Related Works

선행 연구 조사

❖ 김용수 외(2023). 합성곱 신경 회로망 모델을 활용한 흑색종 피부암 진단. [3]

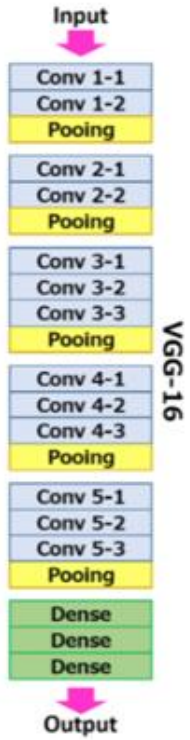


그림 1. VGG16의 전체 구조

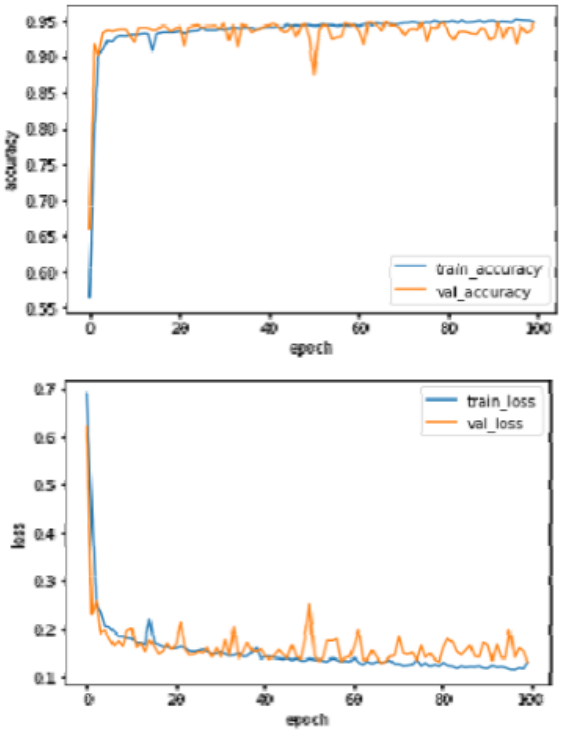


그림 2. epoch에 따른 VGG16의 정확도 변화

표 1. VGG16의 실험 결과

Classfication evalution metrics / Class	Accuracy	Precision	Recall
Negative	0.94	0.94	0.94
Positive		0.94	0.94

- 김용수 외(2023)는 CNN을 활용해 흑색종 피부암 진단 모델을 구현하여 높은 분류 정확도를 보였으나, 해당 모델은 병변의 정량적 특징을 분석하는데 한계가 있음.
- 본 프로젝트에서는 이러한 한계를 보완하여, 병변의 시각적 특징을 정량적으로 추출하고 시각화 기능을 포함시켜 실제 사용자와 의료진에게 실용적인 진단 정보를 제공하는 것임.

Data Preprocessing

데이터 분할

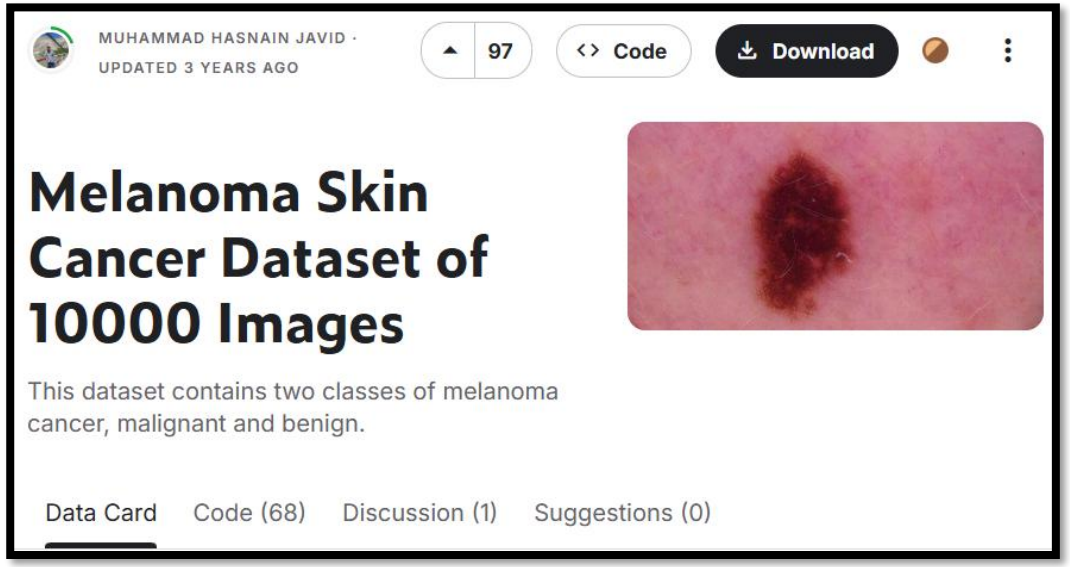
1) 데이터셋 설명

Melanoma Skin Cancer Dataset of 10000 Images. (Kaggle) [4]

표 2. Melanoma Skin Cancer Dataset of 10000 Images 데이터셋 구성

	benign	malignant
train	5,000	4,605
test	500	500

*** 부가적인 csv 파일은 제공하지 않음**



데이터 분할(2)

2) 데이터셋 재구성을 위한 데이터 분할

흑색종(malignant) 1,500장 + 비흑색종(benign) 4,500장 = 총 6,000장
분할 비율 : train 70% / val 15% / test 15%

표 3. 데이터셋 재구성을 위한 데이터셋 설정

	흑색종(malignant)	비흑색종(benign)	전체 샘플 수
Train (70%)	1,050	3,150	4,200
Validation (15%)	225	675	900
Test (15%)	225	675	900
Total	1500	4500	6000

EDA

1) 클래스 분포 분석

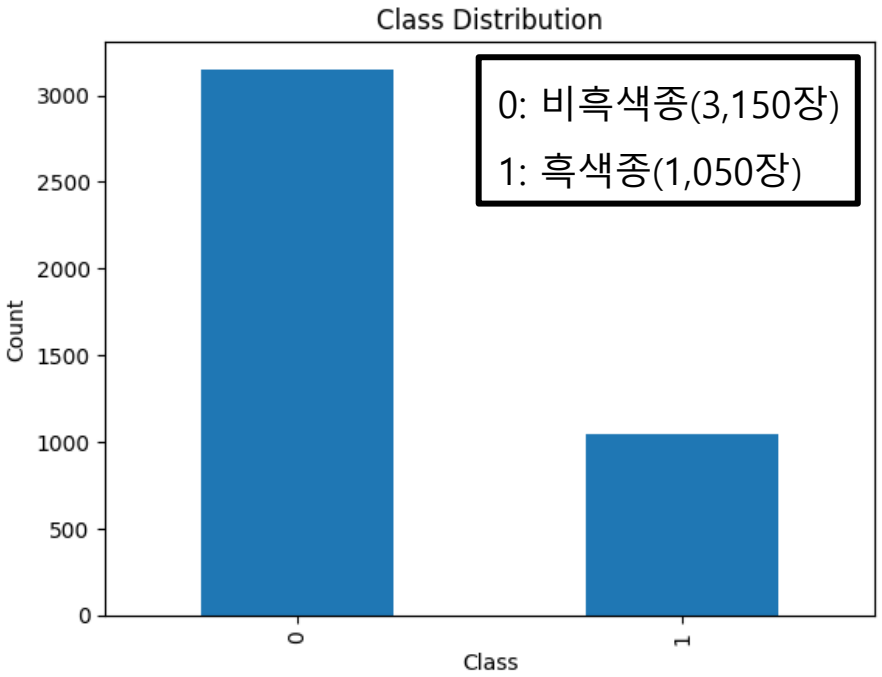
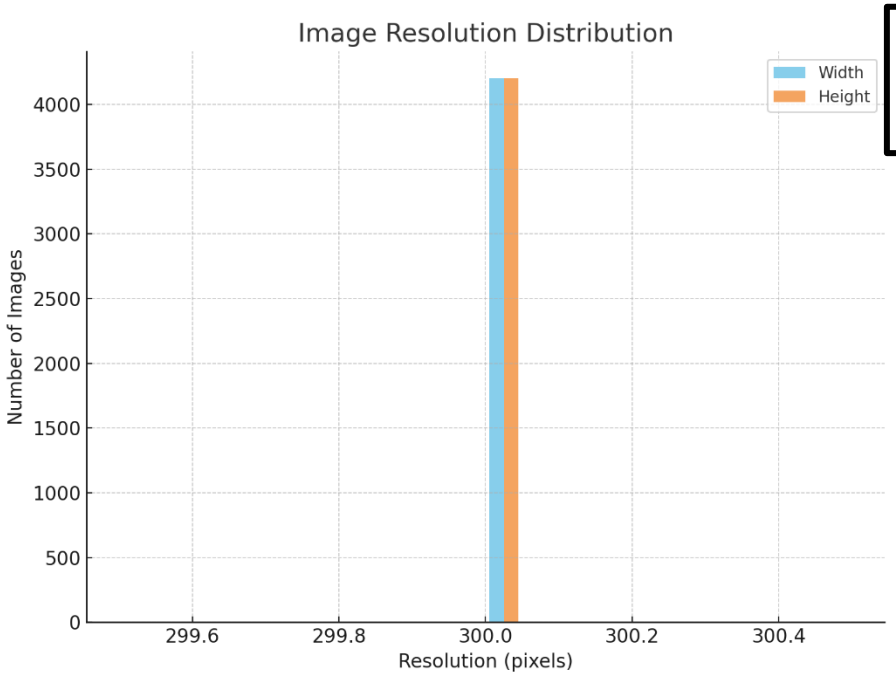


그림 3. train 클래스 분포

3:1 비율의 클래스 불균형
→ 대응 방안 필요

2) 이미지 크기 및 해상도 분석



X축: 이미지의 너비와 높이
Y축: 해당 해상도의 이미지 개수

그림 4. train 이미지 해상도 분포

이미지 4,200장 해상도: 300x300
→ 224x224 해상도로 보정 필요

EDA(2)

3) 채널별 색상 분포 분석

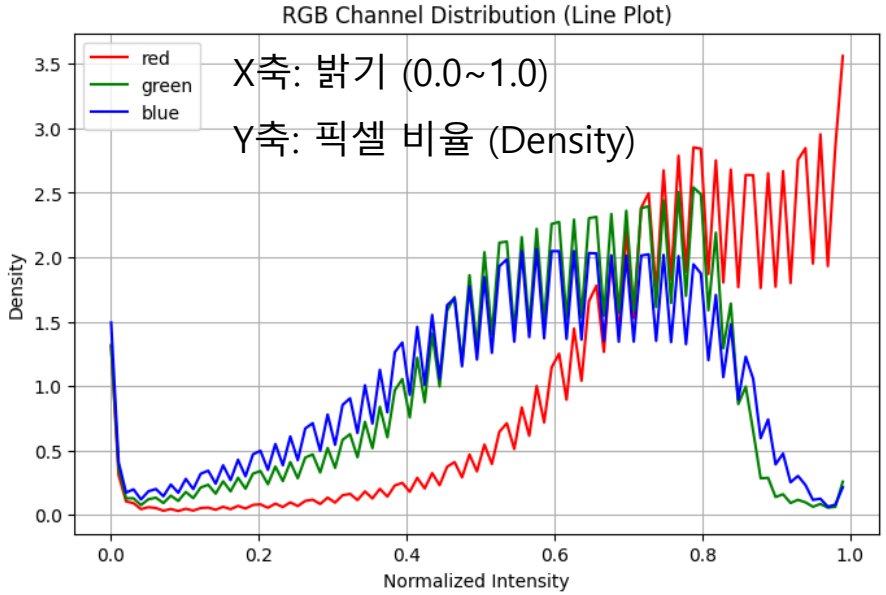


그림 5. train 채널별 색상 분포

모델 학습 시 정규화를 위한 기준값 설정
RGB Mean : [0.76025333 0.5931325 0.56823859]
RGB Std : [0.19433735 0.19554713 0.2141679]

피부 + 병변 대비
→ 전체적으로 **붉은색 계열**이 **더 강함/밝음**

4) 병변 위치 및 형태 분석

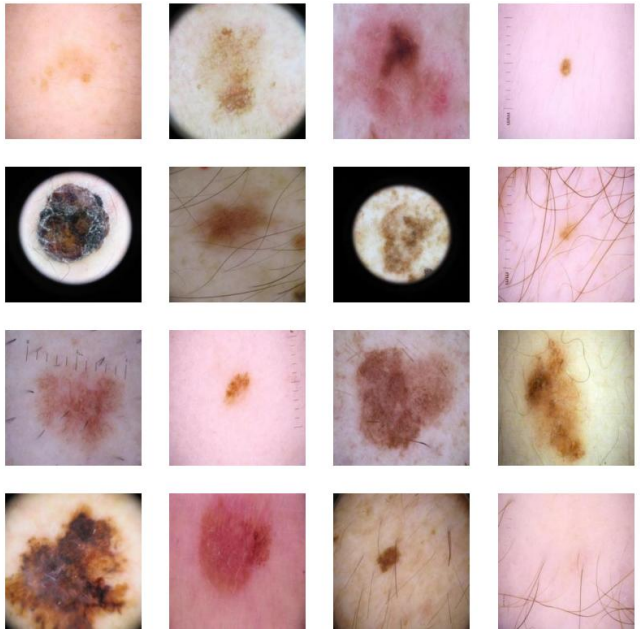


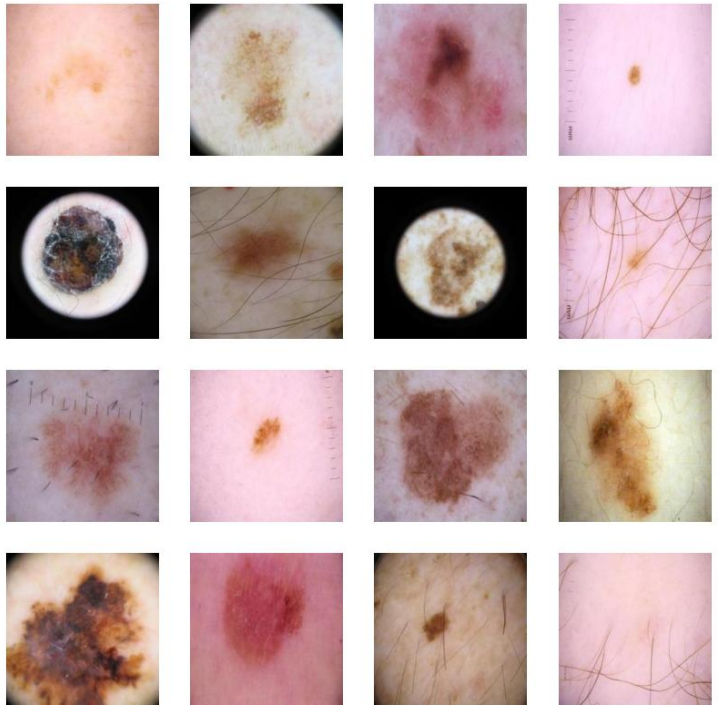
그림 6. train 병변 위치 및 형태 다양성 예시

- 중심 위치 여부:
  대부분 **중앙 위치**
- 형태:
  둥금, 퍼짐, 경계 불분명 다양
- 크기:
  매우 작은 병변,
  큰 병변(전체 이미지의 1/3 이상) 존재
- 경계 및 색상 대비:
  배경 피부색과 유사해 경계 흐린 병변,
  강한 색 대비로 명확하게 구분된 병변

크기, 형태, 경계 대비 등
다양한 시각적 변이 고려
→ 설계 + 증강 필요

EDA(3)

5) 데이터 다양성 및 Augmentation 필요성 평가



- 병변:
  형태·크기 등
  병변 다양
- 배경:
  색상·조명·위치 등
  유사 패턴 반복

그림 7. Augmentation 필요성을 시사하는 시각적 다양성과 반복 예시

일반화 성능 확보 필요 → Augmentation 필요

6) 필요 항목

- 이미지 **리사이즈 (300x300 → 224x224)**
- 일반화 성능 확보를 위한 **Augmentation**
- 클래스 불균형 완화를 위한 **학습 전략**

전처리

1) 전처리 적용 코드

- 이미지 리사이즈 (300x300 → 224x224)
- 텐서 변환
- 정규화

모델 학습 시 정규화를 위한 기준값 설정
RGB Mean : [0.76025333 0.5931325 0.56823859]
RGB Std : [0.19433735 0.19554713 0.2141679]

```
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.7603, 0.5931, 0.5682],
                         std=[0.1943, 0.1956, 0.2142])
])
```

Augmentation

건양대학교 KONYANG UNIVERSITY

1) 조합 설계

- **EDA 기반 Augmentation 필요 항목:**

크기, 형태, 경계, 색상, 조명에 대한 **Augmentation 필요**

→ 이러한 이유로 Augmentation 선정함

(1) RandomHorizontalFlip: 흑색종은 좌우 대칭성이 없음 → 좌우 반전은 데이터 다양성 확보에 유효

(2) RandomRotation(degrees): 병변 방향은 진단에 무관 → 회전은 일반화 향상에 도움

(3) ColorJitter: 밝기/대비 조정은 조명 환경 변화에 대한 강건성 확보에 효과적

(4) RandomAffine: 이동/스케일을 병변 위치 다양성 확보 가능

(5) RandomEqualize: 전체 대비 균등화, 피부톤 정규화 효과 가능

Augmentation(2)

1) 조합 설계

표 4. Augmentation 조합 설계

조합명	적용 증강 기법	설명
F1	**RandomHorizontalFlip**	기본적인 **시각 다양성 확보**를 위해 가장 많이 사용
F2	**RandomRotation**	병변의 **방향 변화에 견딜 수 있도록** 회전 단독 효과 확인
F3	**ColorJitter**	**조명/대비 변화에 대한 강건성** 단독 효과 확인
F4	**RandomAffine**	**위치 이동이나 스케일 변화**에 대한 민감도 평가
F5	**RandomEqualize**	**색소 대비 변화**가 모델 학습에 미치는 영향 단독 측정
C1	**Flip + Rotation**	**의료 영상**에서 가장 기본적이고 필수적인 **구조 다양화** 조합
C2	**Flip + Jitter**	**촬영 환경(밝기, 대비 등) 변화 대응**을 위한 조명 중심 조합
C3	**Rotation + Jitter**	**Flip 없이도 일반화 성능이 확보**되는지 비교군으로 사용
C4	**Flip + Rotation + Jitter**	**구조 + 조도 변화**에 안정적으로 반응하는지 확인
C5	**Flip + Rotation + Affine**	**복합 구조 다양화**에 따른 일반화 성능 향상 확인
C6	**Flip + Affine + Equalize**	**색 대비 + 구조 변화의 실험적 대안 조합**으로 성능 보완 가능성 평가
C7	**Flip + Rotation + Jitter + Equalize**	**색소 분포를 강조**하면서도 과하지 않은 복합 증강 조합
C8	**모든 5개**	모든 증강을 한 번에 적용했을 때 **과적합 여부 및 최대 효과 테스트**

Augmentation(3)

2) 조합 성능 비교 및 조합 선정

표 5. test 데이터셋에 대한 Augmentation 조합 성능 표

조합명	Accuracy	Recall	Precision	F1-score
Baseline	0.9567	0.8756	0.9471	0.9099
F1	0.9556	0.8400	0.9793	0.9043
F2	0.9444	0.8933	0.8855	0.8894
F3	0.9533	**0.9067**	0.9067	0.9067
F4	0.9511	0.8889	0.9132	0.9009
F5	0.9511	0.8800	0.9209	0.9000
C1	0.9179	0.8143	0.8507	0.8321
C2	**0.9644**	0.8667	0.9898	**0.9242**
C3	0.9522	0.8844	0.9213	0.9025
C4	0.9556	**0.9067**	0.9148	0.9107
C5	0.9567	0.8622	0.9604	0.9087
C6	0.9444	0.8800	0.8959	0.8879
C7	0.9600	0.8622	0.9749	0.9151
C8	0.9567	0.8311	**0.9947**	0.9056

Augmentation(4)

2) 조합 성능 비교 및 조합 선정

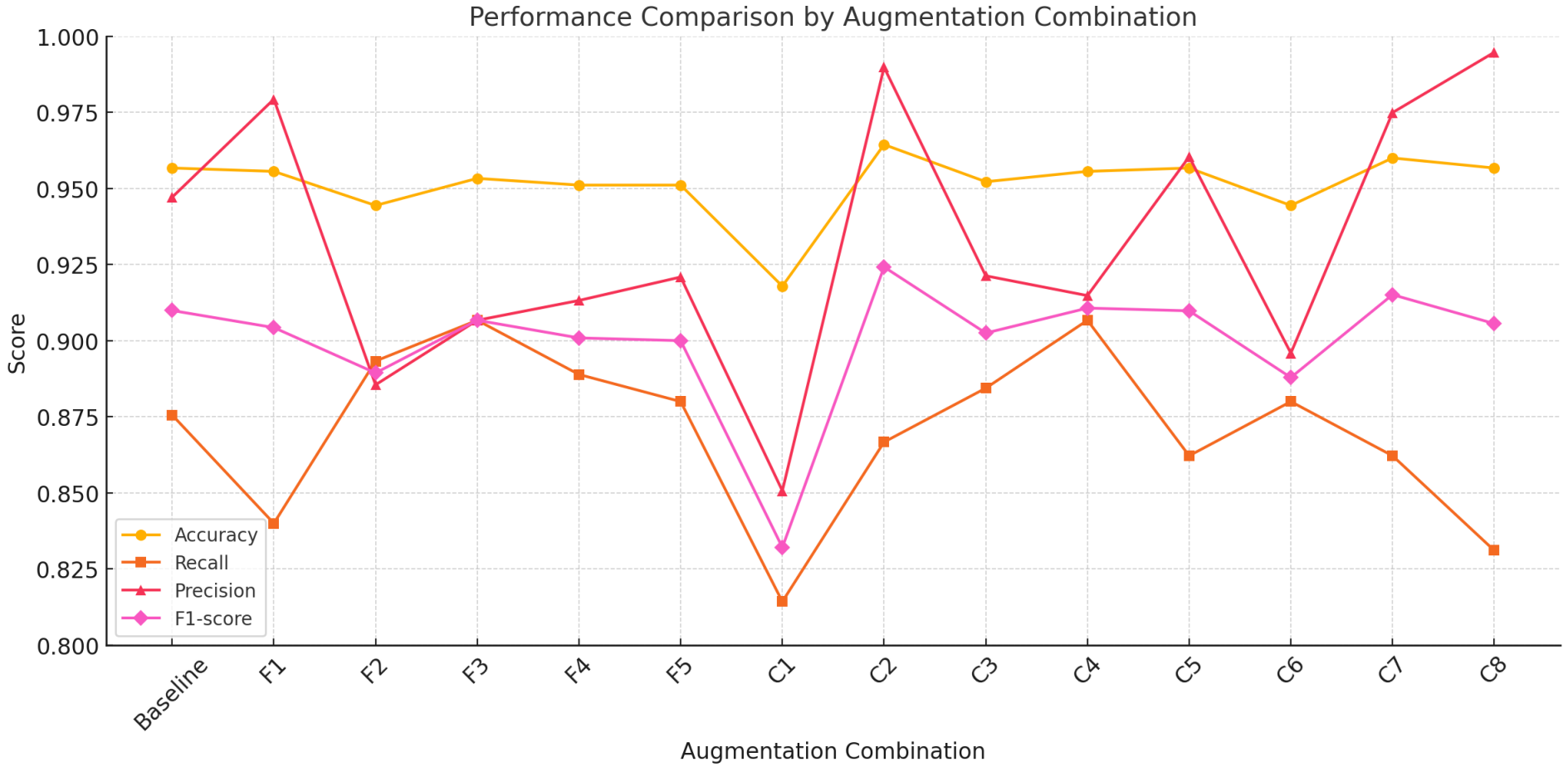


그림 8. test 데이터셋에 대한 Augmentation 조합 성능 시각화

Augmentation(5)

건양대학교 KONYANG UNIVERSITY

3) Balanced Sampler(Weighted Sampling)

- EDA 기반 Augmentation 필요 항목:

클래스 불균형 완화를 위한 학습 전략 필요

- Balanced Sampler:
  - 클래스 불균형 해소 목적의 샘플링 전략 전체

- Weighted Sampling:
  - Balanced Sampler의 일종
  - 클래스 불균형을 해결하기 위해 클래스별 가중치 기반 확률 샘플링

Augmentation(6)

4) Augmentation 조합별 Weighted Sampling 적용 및 성능 비교

표 6. test 데이터셋에 대해 Augmentation 조합별 Weighted Sampling 적용한 Augmentation 조합 성능 표

조합	Accuracy	Recall	Precision	F1-score
Basline + WS	0.9578	0.8667	0.9606	0.9112
F1 + WS	0.9511	0.8356	0.9641	0.8952
F2 + WS	0.9533	0.9200	0.8961	0.9079
F3 + WS	0.9611	0.9111	0.9318	0.9213
F4 + WS	0.9567	0.9200	0.9079	0.9139
F5 + WS	0.9544	0.8533	0.9600	0.9035
C1 + WS	0.9422	0.9156	0.8619	0.8879
C2 + WS	0.9611	0.8756	0.9657	0.9184
C3 + WS	0.9611	0.8800	0.9612	0.9188
C4 + WS	**0.9656**	0.8933	**0.9663**	**0.9284**
C5 + WS	0.9300	**0.9333**	0.8140	0.8696
C6 + WS	0.8898	0.8743	0.9015	0.8877
C7 + WS	0.9533	0.8756	0.9336	0.9037
C8 + WS	0.9589	0.9111	0.9234	0.9172

Augmentation(7)

4) Augmentation 조합별 Weighted Sampling 적용 및 성능 비교

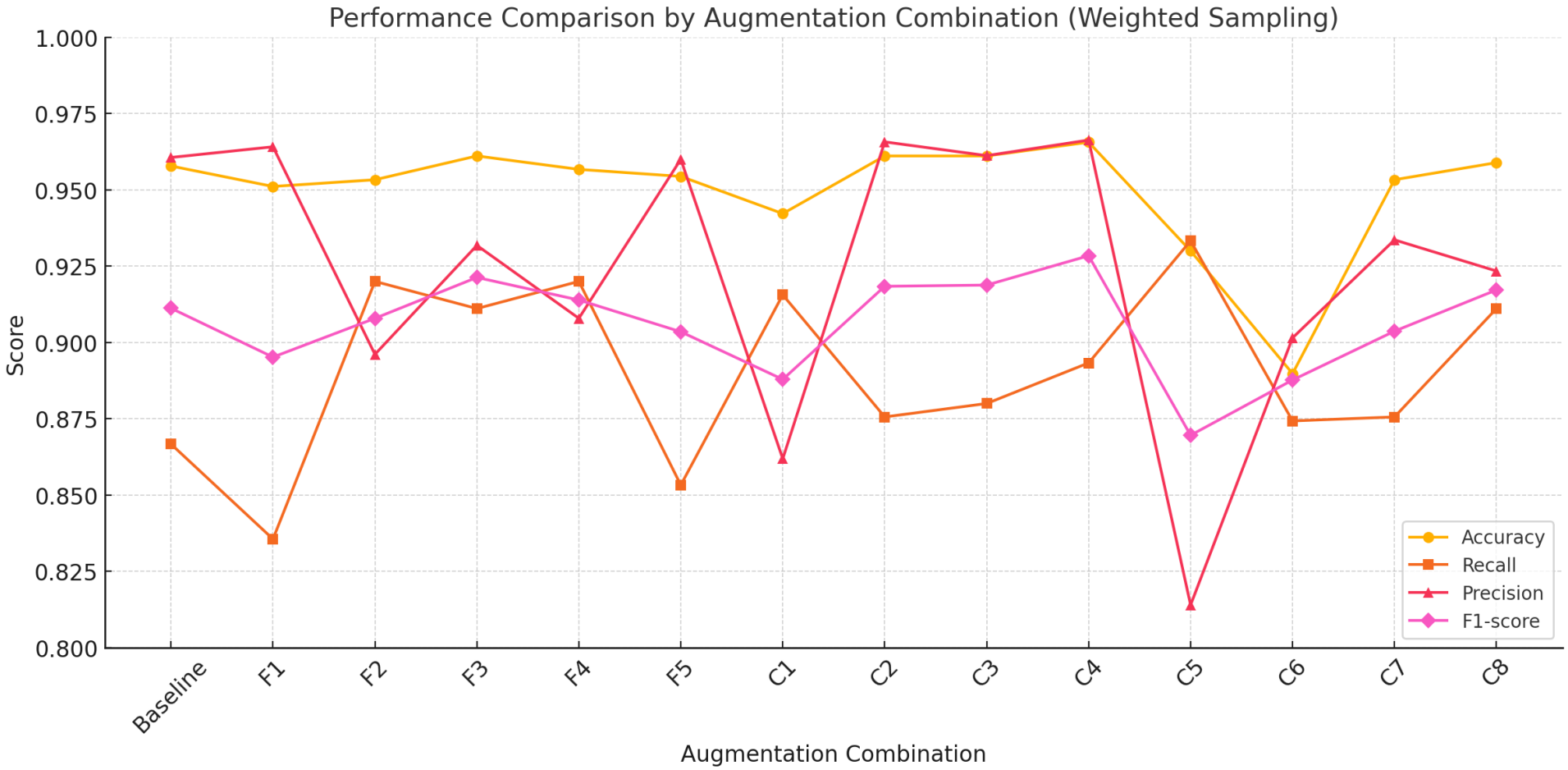


그림 9. test 데이터셋에 대해 Augmentation 조합별 Weighted Sampling 적용한 Augmentation 조합 성능 시각화

Augmentation(8)

건양대학교 KONYANG UNIVERSITY

5) Weighted Sampling 사용 여부 결정 및 최종 조합 선정

표 7. Augmentation 최종 조합 선정

조합명	구성	Accuracy	Recall	Precision	F1-score
C2	**Flip + Jitter**	0.9644	0.8667	0.9898	0.9242
C4 + WS	**Flip + Rotation + Jitter + Weighted Sampling**	**0.9656**	**0.8933**	0.9663	**0.9284**

Weighted Sampling 적용이 불균형 클래스 문제에 긍정적 기여를 했다고 판단 가능

따라서, Weighted Sampling 사용

Model Architecture

모델 선정

건양대학교 KONYANG UNIVERSITY

3. 흑색종 이미지 분류 모델 선정

표 8. 딥러닝 모델별 장점 및 한계

모델	주요 장점	모델 한계
ResNet [5]	잔차 연결, 학습 안정성 우수	feature reuse 제한, 고해상도 병변 정보 누락
EfficientNet [6]	연산 효율성 탁월, 파라미터 최적화	미세 병변 특징 추출 한계, 적은 데이터에 과적합
VGG [7]	단순한 구조, 의료 영상 연구에서 자주 쓰임	파라미터 수 많고 연산량 큼, 깊은 정보 흐름 제한
DenseNet [8]	feature reuse, gradient 흐름 우수 → 미세 병변에 강함	연산량 상대적으로 많음
ViT [9]	시야 범위 넓음, 표현력 뛰어남	대용량 데이터 필요, 소규모 데이터에서 불안정
ConvNeXt [10]	ViT 구조 차용한 최신 CNN, 일반화/효율성 균형	최신 구조, 환경/튜닝 고려 필요

모델 선정(2)

건양대학교 KONYANG UNIVERSITY

표 9. Model 학습 전략표

항목	내용
Seed 고정	42로 고정
평가 지표	Accuracy, Recall, Precision, F1-score(소수점 4자리까지)
Epoch	100
Batch_size	32
EarlyStopping	Patience 3
Model Checkpoint	매 10 epoch마다 best 모델 저장

모델 선정(3)

건양대학교 KONYANG UNIVERSITY

표 10. 모델 학습 결과 표

Model	Accuracy	Recall	Precision	F1-score
Resnet50	0.9489	0.8533	0.9366	0.8930
EfficientNet	0.9589	0.8933	0.9393	0.9157
DenseNet	0.9500	0.8667	0.9286	0.8966
ConvNeXt	**0.9611**	**0.9156**	0.9279	**0.9217**
VGG	0.9533	0.8622	**0.9463**	0.9023
InceptionV3	0.9400	0.8844	0.8767	0.8805

모델 선정(4)

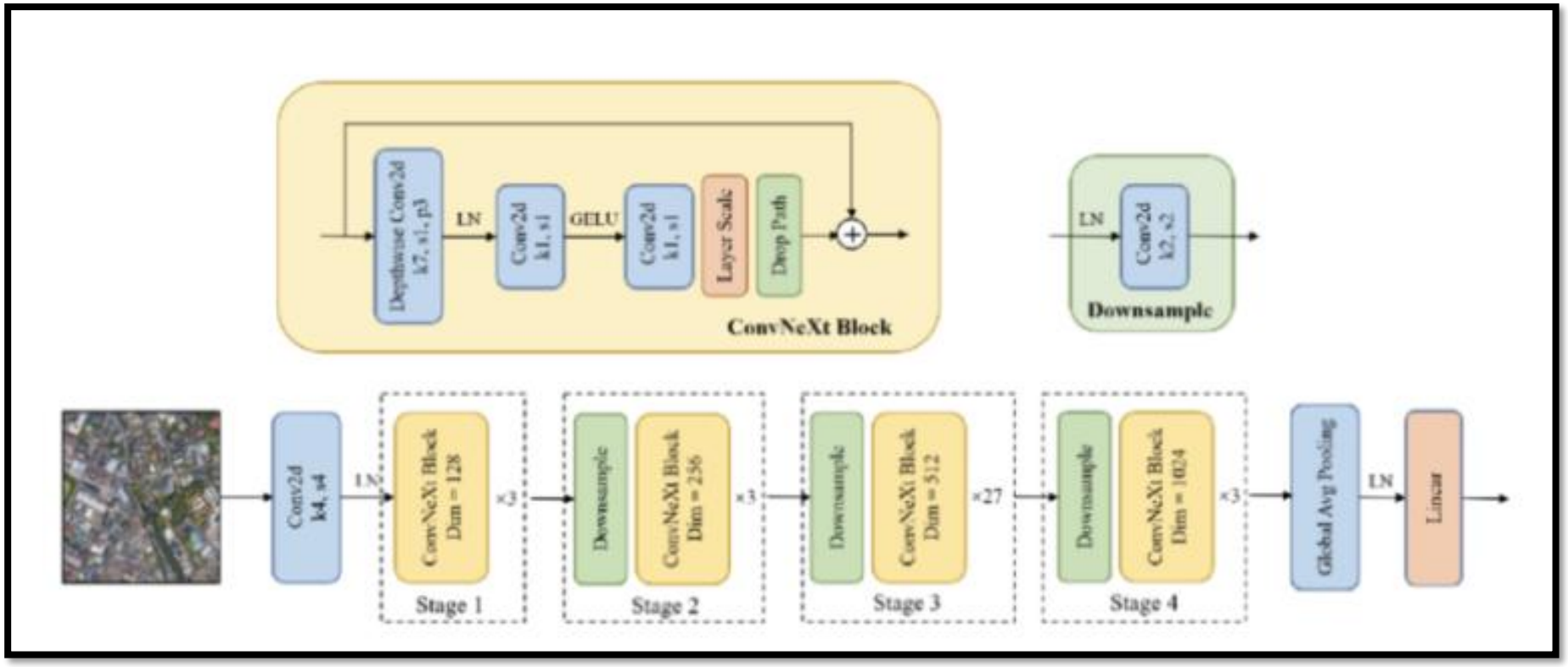
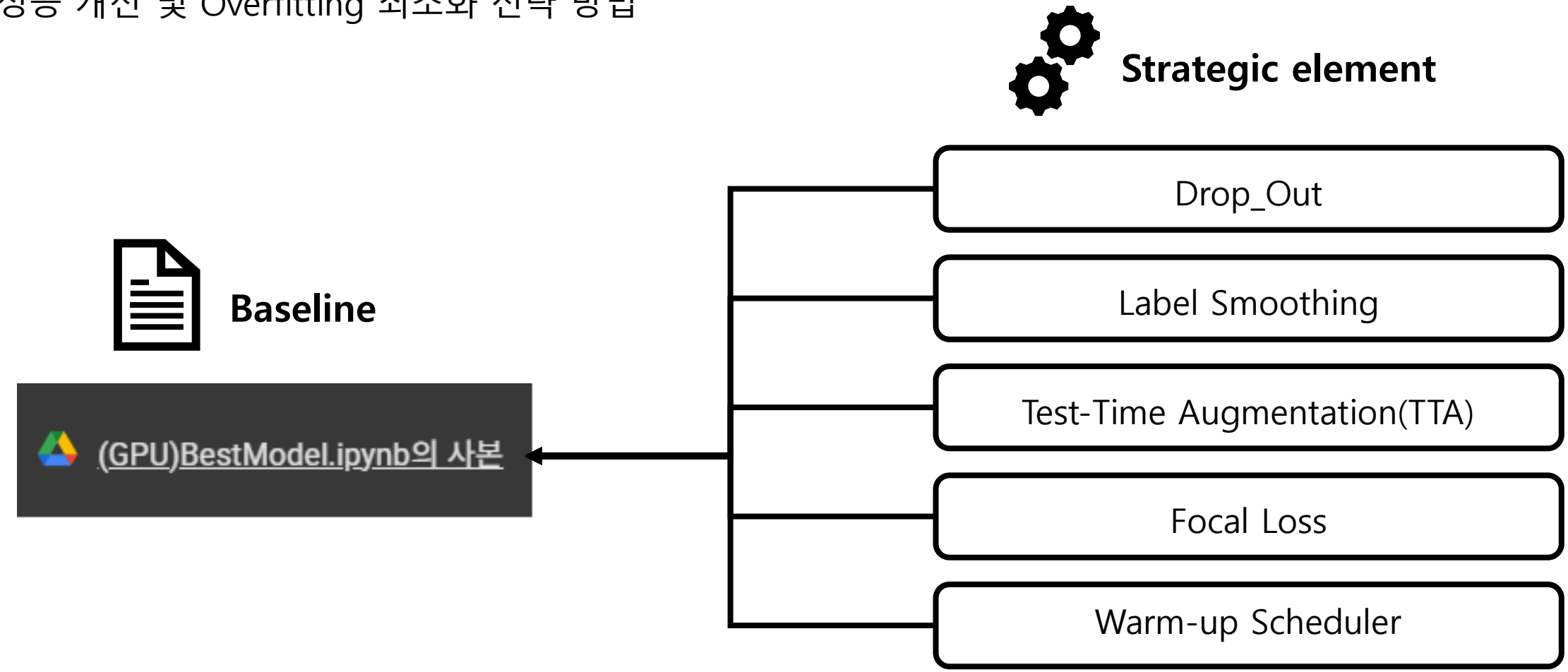


그림 10. ConvNeXt 모델 구조

- **ConvNeXt는 2022년 Meta AI에서 발표한 고성능 CNN 모델로, Vision Transformer의 구조적 장점을 CNN에 접목하여 성능을 크게 개선한 모델**
- **기존 CNN 구조를 Transformer Style로 재설계함으로써, CNN의 효율성과 Transformer의 표현력을 모두 결합한 모델**

Overfitting 해결을 위한 전략

❖ 성능 개선 및 Overfitting 최소화 전략 방법



- **기존 Best_Model에 없던 전략적 요소들을 설계하고, 구성하여 오버피팅을 최소화 시키는 방향 설정**

Overfitting 해결 전략을 위한 전략(2)

건양대학교 KONYANG UNIVERSITY

❖ 성능 개선 및 Overfitting 최소화 전략 요소

표 11. 전략적 요소 지표

Strategic Element	Method	Effect
Drop_Out	ConvNeXt의 Classifier에 nn.Dropout(p=0.3) 삽입	모델의 일반화 성능 향상, 과적합 감소
Label Smoothing	CrossEntropy -> Label Smoothing 적용	예측을 부드럽게 하여 과적합 완화 및 일반화
Test-Time Augmentation(TTA)	테스트 시 여러 변형 이미지를 예측에 평균 적용	예측의 안정성과 강인성 증가
Focal Loss	Hard Example에 가중치를 부여하는 Focal Loss 적용	클래스 불균형 문제 대응
Warm-up Scheduler	학습 초기 작은 학습률로 시작해 점진적으로 증가	초반 학습 불안정 해소

Performance Evaluation

평가 지표

건양대학교 KONYANG UNIVERSITY

		Predicted: Positive	Predicted: Negative
Actual	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TP (True Negative)

그림 11. Confusion Matrix

Recall
=TP/(TP + FN)

Precision
=TP/(TP + FP)

Accuracy
=TP + TN/(TP + TN + FP + FN)

F1-Score
= 2 x (Recall x Precision)/(Recall + Precision)

*** 평가지표를 Accuracy, Recall, Precision, F1-score로 선정 이유**

- **Accuracy(정확도):** 모델이 전체적으로 얼마나 잘 맞췄는지를 보여주는 기본 지표.
하지만 데이터가 불균형할 경우, 정확도만으로는 모델의 진짜 성능을 파악하기 어렵기 때문에 다른 지표들도 함께 찾아봄
- **Recall(재현율):** 흑색종을 실제로 잘 찾아냈는지를 보여주는 지표.
- **Precision(정밀도):** 모델이 흑색종이라고 판단한 것 중에서 진짜 흑색종이 얼마나 정확했는지를 나타내는 지표.
- **F1-score:** 위 두 가지 지표의 균형을 보여주는 점수. 둘 사이 균형이 잘 맞는 모델이 가장 바람직하다고 판단하여 선정

모델 평가

건양대학교 KONYANG UNIVERSITY

표 12. Overfitting 최소화 전략의 종합적 결과(Test)

Model	Epochs	Loss	Accuracy	Recall	Precision	F1-Score
Baseline	7	0.1352	0.9511	0.8711	0.9289	0.8991
Drop_Out	10	0.1329	0.9478	0.8622	0.9238	0.8920
Label Smothing	15	0.2822	0.9611	0.8978	0.9439	0.9203
TTA	10	0.1336	0.9489	0.8844	0.9087	0.8964
FocalLoss(Alpha=0.25)	9	**0.0097**	0.9478	0.8756	0.9120	0.8934
FocalLoss(Alpha=0.5)	10	0.0191	0.9533	0.8978	0.9140	0.9058
FocalLoss(Alpha=1.0)	10	0.0382	0.9533	0.8978	0.9140	0.9058
FocalLoss(Alpha=2.0)	10	0.0765	0.9533	0.8978	0.9140	0.9058
Warm-up(Epochs=5)	6	0.1520	0.9456	0.8933	0.8894	0.8914
Warm-up(Epochs=10)	5	0.1233	**0.9622**	0.8933	**0.9526**	**0.9220**
Warm-up(Epochs=20)	5	0.1420	0.9533	0.8889	0.9217	0.9050
FocalLoss(Alpha=0.35, Gamma=2.5)	10	0.0100	0.9544	**0.9022**	0.9144	0.9083

모델 평가(2)

❖ 각 요소별 성능 비교 및 결과

표 13. Warm-up Scheduler(Epochs = 10) 모델의 성능 평가표

Strategic Elements	Loss	Accuracy	Recall	Precision	F1-Score
Train	0.0794	0.9717	0.9655	0.9772	0.9713
Validation	0.1583	0.9533	0.8933	0.9178	0.9054
Test	**0.1233**	**0.9622**	**0.8933**	**0.9526**	**0.9220**

Result & Visualization

모델 시각화

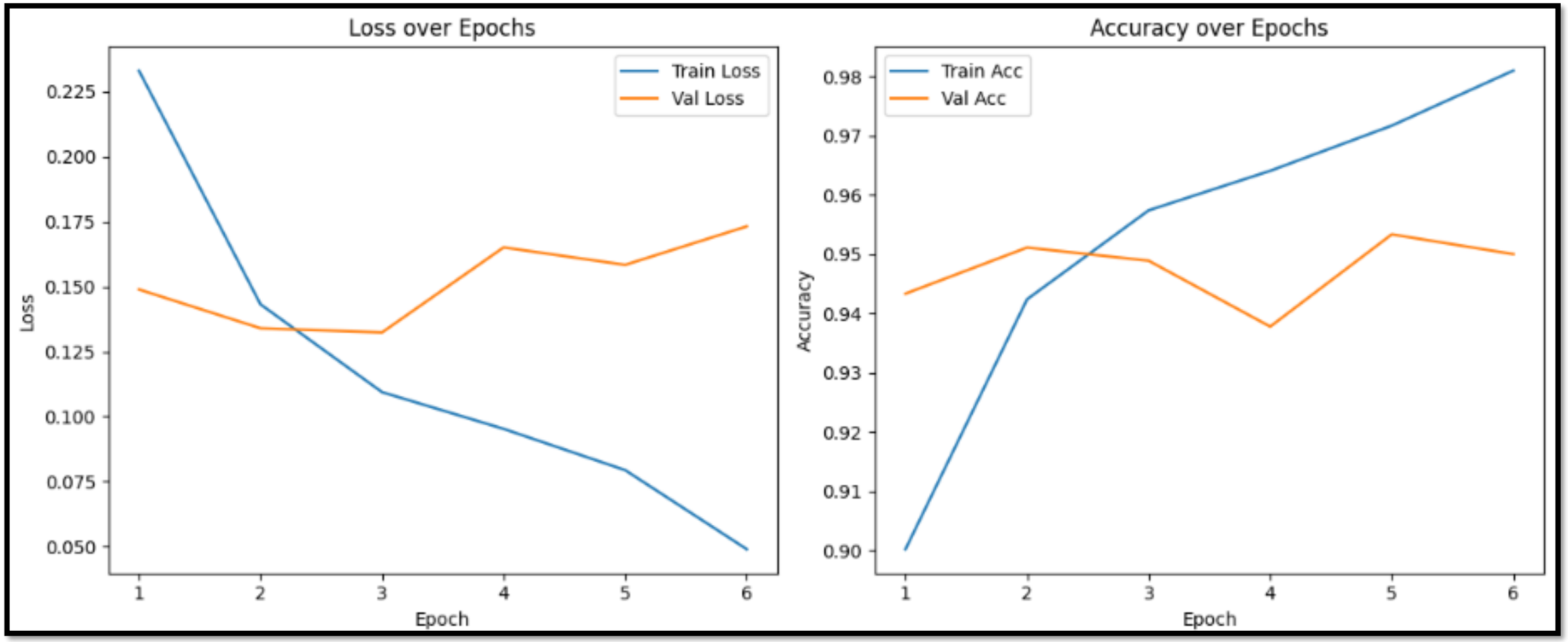


그림 12. Warm-up Scheduler(Epochs = 10), Train/Validation의 Accuracy, Loss 그래프

모델 시각화(2)

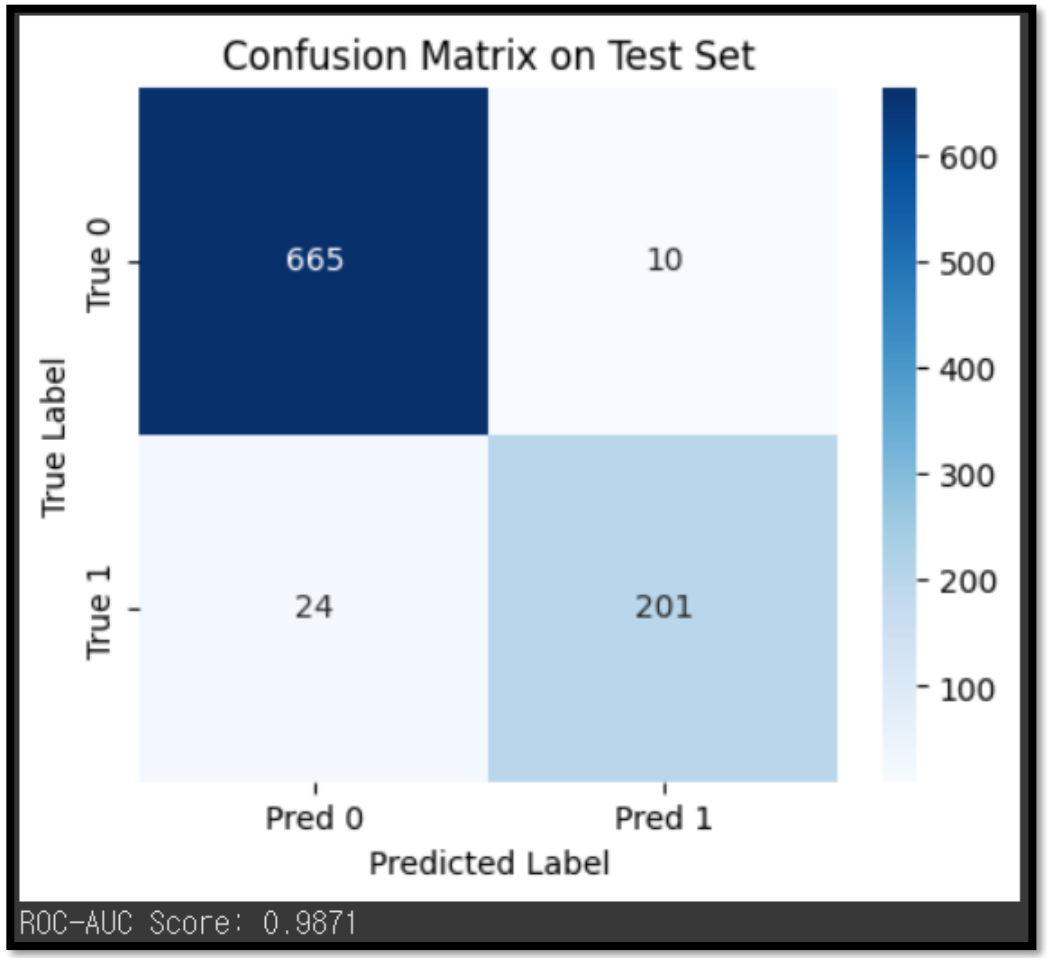


그림 13. Warm-up Scheduler(Epochs = 10), [Test] Confusion Matrix, ROC-AUC 등 정량적 평가 지표 및 분류 결과

모델 시각화(3)

❖ [Test] Confusion Matrix, ROC-AUC 등 정량적 평가 지표 및 분류 결과

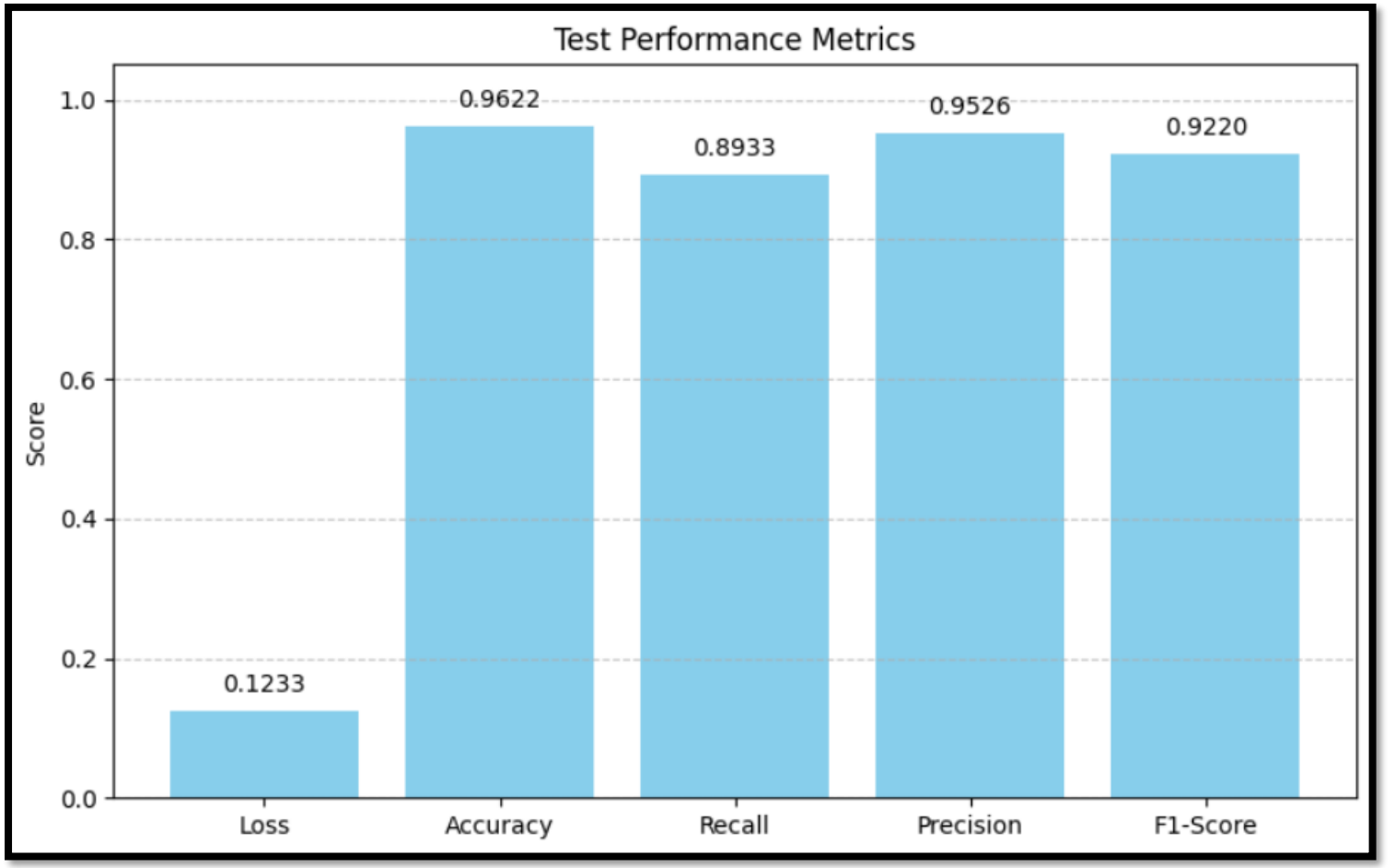


그림 14. Warm-up Scheduler(Epochs = 10), Test Performance Metrics 결과 그래프

Web Development

Overview

건양대학교 KONYANG UNIVERSITY

- 앱 이름: MelaScan (Melanoma + Scan)

- 목표: 병변의 특징 및 흑색종 여부를 분석하여 예측하고,
  사용자에게 시각화된 결과와 위험도까지 제공하는 딥러닝 기반 Flask 웹&앱

- 주요 기능:
  - 사용자가 업로드 한 이미지 기반으로 진단
  - AI 모델 추론 및 병변 특징 추출
  - 결과 시각화 및 저장 기능 구현

병변 특징 추출

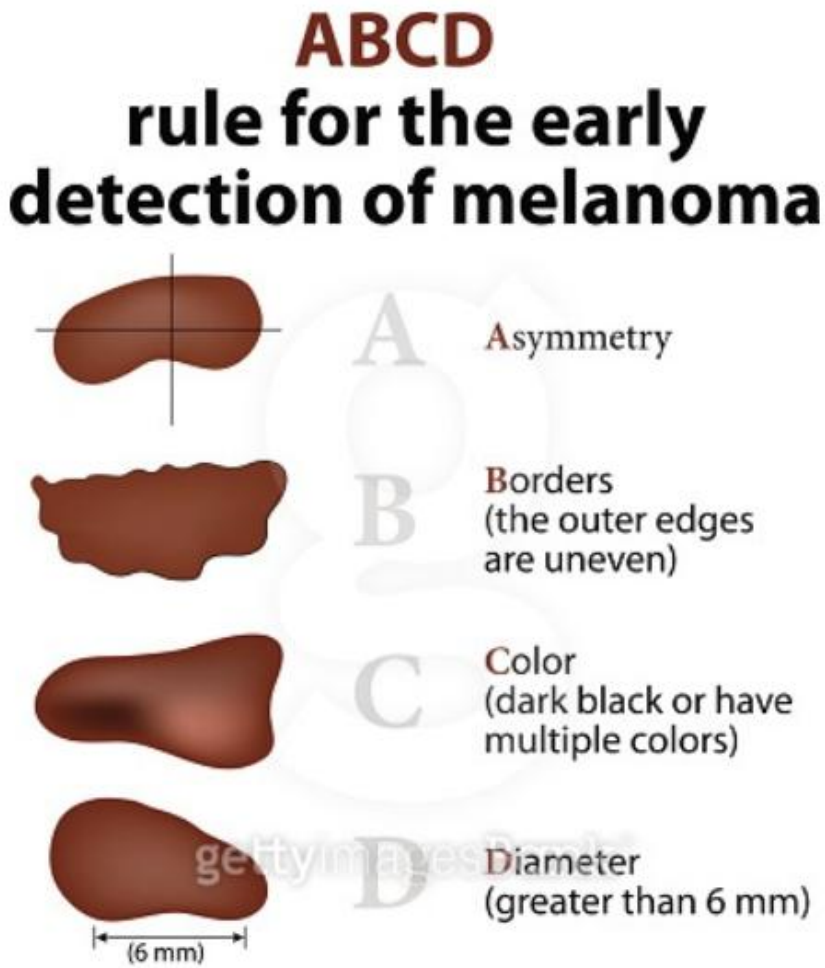
ABCD Rule Skin을 코드 구현하여 각 A, B, C, D별 값을 출력

A 비대칭성

B 경계 불규칙성

C 색상 다양성

D 크기



Tech Stack

건양대학교 KONYANG UNIVERSITY

- **개발 도구**: Visual Studio Code (Vs Code)

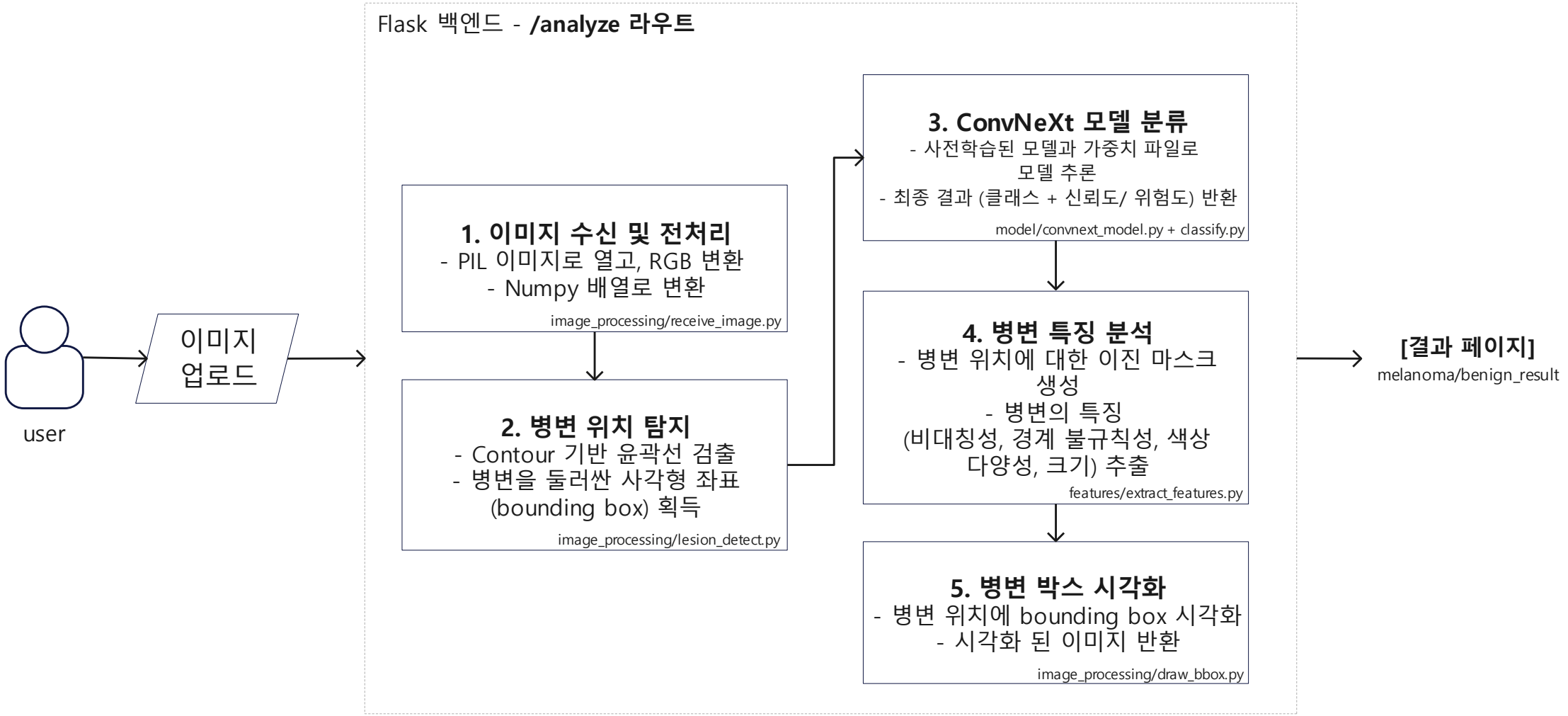
- **프론트엔드**:
  - HTML
  - CSS
  - JavaScript + Jinja2

- **백엔드**:
  - Flask (Python)
  - PyTorch + timm 라이브러리
    - ConvNeXt 기반 딥러닝 분류 모델 적용

- **웹 브라우저 기반 클라이언트 기술**:
  - Fetch API
  - localStorage / sessionStorage

- **이미지 처리**
  - OpenCV
  - PIL (pillow)

System Architecture Flow



화면 구성 미리보기

건양대학교 KONYANG UNIVERSITY

/splash 화면

앱이 시작될 때
사용자에게 보여지는
일시적인 화면

/intro 화면

흑색종이 무엇인지,
왜 중요한지를 소개

/start 화면

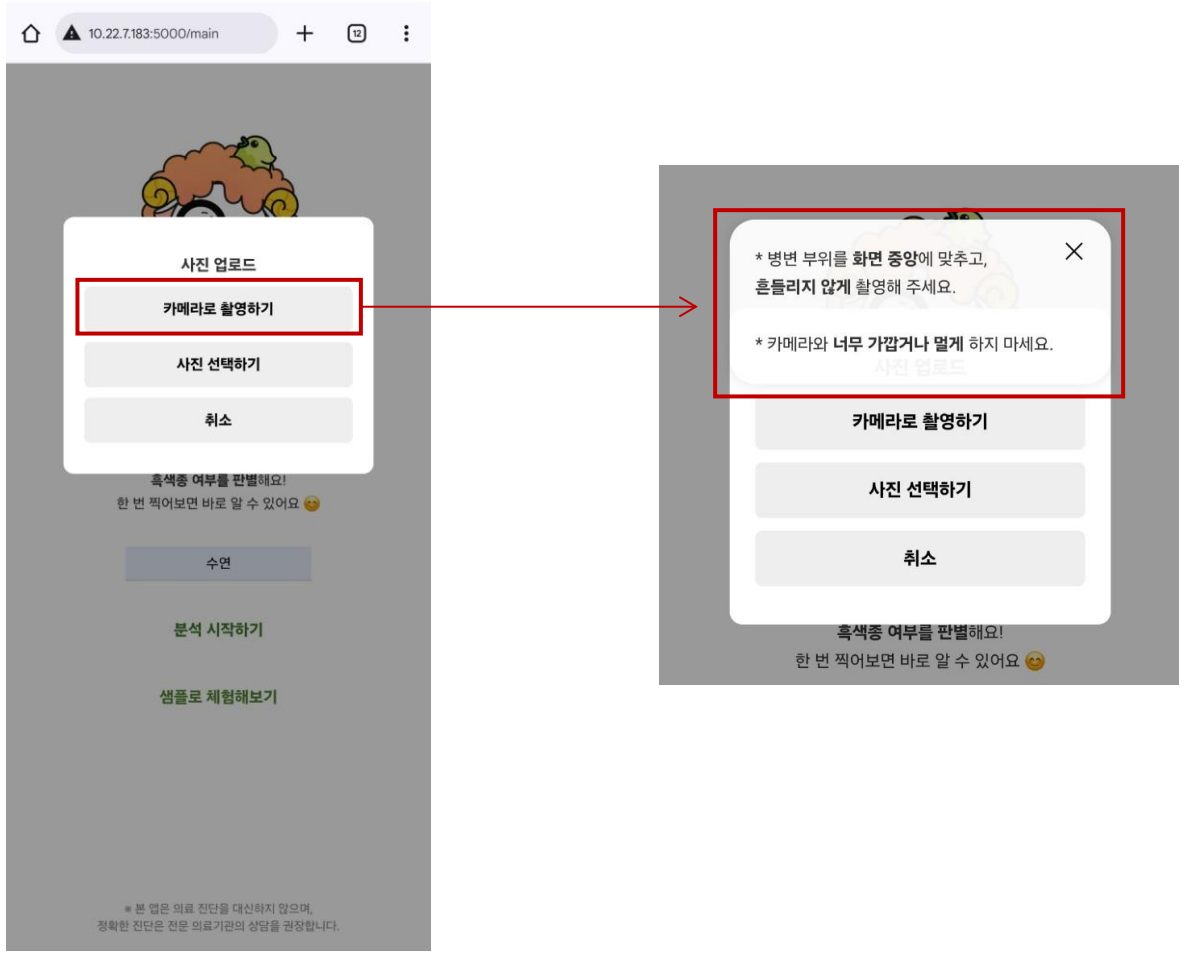
앱의 주요 기능을 요약,
분석 시작을 유도하는
첫 진입 화면

화면 구성 미리보기

/main 화면

직접 이미지를 업로드하거나
샘플 이미지로 체험할 수 있는
메인 기능 시작 화면

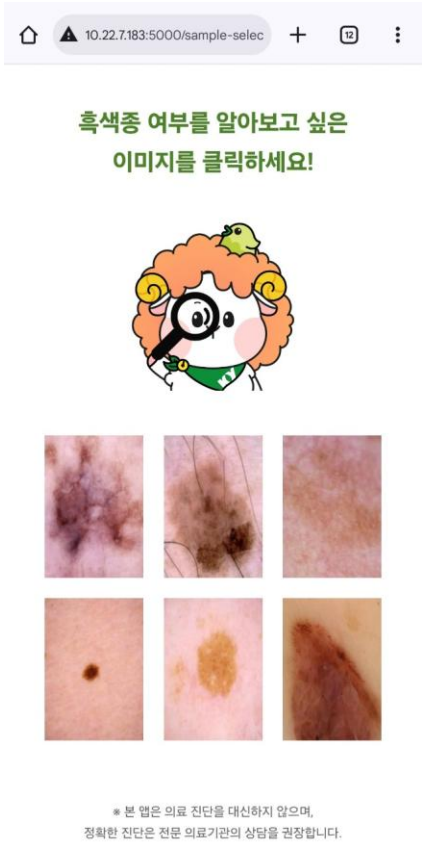
화면 구성 미리보기


10.22.7.183:5000/main
사진 업로드
카메라로 촬영하기
사진 선택하기
취소
흑색종 여부를 판별해요!
한 번 찍어보면 바로 알 수 있어요
수연
분석 시작하기
샘플로 체험해보기
* 본 앱은 의료 진단을 대신하지 않으며,
정확한 진단은 전문 의료기관의 상담을 권장합니다.
* 병변 부위를 화면 중앙에 맞추고,
흔들리지 않게 촬영해 주세요.
* 카메라와 너무 가깝거나 멀게 하지 마세요.
카메라로 촬영하기
사진 선택하기
취소
흑색종 여부를 판별해요!
한 번 찍어보면 바로 알 수 있어요

화면 구성 미리보기

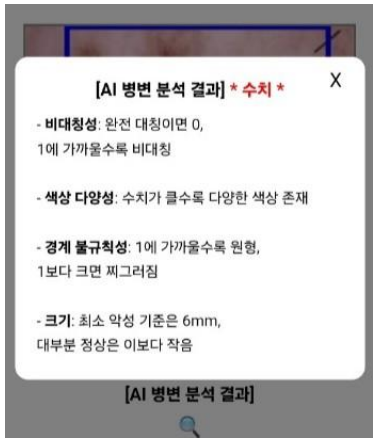
건양대학교 KONYANG UNIVERSITY

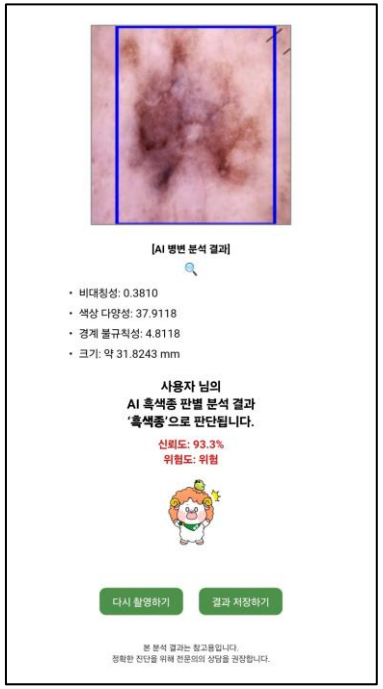
/sample-select 화면



미리 준비된 샘플 이미지를 선택하여
분석 결과를 체험할 수 있는 화면

/melanoma_result 화면





병변이 흑색종으로 판단된 경우
출력된 병변 특징 함께 위험도 안내 제공
('뀨'가 놀란 이미지)

화면 구성 미리보기

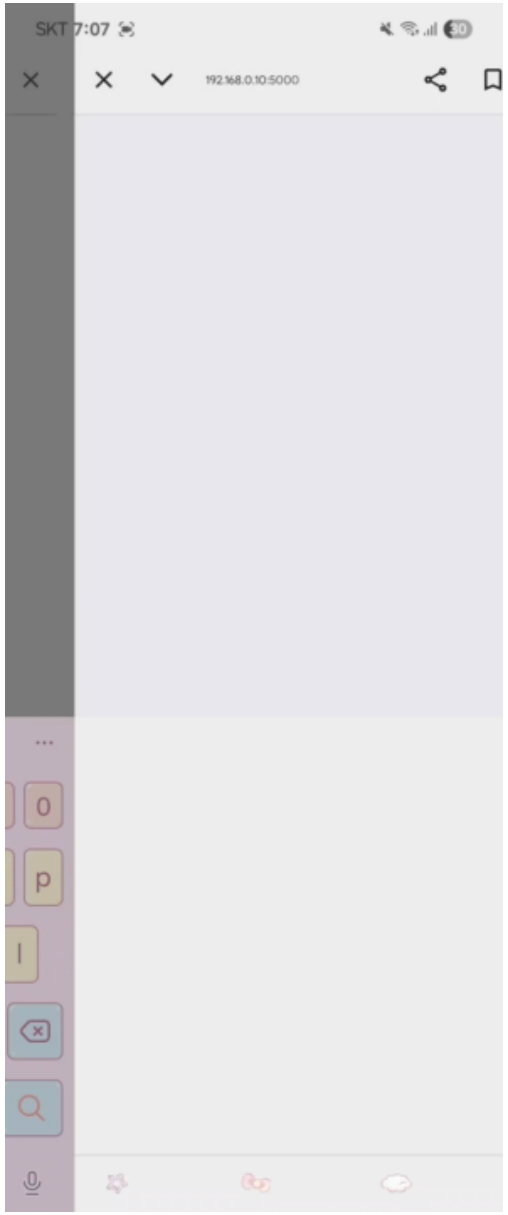
/benign_result 화면

병변이 비흑색종으로 판단된 경우
출력된 병변 특징 함께 위험도 안내는 미제공
('뀨'가 떳떳한 이미지)

System Demonstration

Future Works

System Demonstration



Future Works

- **성과:**
  - 실전 Flask + AI 통합 경험
  - 의료 AI 서비스의 가능성 실현
  - CNN 모델 기반 흑색종 분류, 특징 추출 구현

- **향후 계획**
  - 사용자 이력 저장 및 관리 기능 확대
  - Android Studio + WebView 활용하여 앱 배포
  - 앱과 병원 서버 연동으로 환자와 의료진 간 커뮤니케이션 활성화

Reference

References

[1] American Cancer Society. (2023). *Cancer Facts & Figures 2023*. American Cancer Society. (https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures.html)

[2] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115–118. (https://doi.org/10.1038/nature21056)

[3] 김용수, 조우성, 오승민, 조효은, 백용선. (2023). 합성곱 신경 회로망 모델을 활용한 흑색종 피부암 진단. 한국지능시스템학회 논문지, 33(3), 228-241. 10.5391/JKIIS.2023.33.3.228 (합성곱 신경 회로망 모델을 활용한 흑색종 피부암 진단 - 한국지능시스템학회 논문지 - 한국지능시스템학회 : 논문 - DBpia)

[4] Nodariy. (2022). *Melanoma Skin Cancer Dataset of 10000 Images* [Data set]. Kaggle. (https://www.kaggle.com/datasets/nodariy/melanoma-skin-cancer-dataset-of-10000-images)

[5] He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016. [링크]

[6] Tan and Le, "EfficientNet: Rethinking Model Scaling for CNNs," ICML, 2019. [링크]

[7] Simonyan and Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," ICLR, 2015. [링크]

[8] Huang et al., "Densely Connected Convolutional Networks," CVPR, 2017. [링크]

[9] Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," ICLR, 2021. [링크]

[10] Liu et al., "A ConvNet for the 2020s," CVPR, 2022. [링크]

+

1-1) Augmentation 조합 성능 비교

표 1. test 데이터셋에 대한 Augmentation 조합 성능 표

조합명	Accuracy	Recall	Precision	F1-score
기본	0.9567	0.8756	0.9471	0.9099
F1	0.9556	0.8400	0.9793	0.9043
F2	0.9444	0.8933	0.8855	0.8894
F3	0.9533	**0.9067**	0.9067	0.9067
F4	0.9511	0.8889	0.9132	0.9009
F5	0.9511	0.8800	0.9209	0.9000
C1	0.9179	0.8143	0.8507	0.8321
C2	**0.9644**	0.8667	0.9898	**0.9242**
C3	0.9522	0.8844	0.9213	0.9025
C4	0.9556	**0.9067**	0.9148	0.9107
C5	0.9567	0.8622	0.9604	0.9087
C6	0.9444	0.8800	0.8959	0.8879
C7	0.9600	0.8622	0.9749	0.9151
C8	0.9567	0.8311	**0.9947**	0.9056

- **Augmentation 조합별 성능 비교 결과**

- 가장 우수한 조합:

C2 조합(RandomRotation + ColorJitter)

[F1-score(0.9242) / Accuracy(0.9644)]

1-1) Augmentation 조합 성능 비교

표 2. Augmentation 조합 C2에 대한 성능 표

Dataset	Epoch	Loss	Accuracy	Recall	Precision	F1-score
train	1	0.2096	0.9240	0.8371	0.8559	0.8464
	2	0.1071	0.9617	0.8943	0.9494	0.9210
	3	0.0659	0.9788	0.9381	0.9762	0.9568
	4	0.0425	0.9855	0.9629	0.9787	0.9707
	5	**0.0308**	**0.9898**	**0.9762**	**0.9827**	**0.9795**
val	1	0.1672	0.9422	0.8400	0.9220	0.8791
	2	**0.1418**	0.9467	0.8844	0.9005	0.8924
	3	0.1594	0.9400	0.8756	0.8834	0.8795
	4	0.1597	0.9567	**0.8933**	0.9306	**0.9116**
	5	0.1726	**0.9622**	0.8711	**0.9751**	0.9202
test	-	**0.1524**	**0.9644**	**0.8667**	**0.9898**	**0.9242**

1-1) Augmentation 조합 성능 비교

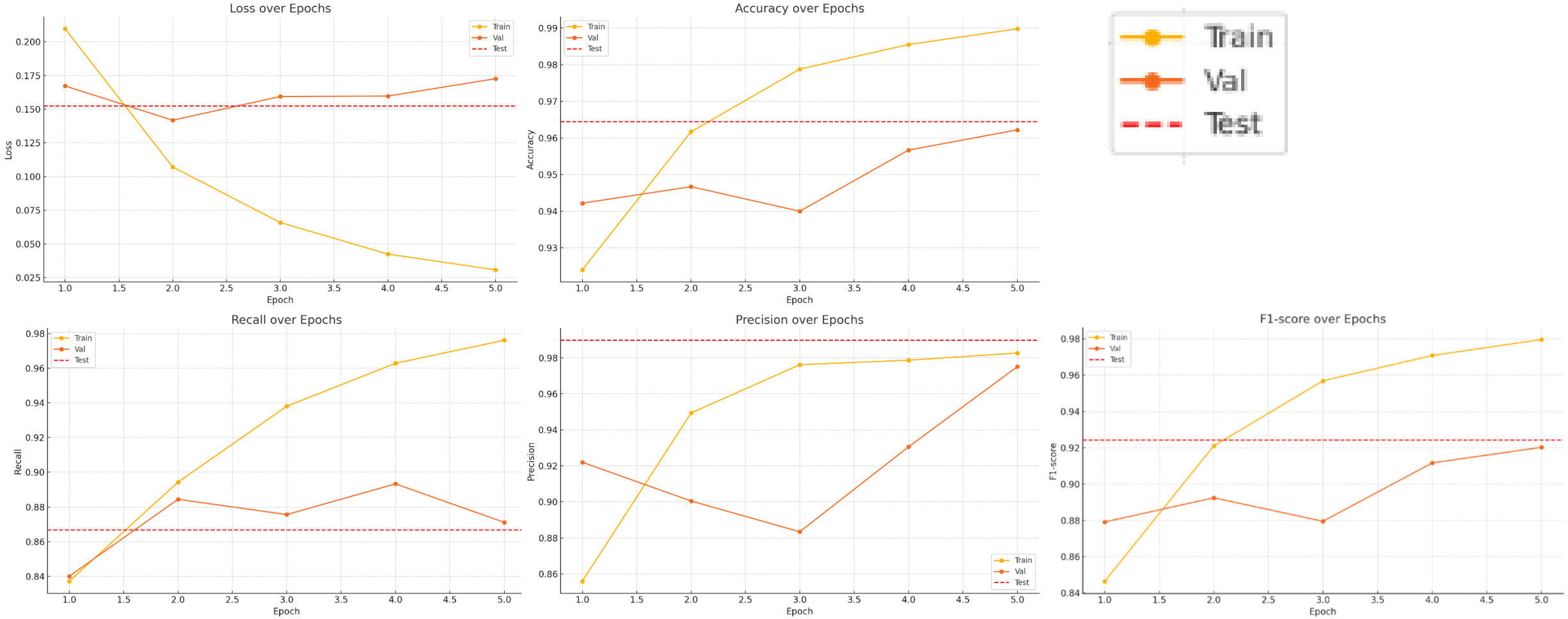


그림 1. Augmentation 조합 C2에 대한 성능 시각화

1-1) Augmentation 조합 성능 비교

(1) Loss 및 주요 지표 추이

Train Loss: 꾸준히 감소(**0.2096→0.0308**)

Val Loss: 2 epoch 최저 후 반등

→ **일반화 성능 저하 우려**

Accuracy, F1-score:

Train에서 지속 상승, Val은 정체 후 회복

→ **전반적 수렴 양상**

Test 지표는 Val 최고값과 유사

→ **성능 안정성 확보**

(2) 과적합 판단

2 epoch 이후

Train-Val 간 Loss 격차 확대 및 Recall·F1 간 간극 발생

→ **과적합 진행 중**

(3) Best Epoch 기준

Val Loss 기준: 2 epoch

Val F1 기준: 5 epoch

Test 근접성 기준: 4~5 epoch

→ **4 epoch 최적**

1-1) Augmentation 조합 성능 비교

(4) Early Stopping 평가 (patience=3)

Val Loss가 2 epoch 이후 3**회 연속 개선되지 않음**(3~5 epoch)

→ **5 epoch 종료 후 Early Stopping 작동**

(5) 학습 안정성 및 수렴 특성

Train은 모든 지표에서 **일관된 수렴**

Val은 중간 하락 후 회복

→ **과적합 후 안정화 흐름**

1-2) Augmentation 조합 + Weighted Sampling의 성능

표 3. test 데이터셋에 대해 Weighted Sampling 적용한 성능 표

조합	Accuracy	Recall	Precision	F1-score
기본	0.9578	0.8667	0.9606	0.9112
F1	0.9511	0.8356	0.9641	0.8952
F2	0.9533	0.9200	0.8961	0.9079
F3	0.9611	0.9111	0.9318	0.9213
F4	0.9567	0.9200	0.9079	0.9139
F5	0.9544	0.8533	0.9600	0.9035
C1	0.9422	0.9156	0.8619	0.8879
C2	0.9611	0.8756	0.9657	0.9184
C3	0.9611	0.8800	0.9612	0.9188
C4	**0.9656**	0.8933	**0.9663**	**0.9284**
C5	0.9300	**0.9333**	0.8140	0.8696
C6	0.8898	0.8743	0.9015	0.8877
C7	0.9533	0.8756	0.9336	0.9037
C8	0.9589	0.9111	0.9234	0.9172

- **Weighted Sampling 적용한 결과**

- 가장 우수한 조합:

C4 조합(Flip + Rotation + Jitter)

[F1-score 0.9284, Accuracy 0.9656, Precision 0.9663]

1-2) Augmentation 조합 + Weighted Sampling의 성능

표 4. Weighted Sampling 적용한 Augmentation 조합 C4에 대한 성능 표

Dataset	Epoch	Loss	Accuracy	Recall	Precision	F1-score
train	1	0. 2396	0. 8993	0. 8834	0. 9185	0. 9006
	2	0. 1536	0. 9424	0. 9290	0. 9545	0. 9415
	3	0. 1104	0. 9550	0. 9500	0. 9579	0. 9539
	4	0. 0821	0. 9676	0. 9648	0. 9703	0. 9675
	5	0. 0754	0. 9693	0. 9690	0. 9695	0. 9692
	6	0. 0571	0. 9788	0. 9739	0. 9837	0. 9788
	7	0. 0383	0. 9864	0. 9875	0. 9861	0. 9868
	8	**0. 0259**	**0. 9924**	**0. 9928**	**0. 9918**	**0. 9923**
val	1	0. 1900	0. 9311	0. 8889	0. 8439	0. 8658
	2	0. 1919	0. 9300	0. 9156	0. 8240	0. 8674
	3	0. 1451	0. 9544	0. 8667	**0. 9466**	0. 9049
	4	0. 1496	0. 9389	0. 9156	0. 8512	0. 8822
	5	**0. 1444**	**0. 9578**	0. 8756	0. 9517	**0. 9120**
	6	0. 1793	0. 9478	**0. 9378**	0. 8648	0. 8998
	7	0. 1856	0. 9522	0. 8667	0. 9375	0. 9007
	8	0. 1913	0. 9511	0. 8978	0. 9058	0. 9018
test	-	**0. 1554**	**0. 9656**	**0. 8933**	**0. 9663**	**0. 9284**

1-2) Augmentation 조합 + Weighted Sampling의 성능

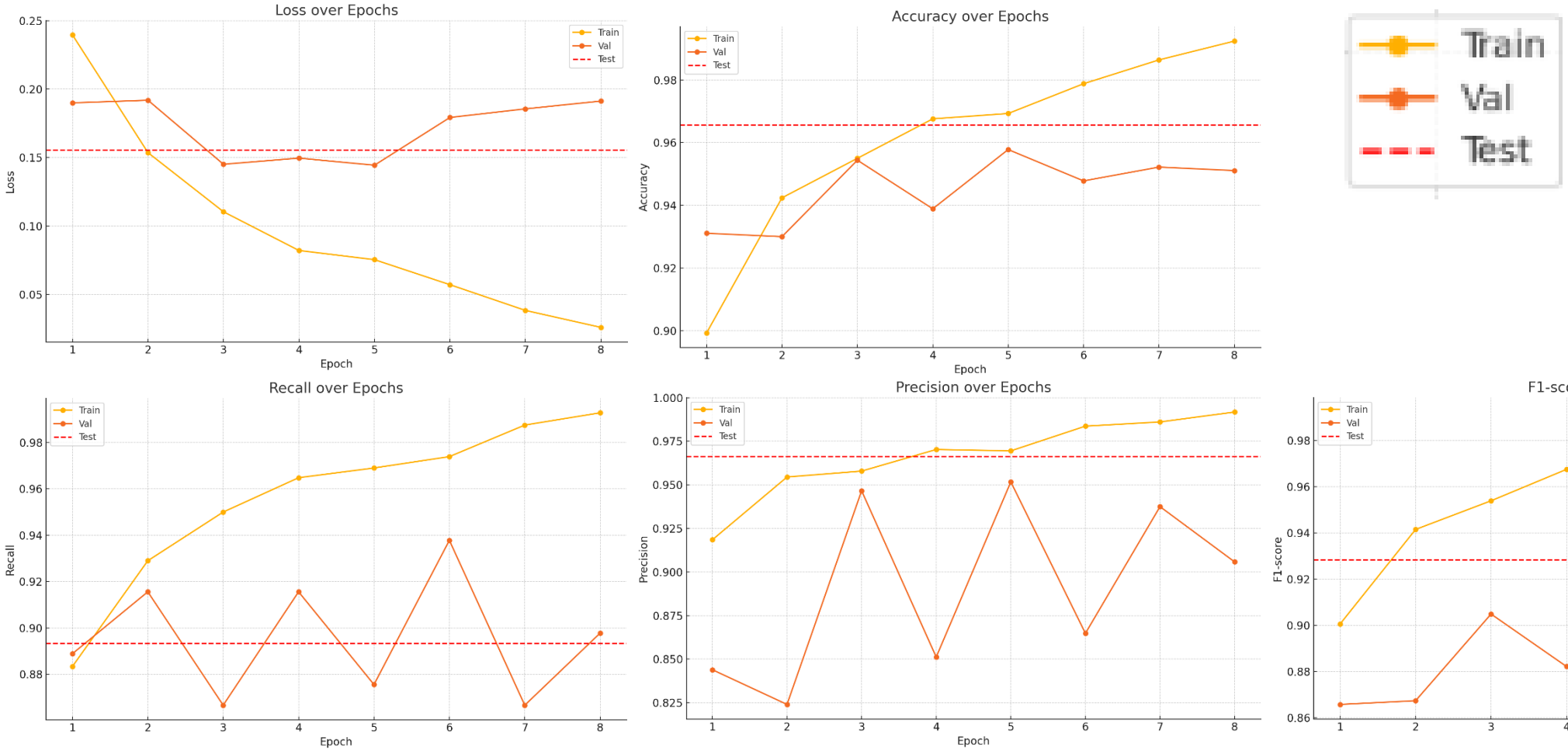


그림 2. Weighted Sampling 적용한 Augmentation 조합 C4에 대한 성능 시각화

1-2) Augmentation 조합 + Weighted Sampling의 성능

(1) Loss 및 주요 지표 추이

Train Loss: 꾸준히 감소(**0.2396 → 0.0259**)

Val Loss: 3~5 epoch 하락(**최저 0.1444**) 후 **다시 증가**

Train Accuracy/F1: **급상승 및 안정적 수렴**

Val F1: **5 epoch**에서 **최고(0.9120)** 후 정체

Test 지표: 대부분 Val 최고값과 유사하거나 근접

→ **양호한 일반화 성능 확보**

(2) 과적합 판단

Train-Val 간 Loss 및 Precision·F1-score 격차:

6 epoch 이후 확대

→ **과적합 발생 시점 6 epoch 전후로 판단**

(3) Best Epoch 기준

Val Loss 기준: 5 epoch (0.1444)

Val F1 기준: 5 epoch (0.9120)

Test 근접성 기준: 5 epoch

→ **5 epoch 최적**

1-2) Augmentation 조합 + Weighted Sampling의 성능

(4) Early Stopping 평가 (patience=3)

Val Loss가 5 epoch 이후 **3회 연속 상승(6~8 epoch) → Early Stopping은 8 epoch에서 작동**

(5) 학습 안정성 및 수렴 특성

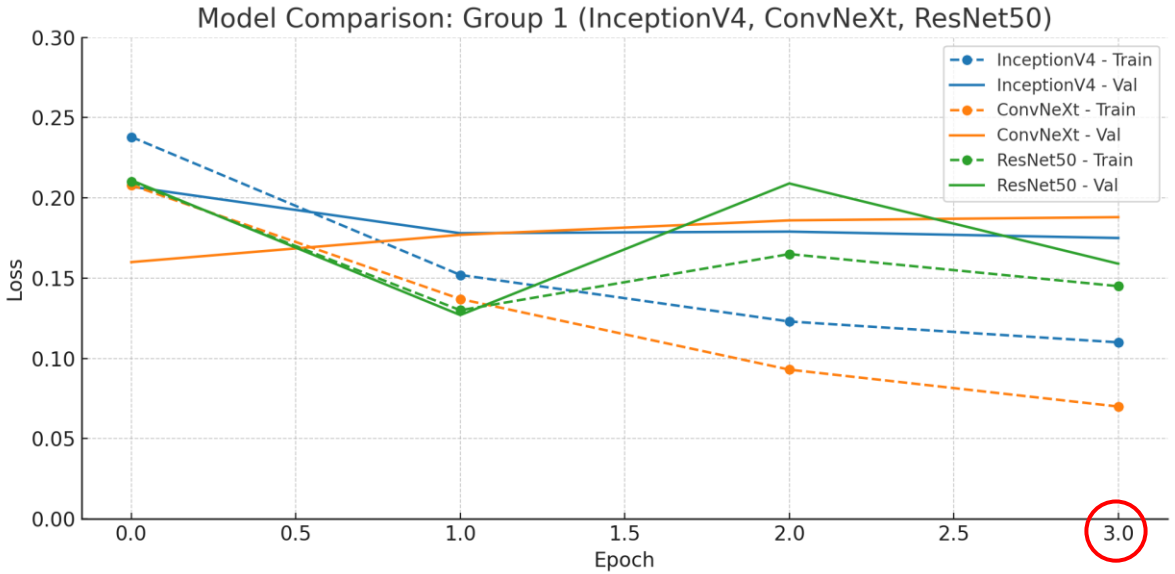
Train 지표는 명확한 수렴

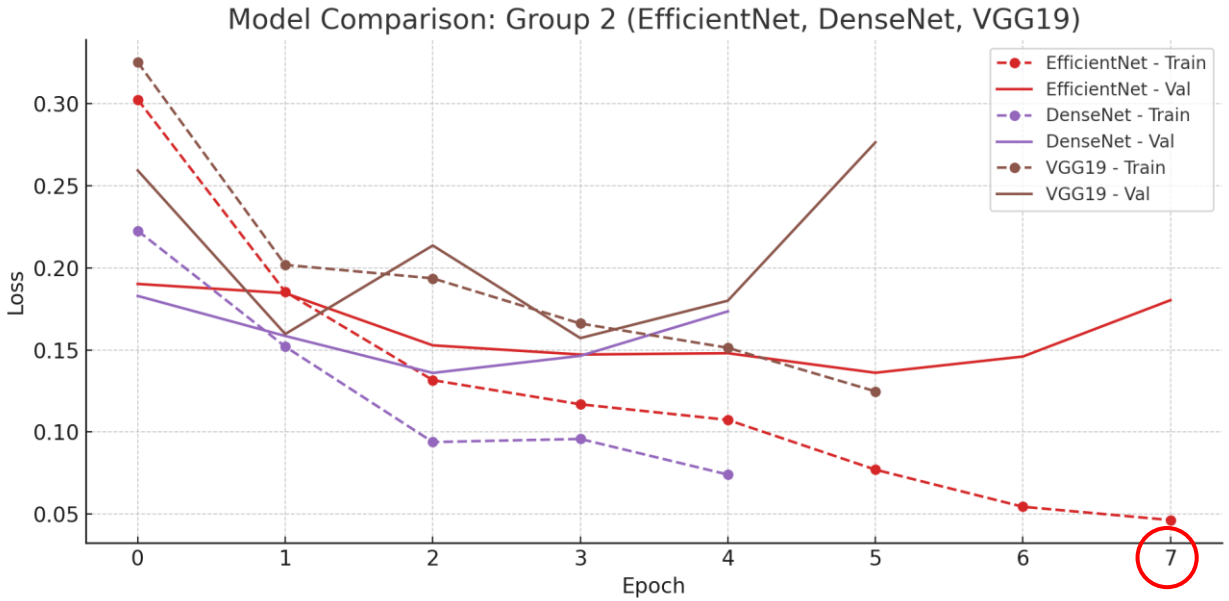
Val 지표는 5 epoch 이후 **변동성 증가**

전반적으로 학습 후반 과적합 발생

Val 성능은 **Test 기준 허용 가능한 범위 내 유지**

2) 각 모델별 Train/Val Loss





→ ConvNeXt가 가장 안정적이고 일반화 성능이 우수.

→ InceptionV4는 무난.

→ ResNet50은 과적합 가능성이 가장 큼.

→ DenseNet이 가장 균형 잡힌 학습, 과적합 적음.

→ EfficientNet과 VGG19는 후반부에서 과적합이 가능성이 큼.

2-1) 각 모델별 Test Loss

→DenseNet121 가장 우수한 일반화 성능. 안정적인 학습 가능성 높음

→ConvNeXt DenseNet121에 근접. 효율성과 성능의 균형이 좋음

→EfficientNetB0 ConvNeXt와 비슷한 성능. 경량 구조 대비 우수함

2-2) ConvNeXt의 특성과 흑색종 진단 성능 향상의 관계

1. Transformer에서 영감을 받은 구조
2. Layer Normalization + GELU + Inverted Bottleneck
3. 모던 CNN 최적화 전략 적용
4. 높은 Recall과 F1-score

모델	상대적 한계
ResNet50	오래된 구조, 작은 receptive field
EfficientNet	성능은 좋지만 모바일 친화적으로 설계되어 복잡한 병변 구조표현에 제한적
DenseNet	모든 레이어 간 연결로 인해 메모리와 계산량 증가, 일반화 성능은 평균
VGG	파라미터가 너무 많고 깊이에 비해 표현력 한계, 과적합 위험
InceptionV3	다양한 scale의 filter 사용은 장점이지만 최근 모델에 비해 표현력이 낮음

3) Flask 디렉토리 구조:

```
melanoma_webapp/
│
├── app.py                          # Flask 웹 메인 실행 파일
│
├── static/                         # 정적 파일 (이미지, 폰트, 샘플 이미지)
│   ├── font/
│   ├── images/                     # 마스코트 및 기타 시각 요소
│   └── samples/                    # 샘플 병변 이미지 (체험용)
│
├── templates/
│   ├── splash.html
│   ├── intro.html
│   ├── start.html
│   ├── main.html
│   ├── sample_select.html
│   ├── melanoma_result.html
│   └── benign_result.html
│
├── model/          # AI 관련 모듈
│   ├── convnext_model.py           # ConvNeXt 모델 구조 및 로딩 함수
│   └── classify.py                 # 이미지 분류 및 확률 반환
│
├── image_processing/               # 이미지 전처리 관련 모듈
│   ├── receive_image.py            # 업로드 이미지 수신 및 PIL 변환
│   ├── lesion_detect.py            # 병변 위치 탐지 (바운딩 박스)
│   └── draw_bbox.py
│
├── features/                       # 병변 특징 추출 관련 모듈
│   ├── extract_features.py         # 수치 추출
│   └── text_generator.py           # 결과 요약 텍스트 생성
│
├── utils/                          # 유틸리티 함수들 모음
│   ├── io_utils.py                 # 이미지 base64 인코딩 등 입출력 보조
│   └── response_builder.py         # 최종 결과 요약 포맷 생성
│
└── best_model.pth                  # 학습 완료된 ConvNeXt 모델 가중치 파일
```