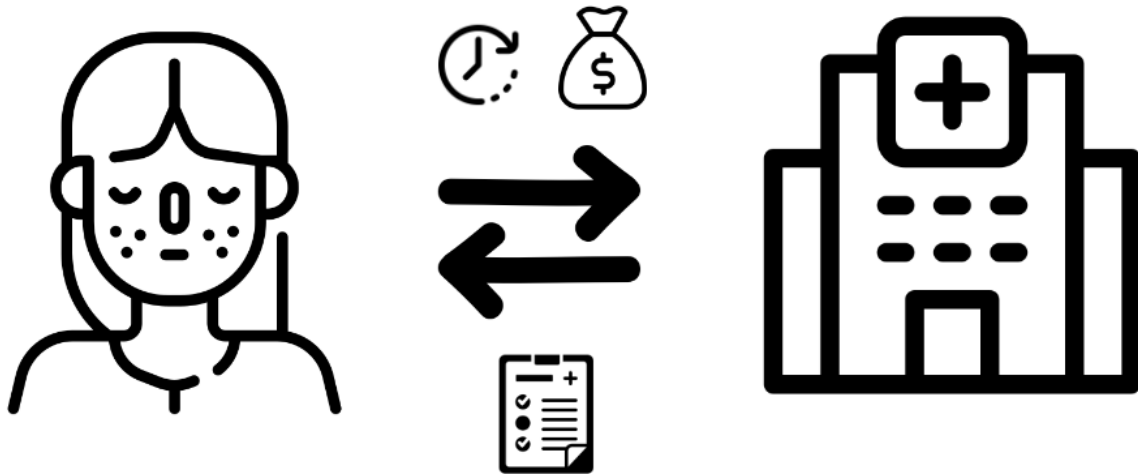




랩업리포트

😊 피부 평가 (XAI) 시스템

- 서비스 배경 및 목표



아직까지 사람들은 피부를 진단하고 지속적으로 관리하기 위해서 피부과를 방문하고 비싼 비용을 지불해야 합니다.

이러한 물리적 제약을 없애고자, 간단하게 피부를 진단할 수 있는 서비스를 만들어 사용자들에게 편의성을 제공하기 위해 프로젝트를 기획하게 되었습니다.

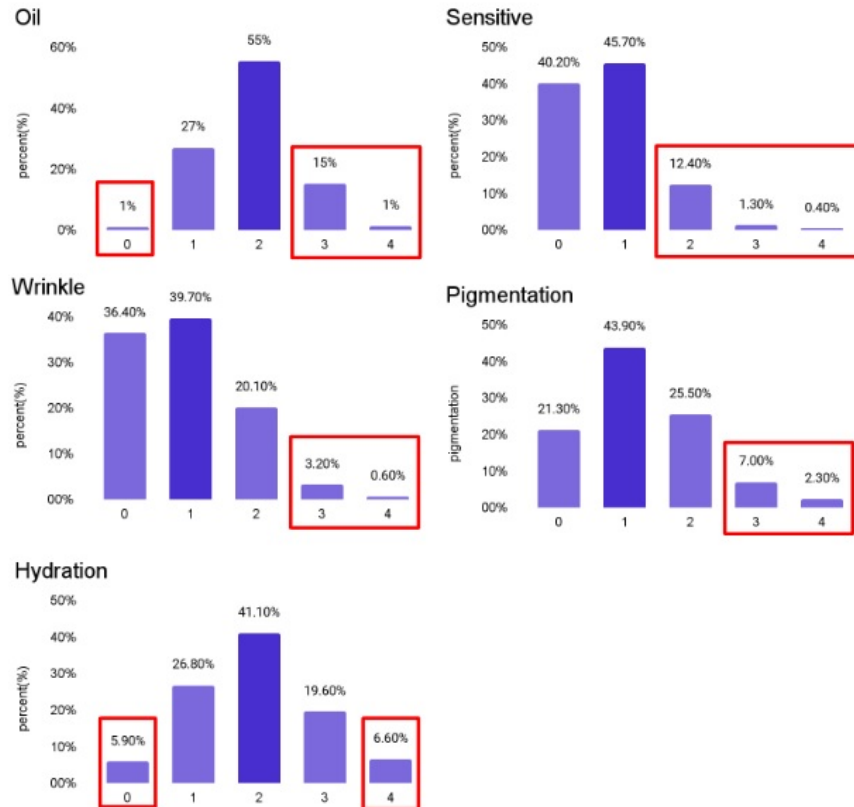
이 피부 진단 서비스의 목표는 스마트폰 혹은 웹캠을 통해 얼굴 사진을 촬영하여, 각 항목에 대한 점수와 이에 대한 시각적 근거를 함께 제공하는 것입니다.

👤 개발 환경

- GPU환경 : V100 서버, Google Cloud Platform(GCP) 서버
- 팀 협력 Tools : Notion, Weights&Biases, Github, Google Drive, MLflow
- 개발 Tools : VScode, Jupyter Lab
- 라이브러리 및 프레임워크 : PyTorch, Streamlit

📁 Dataset

- 피부를 유분, 수분, 주름, 색소, 민감도 5가지 항목으로 나누고, 정도에 따라 0부터 4까지 평정한 피부 데이터



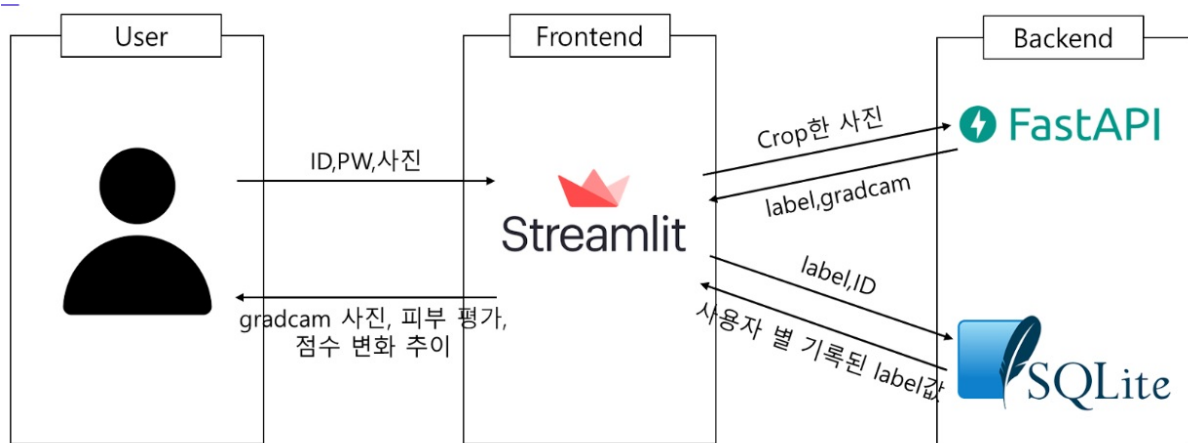
- 각 항목의 일부 라벨이 매우 부족한 imbalance data
- 보안 규정 상 피부 이미지, 데이터 수, 라벨링 기준 등은 공개 불가

👤 평가 Metric

- Penalty Recall (P-Recall)
 - Ordinal data이기 때문에 정답만 맞추는 것 뿐 아니라 정답과 얼마나 비슷하게 예측했는지도 중요함
 - 정답과 인접한 라벨로 예측하면 일정 가중치를 부여한 정답으로 인정하고, 정답과 먼 라벨로 예측하면 패널티를 주어 recall을 계산

	Label 0	Label 1	Label 2	Label 3	Label 4
Prediction 0	1	0.5	-0.5	-1	-1.5
Prediction 1	0.5	1	0.5	-0.5	-1
Prediction 2	-0.5	0.5	1	0.5	-0.5
Prediction 3	-1	-0.5	0.5	1	0.5
Prediction 4	-1.5	-1	-0.5	0.5	1

🧱 Structure



- Frontend
 - User의 ID와 password를 입력하면 개인 피부 사진을 업로드할 수 있음
 - 사진을 업로드하면 피부 평가 점수와 gradcam을 제공
 - 평가 점수가 저장되어 피부 변화 추이를 그래프로 확인 가능
- Backend
 - FastAPI로 모델 서버를 구축하여 평가 점수와 gradcam을 빠르게 계산하여 제공
 - SQLite로 사용자들의 데이터 정보를 저장

Experiments

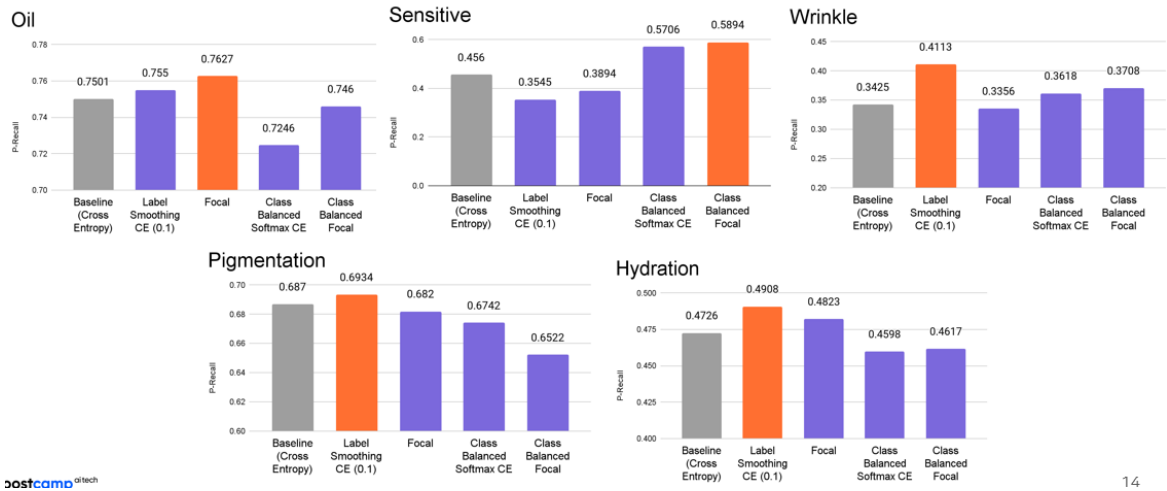
- Model
 - 서비스를 하기 위해 모델의 성능 뿐 아니라 속도와 크기도 고려하여 EfficientNet-B0로 최종 결정

	Parameters	Metric 성능	Training Time
<u>Swin v2</u>	88M	<u>Precall: 0.5331</u>	13m 45s
<u>Swin transformer</u>	28.3M	<u>Precall: 0.5822</u>	16m 01s
<u>Twin</u>	43.8M	<u>Precall: 0.5435</u>	13m 37s
EfficientNet-B4	19M	<u>Precall: 0.4084</u>	9m 56s
EfficientNet-B0	5.3M	<u>Precall: 0.5002</u>	5m 4s

- Loss
 - Class Imbalance가 있을 때 사용하는 다양한 Loss function을 실험

Cross Entropy	label Smoothing CE	Focal	Class Balanced Softmax CE	Class Balanced Focal
다중 분류를 위한 손실 함수 (Baseline)	Soft target으로 바꾸어 정답과 오답간의 score 격차를 줄임	어렵거나 쉽게 오분류되는 케이스에 더 큰 가중치를 주어 불균형 문제를 개선	새로운 케이스를 학습할 때 가중치를 주어 불균형 문제 개선	새로운 케이스를 학습할 때 가중치를 주어 불균형 문제 개선

- 각 항목 별 가장 높은 P-Recall을 달성한 Loss function을 사용



Loss function	Oil	Sensitive	Wrinkle	Pigmentation	Hydration
Baseline (Cross Entropy)	0.7501	0.456	0.3425	0.687	0.4726
Label Smoothing CE (0.1)	0.755	0.3545	◆ 0.4113	◆ 0.6934	◆ 0.4908
Focal	0.7627	0.3894	0.3356	0.682	0.4823
Class Balanced Softmax CE	◆ 0.7246	0.5706	0.3618	0.6742	0.4598
Class Balanced Focal	0.746	◆ 0.5894	0.3708	0.6522	0.4617

• Masked Label Smoothing

- 다른 라벨과의 score 격차를 줄여주는 Label Smoothing의 특성을 적용하여, 인접한 라벨에만 확률을 배분

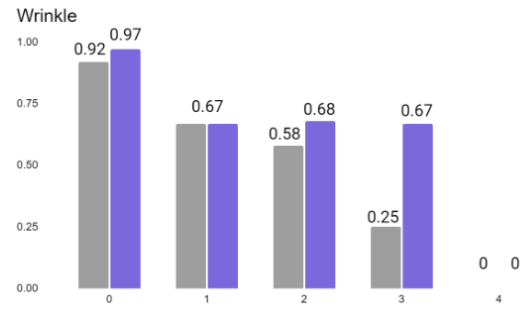
	Label 0	Label 1	Label 2	Label 3	Label 4
Prediction 0	0.9	0.025	0.025	0.025	0.025
Prediction 1	0.025	0.9	0.025	0.025	0.025
Prediction 2	0.025	0.025	0.9	0.025	0.025
Prediction 3	0.025	0.025	0.025	0.9	0.025
Prediction 4	0.025	0.025	0.025	0.025	0.9

→

	Label 0	Label 1	Label 2	Label 3	Label 4
Prediction 0	0.95	0.05	0	0	0
Prediction 1	0.05	0.9	0.05	0	0
Prediction 2	0	0.05	0.9	0.05	0
Prediction 3	0	0	0.05	0.9	0.05
Prediction 4	0	0	0	0.05	0.95

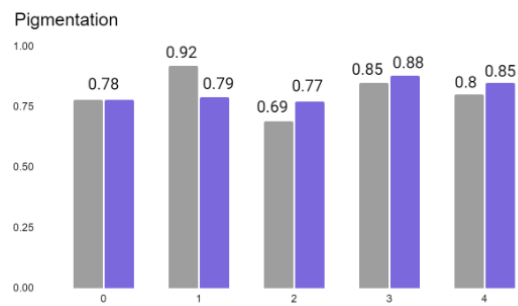
- Wrinkle

	0	1	2	3	4
Label Smoothing	0.92	0.67	0.58	0.25	0
Masked Label Smoothing	0.97	0.67	0.68	0.67	0



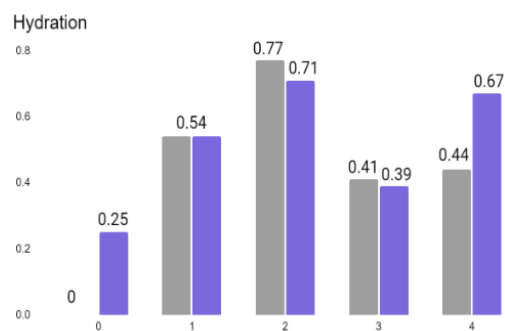
◦ Pigmentation

	0	1	2	3	4
Label Smoothing	0.78	0.92	0.69	0.85	0.8
Masked Label Smoothing	0.78	0.79	0.77	0.88	0.85



◦ Hydration

	0	1	2	3	4
Label Smoothing	0	0.54	0.77	0.41	0.44
Masked Label Smoothing	0.25	0.54	0.71	0.39	0.67



- Masked Label Smoothing 적용 결과, Label Smoothing을 사용했을 때 보다 P-Recall이 전체적으로 고르게 분포하는 것을 볼 수 있다.

• Imbalanced Data Sampler

- 비율이 높은 라벨은 undersampling, 비율이 낮은 라벨은 oversampling하여 학습 (P-Recall)

Oil	Wrinkle	Sensitive	Pigmentation	Hydration
0.5879	0.4113	0.456	0.6934	0.4902

Oil	Wrinkle	Sensitive	Pigmentation	Hydration
♦ 0.6207	♦ 0.4177	♦ 0.5551	0.6856	0.4701

- Oil,Wrinkle,Sensitive의 경우 P-Recall이 증가

- Augmentation

- Augmentation 실험은 다음 두 가지 관점에서 진행 (P-Recall)

1. 주어진 데이터들의 크기와 종횡비가 다르므로 해상도와 비율에 대한 영향을 실험

Augmentation	Oil	Wrinkle	Sensitive	Pigmentation	Hydration
Resize (512,512)(Baseline)	0.5879	0.4113	0.2702	0.6934	0.4908
Resize (1024,1024)	0.7552	0.3800	0.7547	♦ 0.8018	0.2022
CenterCrop (512,512)	0.6286	0.2765	0.3815	0.6671	0.4686
RandomCrop (512,512)	0.7516	♦ 0.4480	0.6782	0.7460	0.4745
RandomResizedCrop (512,512)	0.6580	0.3283	0.4560	0.7324	♦ 0.5125
RandomResizedCrop (1024,1024)	♦ 0.7631	0.4213	♦ 0.7568	0.7803	0.2181

2. 피부를 촬영하는 환경 차이를 고려한 실험

	Oil	Wrinkle	Sensitive	Pigmentation	Hydration
RGBShift	♦ 0.6197	0.4037	♦ 0.2801	♦ 0.7078	♦ 0.5385
ISONoise	♦ 0.6416	0.3651	♦ 0.4452	0.6920	♦ 0.5035
OpticalDistortion	♦ 0.6441	0.3613	♦ 0.3407	♦ 0.7151	0.4390
RandomBrightness	♦ 0.6115	♦ 0.4175	0.2681	0.6904	0.4653
HorizontalFlip	♦ 0.6516	0.3642	♦ 0.2733	0.6768	0.4637

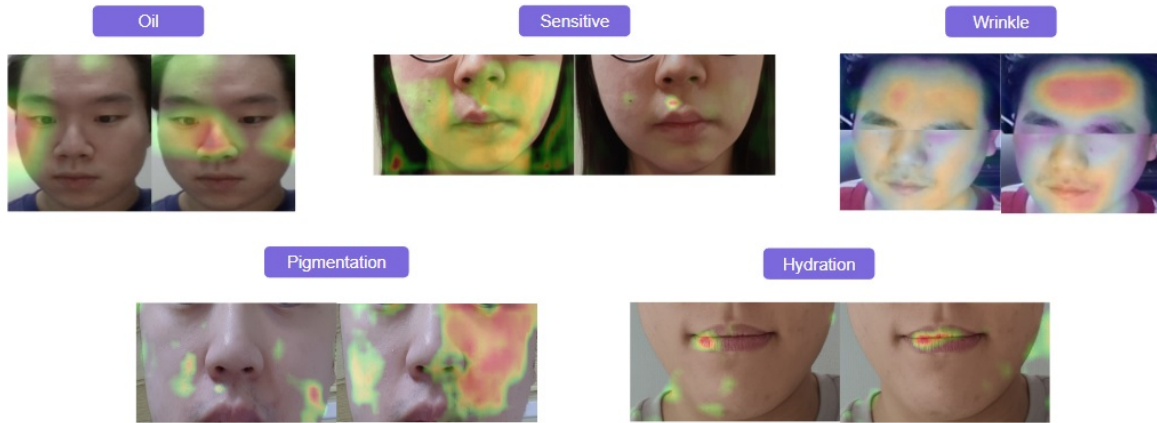
Result

- 최종 결과

- 각 항목별로 P-Recall 향상에 영향을 주었던 기법들을 모두 적용하여 최종 모델 학습

	Oil	Wrinkle	Sensitive	Pigmentation	Hydration
Loss function	focal	Masked Label Smoothing	Class Balanced Focal	Masked Label Smoothing	Masked Label Smoothing
Augmentations	RandomResizedCrop, RGBShift, ISONoise, OpticalDistortion, RandomBrightness, Horizontal Flip	RandomCrop, RGBShift, RandomBrightnessContrast	Resize, ISONoise, Horizontal Flip	Resize, RandomCrop, RGBshift, ISONoise	RandomResizedCrop, RGBshift, ISONoise
Baseline P-Recall	0.5879	0.4113	0.4560	0.6934	0.4908
P-Recall	0.7674	0.6524	0.7705	0.8278	0.5859

- baseline -> 최종 모델 CAM 비교



- 시연 영상 (<https://youtu.be/nDWqPPHq6UQ>)

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0405729d-41e6-46bd-82ef-d858117d2b19/최종시연영상.mp4>

- 아쉬웠던 점
 - multi class label을 이용해 1개 모델로 학습을 실험해보지 못해 아쉬웠다.
 - 얼굴을 인식하는 segmentation model을 이용하여 데이터 전처리를 했다면 CAM이 배경을 잡는 일을 줄일 수 있었을 것이다.
 - 좀 더 다양한 augmentation 실험을 진행해보지 못했다.
 - 디자인에 소질이 없어서 데모 사이트 꾸미는것에 어려움을 많이 겪었습니다.