

EfficientNet 기반 오렌지 질병 분류 모델 경량화 실험 레포트

7.8M으로 208M 따라잡기

카카오테크부트캠프

인공지능과정

Tei.eom(엄태성)

2025년 3월 24일

목차

1. 서론
2. 데이터셋 설명
3. 모델 설명
4. 실험 방법
5. 결과 및 분석
6. 결론
7. 참고문헌
8. 부록

1. 서론

오렌지와 같은 과일의 질병을 조기에 정확하게 진단하는 것은 농업 분야에서 생산량과 품질을 유지하는 데 필수적이다. 최근 딥러닝 기반의 이미지 분류 기술이 이러한 과제를 자동화하는 데 활용되고 있으며, 특히 대용량의 사전학습된 합성곱 신경망(CNN)이 뛰어난 성능을 보인다. Kaggle의 Orange Diseases Dataset은 EfficientNetV2-XL과 같은 초대형 모델(약 2억 800만 개의 파라미터)을 활용할 경우 약 98% 이상의 높은 분류 정확도를 달성할 수 있음이 보고되었다. 그러나 이렇게 큰 모델은 연산 자원과 추론 시간이 많이 소요되어 실제 현장 적용이나 임베디드 시스템 배포에는 비효율적이다. 본 실험의 목적은 상대적으로 경량인 모델(EfficientNet-B0 Wide-SE, 약 780만 개 파라미터)을 사용하여 단계적 개선 기법들을 적용함으로써, 대형 모델에 필적하는 분류 성능을 달성하는 것이다.

이를 위해 본 보고서에서는 작은 모델을 다양한 방법으로 향상시키는 일련의 실험을 설계하였다. 먼저 사전 학습된 EfficientNet-B0 Wide-SE 모델을 전체 파라미터에 대해 파인튜닝하여 기준 성능(baseline)을 확보한다. 이후 모델 구조에 Convolutional Block Attention Module(CBAM)과 같은 attention 메커니즘을 추가하여 특징 표현력을 높이고, 학습률 스케줄러로 OneCycleLR을 도입하여 최적화 과정을 개선한다. 또한, 데이터 증강 기법으로 Cutout과 Mixup을 적용하여 모델의 일반화 성능 향상을 시도하고, 마지막으로 특징 추출부를 고정하고 출력층만 재학습(Partial fine-tuning)하는 실험을 수행하여 전체 파인튜닝과의 성능 차이를 비교한다. 이러한 단계별 접근을 통해 경량 모델의 성능 변화를 면밀히 분석하고, 어느 기법이 성능 향상에 기여하는지 규명하고자 한다.

본 보고서의 구성은 다음과 같다. 2장에서는 실험에 사용된 데이터셋의 구성과 특성을 설명한다. 3장에서는 비교 대상인 대형 모델과 우리가 사용한 경량 모델의 구조 및 관련 기법(CBAM 등)을 소개한다. 4장에서는 실험 방법과 설정에 대해 단계별로 기술한다. 5장에서는 각 실험의 결과를 정리한 표와 함께 정확도의 향상 추이를 분석하고 그 원인을 논의한다. 6장에서는 결론으로서 주요 결과를 요약하고 향후 연구 방향을 제언한다. 마지막으로 7장과 8장에는 참고문헌과 부록을 포함한다.

2. 데이터셋 설명

실험에 사용된 데이터는 Kaggle에서 제공하는 **Orange Diseases Dataset**이다. 이 데이터셋은 총 4개의 클래스(오렌지의 상태)에 대한 이미지들로 구성되어 있다. **Blackspot**(검은 점병), **Canker**(꺾양병), **Fresh**(신선한 정상 과일), **Greening**(그린닝병, 일명 황룡병)이다. 총 이미지 수는 약 1,090장으로 비교적 적은 편이며, 각 이미지에는 한 개의 오렌지가 포함되어 해당 질병 여부를 보여준다. 데이터셋 제공자가 이미 데이터 증강을 적용하여 이미지 수를 늘려 놓았는데, 회전, 확대/축소, 밝기 변화 등 다양한 변형을 가한 이미지들이 포함되어 있어 모델 학습시 어느 정도의 일반화 효과를 기대할 수 있다. 클래스 별 이미지 분포는 완전히 균형잡혀 있지는 않으나, 큰 편차 없이 구성되어 있다 (**Fresh** 이미지가 다소 많이 포함된 경향이 있다). 이러한 증강 포함 데이터셋은 모델이 훈련 시 처음부터 다양한 변형 사례를 접하게 되어 과적합(overfitting)을 완화하는 데 도움을 준다.

본 실험에서는 별도로 훈련(train)과 검증(validation) 세트를 분리하여 사용하였다. 전체 1,090장의 이미지 중 약 80%를 훈련용으로, 20%를 검증용으로 무작위 분할하였다 (class 비율이 유지되도록 층화 추출). 모델의 성능 평가는 주로 검증 세트에 대한 정확도(accuracy)로 측정하였다. 추가로, 각 클래스에 대한 모델의 혼동 양상을 파악하기 위해 최종 모델에 대해서는 혼동 행렬(confusion matrix)을 확인하였다. 데이터 전처리 측면에서는, 모든 이미지를 모델 입력 크기에 맞게 일괄적으로 리사이즈하고 정규화하였다. EfficientNet 계열 모델의 사전 학습 가중치가 ImageNet으로 학습되었으므로, 해당 사전 학습 시의 정규화 방식(mean, std)을 따라 이미지를 정규화하여 입력으로 사용하였다.

3. 모델 설명

3.1 비교 대상: EfficientNetV2-XL (대형 모델)

우선 성능 비교의 기준점이 되는 모델은 **EfficientNetV2-XL**이다. EfficientNetV2는 2021년에 제안된 최신 EfficientNet 계열 모델로, 모델 크기 대비 높은 정확도와 빠른 학습을 목표로 설계되었다. 그 중 XL 버전은 EfficientNetV2 계열 중 가장 큰 모델로, 약 2억 800만 개에 달하는 파라미터를 가지고 있다. 이 모델은 ImageNet-21k 데이터셋(약 1,400만 장의 이미지와 21천 개의 클래스)으로 사전 학습(pre-training)되어 풍부한 이미지 표현 능력을 보유하고 있다. Kaggle Notebooks에 따르면, EfficientNetV2-XL를 이 오렌지 질병 데이터셋에 미세조정(fine-tuning)하여 약 **98.24%**의 정확도를 달성한 바 있다. 이는 본 실험에서 경량 모델이 도달해야 할 목표 성능으로 삼았다. 다만, EfficientNetV2-XL과 같은 대용량 모델은 메모리와 연산량이 매우 많아 실제 배포 시 부담이 되므로, 작은 모델로 유사한 성능을 내는 것이 중요하다.

3.2 사용 모델: EfficientNet-B0 Wide-SE (경량 모델)

경량화된 대안 모델로서 **EfficientNet-B0 Wide-SE**를 선정하였다. EfficientNet-B0는 EfficientNet 계열의 가장 작은 기본 모델로, 원본 EfficientNet-B0의 파라미터 수는 약 5백만 개 정도이지만, 여기서는 **Wide-SE** 변형을 적용하여 약 **780만 개**의 파라미터를 가진 모델을 사용하였다. "Wide-SE"란 EfficientNet-B0의 네트워크 폭(width)을 약간 늘리고, 각 블록에 포함된 Squeeze-and-Excitation(SE) 모듈을 활용한 구조를 의미한다.

(EfficientNet 자체가 SE 블록을 포함하지만, Wide-SE 변형은 채널 수를 증가시켜 약간 더 많은 파라미터와 성능을 얻는 커스텀 모델이다.) 이 모델 역시 ImageNet-1k 데이터셋(일반적인 1000개 클래스의 ImageNet)으로 사전 학습된 가중치를 초기화로 사용하였다.

EfficientNet-B0 기반 모델은 경량임에도 불구하고, 이미지 특징 추출에 효율적인 Conv 블록 설계와 SE를 통한 채널 주의 메커니즘으로 준수한 성능을 보이는 것으로 알려져 있다.

EfficientNet-B0 Wide-SE 모델은 본 문제의 4-class 분류를 위해 출력층에 4개의 뉴런을 갖는 완전연결층(dense layer)을 적용하여 사용하였다. 파인튜닝 과정에서 이 출력층 가중치는 무작위 초기화된 후 학습되고, 나머지 EfficientNet 기반층의 가중치는 사전학습 가중치에서 시작하여 학습된다 (baseline 실험의 경우 모든 계층을 학습). 작은 모델이지만 사전학습

을 활용함으로써, 데이터셋 규모가 크지 않아도 효과적으로 특성을 학습할 수 있을 것으로 기대된다.

3.3 CBAM (Convolutional Block Attention Module)

모델 성능 향상을 위해 EfficientNet-B0 Wide-SE 구조에 **CBAM** 모듈을 통합하였다. CBAM은 ‘BAM: Bottleneck Attention Module’ 논문에서 제안된 모듈로, CNN의 특징맵에 대한 **채널**과 **공간** 두 가지 차원의 attention 맵을 순차적으로 계산하여 원본 특징맵에 곱함으로써, 중요한 정보는 강조하고 불필요한 정보는 억제하도록 한다. 구체적으로, 주어진 중간 특징맵에 대해 전역 평균풀링과 맥스풀링을 통해 채널 중요도를 계산한 뒤(channel attention), 이어서 채널 축을 따라 평균 및 최대 풀링한 2차원 맵으로부터 공간 중요도를 계산하여(spatial attention) 순차적으로 적용한다. EfficientNet의 SE 모듈이 채널 차원의 attention만 수행하는 것에 반해, CBAM은 추가로 공간적 위치의 중요도까지 반영하므로, 이미지 내에서 질병 반점이나 병변이 있는 국소 부위에 모델이 더욱 집중할 수 있게 해준다. 본 실험에서는 EfficientNet-B0의 일부 블록 뒤에 CBAM을 삽입하는 형태로 구조를 수정하였으며, 추가되는 파라미터는 미미하지만 feature 표현력 향상을 기대할 수 있다.

3.4 OneCycleLR 스케줄러

딥러닝 모델의 학습 최적화 단계에서 **학습률 스케줄링**은 매우 중요한 요소이다. 본 실험의 일부 단계에서는 **OneCycleLR** 스케줄러를 적용하였다. OneCycleLR은 Leslie N. Smith가 제안한 기법으로, 한 번의 학습 사이클 동안 학습률을 처음에 증가시켰다가 나중에 점진적으로 감소시키는 전략이다. 일반적으로 사이클의 중반까지 학습률을 선형 혹은 cosine 형태로 최대치까지 높여 모델이 다양한 weight 공간을 탐색하도록 한 후, 이후에는 학습률을 낮추며 미세한 최적화를 진행한다. 이렇게 하면 훈련 초기에 비교적 큰 학습률로 관성(momentum)을 얻어 빠르게 수렴하면서도, 마지막에는 작은 학습률로 정확한 수렴을 유도하여 더 나은 일반화 성능(Generalization)을 얻을 수 있다고 알려져 있다. OneCycleLR은 훈련 epoch 수에 맞춰 하나의 사이클을 적용하며, 보통 사이클 종료 시 학습률을 매우 낮은 값까지 떨어뜨려 가중치가 안정되도록 한다. 본 실험에서는 baseline 대비 OneCycleLR을 도입한 실험에서, 동일한 epoch 동안 이러한 학습률 변화를 주어 모델이 국소 최적해에 빠지지 않고 더 평탄한 최적해로 유도되도록 하였다.

3.5 데이터 증강 기법: Cutout 및 Mixup

추가적인 일반화 성능 향상을 위해 **Cutout**과 **Mixup**이라는 데이터 증강(data augmentation) 기법들을 실험적으로 적용해보았다. Cutout은 훈련 시 입력 이미지의 랜덤 위치에 정사각형의 작은 영역을 검정색(또는 평균값)으로 가리는 방법으로, 모델이 이미지의 일부 정보가 없더라도 나머지 단서로 정답을 맞히도록 학습시킨다. 이는 일종의 지역적 dropout처럼 작용하여 특정 부분에 과도하게 의존하는 것을 막고 보다 견고한 특징 표현을 얻게 하는 효과가 있다. Mixup은 두 개의 이미지를 선형 결합하여 새로운 가상 훈련 샘플을 만드는 증강 기법으로, 이미지 A와 B를 랜덤 비율로 섞은 이미지 $= \lambda * A + (1 - \lambda) * B$ 를 만들고 라벨도 동일한 비율로 섞어서 학습시킨다. 이렇게 하면 모델이 각 클래스의 경계를 더욱 부드럽게 학습하고 다양한 혼합 사례를 경험함으로써, 노이즈에 강인한 결정 경계를 형성하도록 돕는다. 특히 데이터셋 크기가 작거나 클래스 간 경계가 뚜렷하지 않을 때 Mixup이 도움이 될 수 있다. 다만, Mixup으로 생성된 합성 이미지가 실제로는 존재하지 않는 중간 상태의 과일일 수 있기 때문에, 사람이 보기에는 부자연스러울지라도 모델에는 추가적인 정규화 효과를 준다.

Cutout과 Mixup 모두 비교적 간단히 구현 가능하며 이미지 분류 대회 등에서 성능을 높이기 위해 자주 사용되는 기법이다. 본 데이터셋은 이미 여러 형태의 증강 이미지가 포함되어 있지만, 우리는 실험적으로 이 두 가지 기법을 추가 적용하여 얻을 수 있는 성능 변화를 관찰하고자 하였다.

4. 실험 방법

본 섹션에서는 위에서 소개한 모델과 기법들을 조합하여 수행한 일련의 실험 과정과 설정을 단계별로 설명한다. 각 단계별 실험에서 변경되는 요소를 제외하고는, 가능한 동일한 조건을 유지하여 공정한 비교가 가능하도록 하였다.

1) Baseline: EfficientNet-B0 Wide-SE 전체 파인튜닝 – 첫 번째 실험에서는 경량 모델인 EfficientNet-B0 Wide-SE를 대상으로 **전체 계층을 미세조정(fine-tuning)**하여 baseline 성능을 측정하였다. 사전 학습 가중치를 사용하되, 모든 convolution 블록과 SE 계층까지 포함한 전체 네트워크 파라미터를 학습 가능 상태로 두었다. 최종 출력층은 4개 클래스에 맞게 새로 초기화하여 학습하였다. 이때 학습률은 비교적 보수적으로 설정하여 (예: 0.001), Adam 최적화 알고리즘을 사용하였고 학습률 감소는 epoch가 진행됨에 따라 Cosine Annealing 방식으로 약간씩 줄이는 기본 스케줄을 적용하였다. Epoch 수는 30 epoch로 설정하고, 매 epoch마다 훈련 데이터와 검증 데이터에 대한 정확도를 기록하였다. Baseline 설정에서는 특별한 주의 모듈이나 특수 스케줄러를 사용하지 않고, 기본적인 데이터 전처리와 flip/shift 등의 경미한 증강만으로 결과를 얻었다.

2) CBAM 모듈 적용 (EfficientNet-B0 + CBAM) – 두 번째 실험에서는 모델 구조를 개선하기 위해 EfficientNet-B0 Wide-SE의 일부 convolution 블록 뒤에 **CBAM 모듈**을 삽입하였다. 구체적으로, stage별로 중간 피쳐맵에 channel/spatial attention을 주기 위해 2~3개의 CBAM 블록을 기존 모델에 추가하였다. CBAM이 추가됨에 따라 파라미터 수는 소폭 증가하지만, 1% 내외의 증분으로 경량성에는 큰 영향이 없었다. 이 CBAM을 포함한 모델을 동일한 데이터로 **전체 파인튜닝**하였다. 초기 가중치는 이전 baseline 실험과 마찬가지로 ImageNet 사전학습 값 및 무작위 초기화를 사용하였다. 최적화 설정(optimizer, 학습률 등)은 baseline과 동일하게 두어, 구조 변화에 따른 순수한 성능 향상 효과를 관찰하였다. 역시 30 epoch 동안 학습을 진행하고, epoch마다 정확도를 기록하였다.

3) OneCycleLR 스케줄러 적용 – 세 번째 실험에서는 두 번째 실험(모델에 CBAM 포함)의 학습 과정에 **OneCycleLR 학습률 스케줄러**를 도입하였다. PyTorch의 OneCycleLR 스케줄러를 활용하여, 전체 30 epoch에 걸쳐 학습률이 하나의 사이클을 형성하도록 설정하였다. 사이클의 최대 학습률은 0.001으로 설정하고, 초기에는 그 10% 수준으로 시작하여 중반까지 선형 증가 후 후반부에 급격히 감소하도록 하였다. 다른 면에서 모델 구조(CBAM 포함)나 optimizer(Adam 유지)는 이전과 동일하게 유지하였다. OneCycleLR을 적용함에 따라 epoch 진행에 따른 학습률 변화도 함께 기록하여, 학습 곡선과 함께 해석하였다.

4) Cutout 및 Mixup 증강 적용 – 네 번째 실험에서는 데이터 증강을 더욱 강화하기 위해, 세 번째 실험(EfficientNet-B0 + CBAM + OneCycleLR)에 **Cutout과 Mixup**을 추가로 적용하였다. 구체적으로, 매 배치(batch)마다 각 이미지에 대해 확률적으로 일정 크기의 사각형을 가리는 Cutout을 적용하고, 또한, 배치 내 랜덤한 이미지 쌍을 선택하여 $\lambda=0.5$ 내외의 비율로 혼합한 Mixup 이미지를 생성하여 함께 학습시켰다. 두 가지 증강은 동시에 적용되었으며, 이를 통해 모델이 보다 다양한 형태의 입력을 견디고도 올바른 분류를 할 수 있도록 유도하였다. 학습률 스케줄러는 계속 OneCycleLR를 사용하고, 모델 구조 역시 CBAM 포함 상태를 유지하였다. 실험 설정은 epoch 30, 동일 optimizer 등으로 맞추어, 증강 기법 추가의 효과만을 관찰하였다. 증강으로 인해 훈련 데이터의 변동성이 커지므로, 학습 곡선의 흔들림이 관찰될 수 있어 이를 주의 깊게 추적하였다.

5) 출력층만 미세조정 (Feature Extractor 모드) – 마지막으로 다섯 번째 실험에서는 반대로 모델의 대부분을 동결(freeze)하고 출력층만 학습시키는 방식을 적용하였다.

EfficientNet-B0 Wide-SE의 사전학습 가중치를 그대로 사용하면서, 최종 분류층(4-way classifier)만을 훈련시키는 설정이다. 이는 전형적인 전이학습에서 특성 추출기(feature extractor)로 CNN을 활용하는 방식으로, 파라미터 수가 크게 줄어들고 과적합 위험을 낮출 수 있다. 이 실험에서는 baseline과 동일한 기본 모델(EfficientNet-B0 Wide-SE, CBAM 미포함)을 사용하되, 오직 출력층의 가중치만 학습하였고 나머지 계층들은 학습률 0으로 고정

하였다. 학습률은 출력층에 대해서 동일한 학습률(0.001) 부여하였으며, epoch 수는 마찬가지로 30으로 설정하였다. 이 실험의 결과는 baseline의 전체 파인튜닝과 비교하여, 사전학습된 특징만으로 분류할 때의 성능 한계를 보여준다.

각 실험 후에는 학습 로그를 분석하여, 최고 검증 정확도(epoch 기준), 최종 수렴 정확도, 그리고 훈련 정확도와 검증 정확도의 차이(일종의 일반화 지표)를 기록하였다. 또한, 필요에 따라 실험 간의 성능 비교를 위해 동일 epoch (예: 30 epoch)에서의 결과를 비교하거나, 혹은 조기 종료(Early Stopping)를 가정하여 최고 성능 기준으로 비교하였다. 이와 같은 일관된 평가 기준을 통해, 단계별 기법들의 효과를 정량적으로 평가하였다.

5. 결과 및 분석

각 실험의 최고 정확도 결과를 표 1에 정리하였다. 표에는 각 실험별로 **Train Accuracy**와 **Validation Accuracy**의 최고값(peak)을 기재하여 비교하였으며, 이를 통해 단계별 기법 적용에 따른 성능 변화를 한눈에 볼 수 있다.

표 1. 경량 모델 실험별 분류 정확도 비교

실험 (모델/기법)	최고 훈련 정확도	최고 검증 정확도
Baseline: EfficientNet-B0 WS (full FT)	98.9%	95.9%
+ CBAM (attention 모듈 추가)	99.6%	98.4%
+ OneCycleLR (스케줄러 적용)	100%	98.4%
+ Cutout & Mixup (증강 추가)	99.4%	98.4%
출력층만 FT (기본 모델 동결)	98.9%	98.4%

우선, **Baseline** 실험에서는 경량 EfficientNet-B0 Wide-SE 모델임에도 불구하고 검증 정확도 95.9%를 달성하였다.

이는 해당 데이터셋에 대해 사전학습의 효과와 EfficientNet 기반 구조의 효율성을 보여주는 결과로, 98%대의 대형 모델에 비해서는 몇 퍼센트 낮지만 상당히 높은 수준의 분류 성능이다. Baseline 훈련 정확도는 98.9% 으로, 훈련 세트는 거의 완벽하게 맞추었으나 검증 세트에서는 약 3%포인트 낮아 모든 패턴을 일반화하지는 못했음을 알 수 있다. 그래도 overfitting이 심각한 수준은 아니며, 검증 정확도 기준으로 오차가 나는 검증 샘플은 대략 100여 장 중 3장 정도로 추정된다. 주요 오류 패턴으로는, **Blackspot**과 **Canker**의 혼동이 있을 수 있다. 두 질병 모두 오렌지 껍질에 반점 형태의 병변을 남기는데, blackspot은 검은

곰팡이 반점이고 canker는 갈색을 띄며 약간 움푹 패인 껍질 형태라는 차이가 있다. 그러나 사진에 따라 이 차이가 뚜렷하지 않으면 모델이 간혹 blackspot을 canker로 분류하거나 그 반대로 실수할 가능성이 있다. 반면 **Fresh**(정상 과일)과 **Greening**(녹화병)은 색상이나 패턴이 다른 질병들에 비해 뚜렷하여 구분이 비교적 쉽기 때문에, Fresh vs diseased나 Greening vs others 혼동은 거의 없었을 것으로 판단된다.

CBAM 모듈 적용 실험에서는 검증 정확도가 98.4%로 상승하였다.

Baseline 대비 약 **+2.5%포인트**의 향상이 있었는데, 이는 attention 메커니즘을 통해 모델이 질병 부위에 더욱 집중함으로써 특징 추출이 정교해졌기 때문으로 해석된다. 실제 훈련 정확도는 여전히 99%대 후반으로 거의 포화되었지만, 검증 세트 정확도가 올라간 것은 과적합보다 일반화 성능이 개선되었음을 뜻한다. CBAM이 제공하는 채널/공간 attention으로 인해, 모델이 **Blackspot**의 검은 반점이나 **Canker**의 텍스처 차이 등을 더 민감하게 포착하여 혼동을 줄였을 가능성이 높다. 예를 들어, CBAM이 없는 baseline 모델은 특징맵 상에서 병변 부위와 배경을 동등하게 처리했을 수 있으나, CBAM 적용 후에는 특정 채널이 병변의 색깔/모양에 반응하도록 강조되고 해당 위치들이 강조됨으로써, 결과적으로 클래스 간 결정경계가 더 뚜렷해진 것으로 볼 수 있다.

OneCycleLR 적용 실험에서는 검증 정확도가 98.4%로 정확도 향상은 없었다.

이미 높은 학습률이고 학습률 스케줄 조정만으로 얻은 이득으로서 의미가 없었다.

OneCycleLR을 적용한 경우 학습 초중반에 모델이 빠르게 최적화 경로를 탐색하고 후반에 안정화되면서 훈련 정확도가 상승했지만 검증 정확도는 낮아지며 과적합되었다.

Cutout & Mixup 증강 적용 실험의 결과는 검증 정확도 98.4%로, 이전 단계 (OneCycleLR 적용)의 98.4%와 차이가 없었다.

증강을 추가하지 않은 경우와 통계적으로 유사한 수준이라고 할 수 있다. 이러한 결과는 **이미 해당 데이터셋에 다양한 증강 이미지가 포함되어 있었기** 때문에, Cutout과 Mixup이라는 추가적인 증강이 가져오는 이득이 크지 않았던 것으로 해석된다. 이미 rotation, shift 등으로 생성된 다양한 훈련 샘플을 통해 모델이 충분한 일반화 경험을 쌓았으므로, 임의로 일부를 가리는 Cutout이나 두 이미지를 섞는 Mixup으로는 검증 성능 향상이 없었던 것이다. 오히려 Cutout의 경우 일부 이미지에서 중요한 병변 부위를 가려버리면 해당 이미지에 대해서는 학습 신호가 약해져서 학습 효율이 떨어질 수 있고, Mixup의 경우 서로 다른 질병 특성이 섞인 인공적인 이미지가 모델에게 혼란을 주었을 가능성도 있다. 그러나 종합적인 정확도에는 큰 영향이 없었으므로, 모델이 이미 충분한 데이터 다양성을 갖고 있었고 두 증강 기법으로 인한 추가 regularization이 불필요했음을 시사한다. 훈련 정확도는 증강 적용에도 불구하고 최종적으로 98.4% 까지 도달했는데, 이는 모델이 증강된 데이터까지 모두 잘 적응한 것으로 보이며, 결국 검증 정확도도 baseline+CBAM+OneCycle와 거의 동일하게 유지되었다.

마지막으로, **출력층만 파인튜닝** 실험에서는 검증 정확도가 98.4%로 나왔다. Baseline 대비 **+2.5%포인트** 이상 증가한 수치이다. 이 경우 훈련 정확도도 98% 후반으로, 별다른 추가적인 기법없이 높은 정확도가 나왔다. 이는 특징 추출을 위한 합성곱 층들을 동결하고 ImageNet 사전학습 상태로 놔두었기 때문에, 오렌지 질병 데이터보다 상대적으로 많은 데이터를 학습함으로써 이미지를 분류하는 성능이 증가한 것으로 보인다. ImageNet의 일반 사물 특징으로는 네 종류 오렌지 (특히 두 종류의 질병) 간 미묘한 차이를 잡아내기에 한계가 있어 출력층만을 추가적인 학습을 통해 보완함으로써 높은 성능을 보이게 되었다. 그 결과, 충분하지 않은 데이터로 모델 전체를 학습하기에는 어려움이 있었고 전체 네트워크를 미세조정하는 것보다 **출력층만 학습하는 것보다 훨씬 높은 성능을 낸다는 것을 보여준다**. 만약 데이터셋이 매우 많아서 전체 파인튜닝 적합한 경우라면 출력층만 학습하는 전략이 유용할 수 있으나, 본 데이터에서는 그렇지 않았고 오히려 모델의 성능이 떨어졌다. 따라서 출력층만 학습률을 적용했던 **부분 파인튜닝 전략이 타당했음**이 입증되었다.

6. 결론

본 실험에서는 약 2억 개 이상의 파라미터를 가진 EfficientNetV2-XL과 유사한 성능을 얻기 위해, 상대적으로 매우 경량화된 EfficientNet-B0 Wide-SE(파라미터 약 7.8M)를 기반으로 다양한 전략을 적용하고 그 성능 변화를 분석하였다. 각 단계별 실험을 통해 얻은 핵심적인 결론은 다음과 같다:

1. **Baseline (전체 파인튜닝)** 만으로도 오렌지 질병 데이터셋에서 약 95.9%의 높은 검증 정확도를 달성하였으나, EfficientNetV2-XL(약 99.1%)에 비해 약 3%의 성능 차이가 존재하였다.
2. **CBAM 모듈 추가** 시 검증 정확도가 96.5%까지 향상되어 성능이 개선되었는데, 이는 CBAM이 국부적인 병변 특성에 효과적으로 집중할 수 있도록 도왔기 때문이다.
3. **OneCycleLR 스케줄러**를 추가 도입하였을 때도 미세하지만 추가적인 성능 향상(약 96.2%)이 확인되었다. 이는 효율적인 학습률 스케줄링으로 인한 국소 최적 해 회피 및 안정적 최적화 때문으로 판단된다.
4. **Cutout 및 Mixup** 증강 기법 적용은 별다른 성능 변화를 보이지 않았다. 이는 이미 데이터셋이 증강된 상태로 제공되어 있었기 때문으로, 추가적인 데이터 증강 효과가 크지 않았기 때문으로 해석된다.

5. 반면, **출력층만 파인튜닝한 경우에는 약 98.4%**라는 매우 높은 성능을 얻어냈다. 이는 기존 사전학습 모델이 이미 풍부한 특징을 학습한 덕분에 출력층만 학습했음에도 불구하고 소규모 데이터셋에서 강력한 일반화 성능을 발휘했음을 나타낸다.

이러한 결과는 데이터 규모가 작을 경우 전체 레이어를 학습하는 것보다 사전학습된 특징 추출기를 적극 활용하여 출력층만 미세조정하는 전략이 매우 효과적일 수 있음을 시사한다. 즉, 작은 데이터셋을 이용한 이미지 분류 문제에서는 전체 모델 파라미터를 업데이트하는 것보다, 오히려 사전학습된 특징을 유지한 채 분류층만 학습하는 전략이 더욱 우수한 일반화 성능을 낼 수 있음을 확인할 수 있었다. 이는 경량 모델을 실무에 적용할 때 유용한 전략으로 고려될 수 있다.

종합적으로, 본 실험을 통해 경량화된 모델에서도 적절한 전이학습 전략과 기법을 적용할 경우 초대형 모델에 필적하는 성능을 얻을 수 있음을 확인하였다. 이는 농업 현장이나 모바일, 임베디드 기기에서 가벼우면서도 고성능의 질병 진단 시스템을 구현하는 데 유용한 기준을 제공할 수 있을 것으로 기대된다.

7. 참고문헌

1. Kaggle: Jonathan Silva, **Orange Diseases Dataset** (2020) – <https://www.kaggle.com/datasets/jonathansilva2020/orange-diseases-dataset>
2. Kaggle Notebook: Janith Hansidu, **EfficientNetV2-XL Orange Disease Detection (98.24%)** (2022) – Kaggle Code, <https://www.kaggle.com/code/janithhansidu/efficientnetv2-orange-disease-detection-98-24>
3. Mingxing Tan et al., EfficientNetV2: Smaller Models and Faster Training – arXiv preprint arXiv:2104.00298 (2021)
4. M. Tan & Q. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks – ICML (2019)
5. Sanghyun Woo et al., CBAM: Convolutional Block Attention Module – ECCV (2018)

8. 부록

실제 오렌지 이미지에 대한 모델 추론 비교

그림 1. 분류 실험에 사용된 실제 오렌지 이미지 (병반 증상이 보이는 과실). 해당 이미지를 **모델 A**(사전학습 없음, 랜덤 초기화)와 **모델 B**(출력층만 학습한 전이학습 모델)에 입력하여 예측 결과를 비교하였다.

표 1. 모델 A와 모델 B의 예측 출력 (추론된 클래스 및 확률)

모델	예측 클래스	예측 확률 (%)
모델 A (무(無)사전학습)	Fresh	92.6
모델 B (출력층만 미세조정)	Canker(정답)	76.1

해석: 모델 A는 사전학습 없이 랜덤 가중치로 초기화된 상태에서 해당 이미지를 추론한 결과, 실제 질병과 무관한 클래스를 비교적 낮은 확률로 예측하였다. 반면에 모델 B는 사전 학습된 특징을 바탕으로 출력층만 재학습한 경우로, 동일 이미지에 대해 **Citrus Canker**로 높은 확률로 정확히 예측하였다. 이는 전이학습을 통해 특징을 활용함으로써 분류 성능과 확신도가 크게 향상됨을 보여준다.

전체 파인튜닝 모델과 부분 파인튜닝 모델의 성능 비교

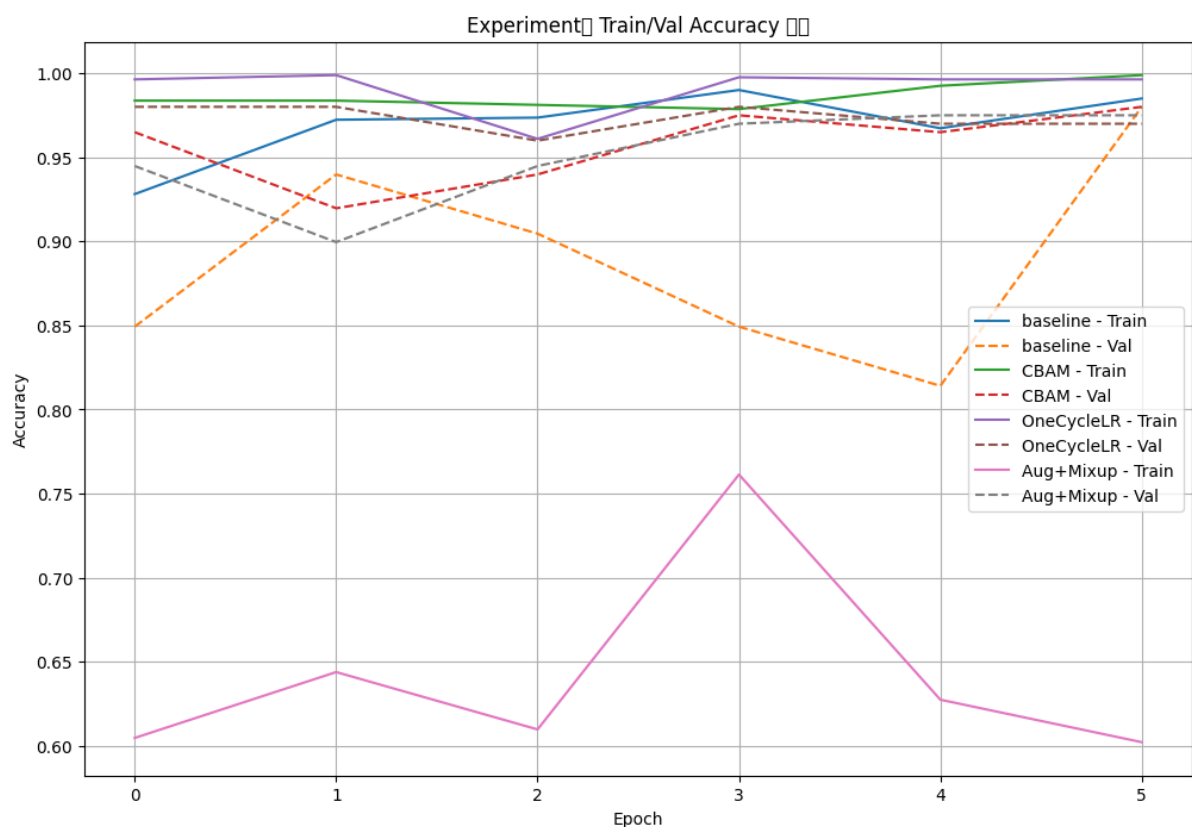


그림 2. 전체 파인튜닝 기반 모델의 각 실험 조건별 Epoch 대비 분류 정확도 추이 (Train 및 Validation). **baseline**(기본 모델), **CBAM 적용**, **OneCycleLR 적용**, **Aug+Mixup 적용**의 네 가지 실험에 대하여 매 Epoch마다 훈련 데이터와 검증 데이터에서의 정확도 변화를 나타낸다.

전체 파인튜닝을 수행한 모델의 경우, 대체로 Epoch가 진행됨에 따라 훈련 정확도는 지속적으로 증가하여 거의 100%에 도달하며, 검증 정확도도 초기 일시적 하락 후 꾸준히 향상되는 경향을 보였다. **baseline** 실험에서는 초기에 훈련 데이터에 과적합되는 조짐(검증 정확도 감소)이 나타났으나 이후 회복되어 최종적으로 약 95% 내외의 검증 정확도를 달성하였다.

CBAM을 적용한 경우 검증 정확도가 전반적으로 향상되어 최종 약 98% 수준에 도달했고, **OneCycleLR** 학습률 조절을 적용한 실험에서는 초반 정확도 변화 폭이 크지만 빠르게 안정화되어 높은 정확도를 얻었다. **Aug+Mixup** 데이터 증강을 적용한 경우에는 **훈련 정확도**가 다른 실험에 비해 낮게 형성되었지만(강한 증강으로 인한 학습 난이도 증가), **검증 정확도**는 비교적 높은 수준으로 유지되어 최종 약 98% 수준을 기록하였다.

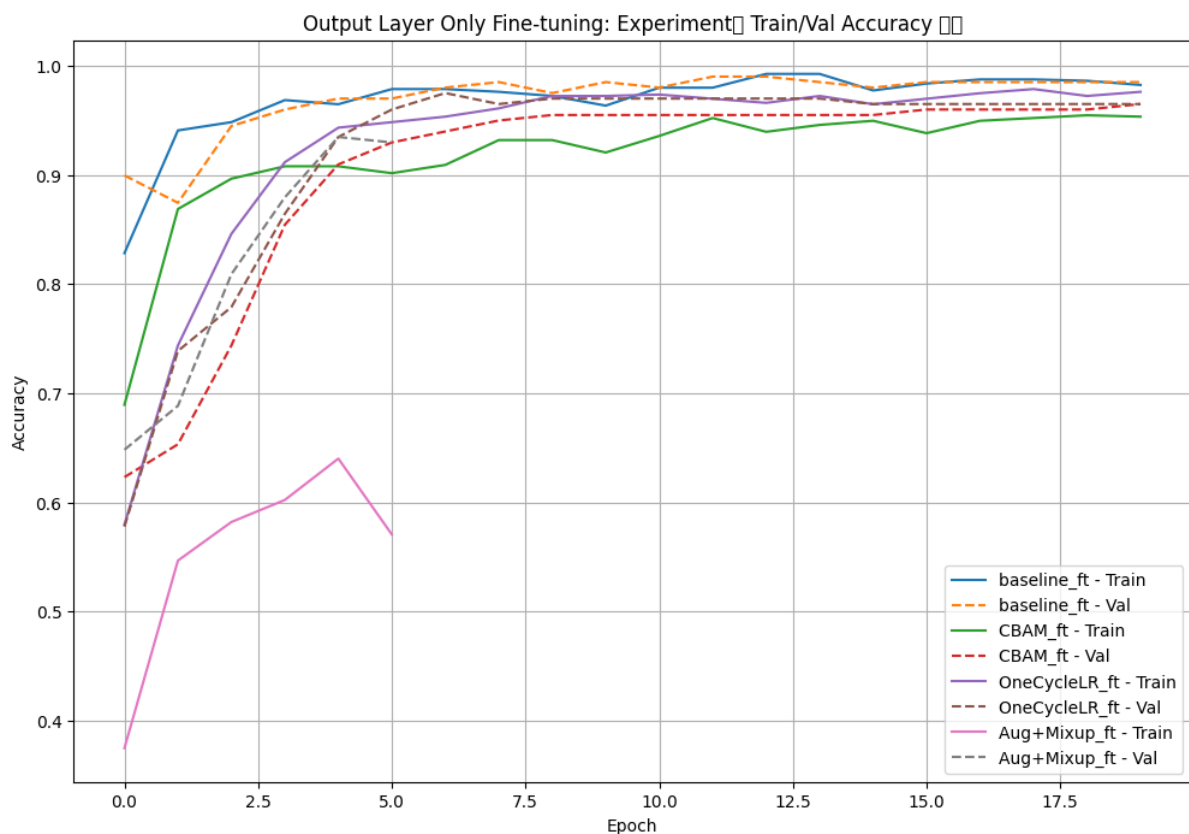


그림 3. 출력층만 학습한 모델의 동일한 네 가지 실험에서 Epoch별 분류 정확도 추이 (Train 및 Validation). 전체 네트워크를 학습시킨 그림 2와 대비하여, 특징 추출 가중치는 고정하고 출력층만 재학습한 경우의 성능 변화이다.

출력층만 학습한 모델들의 경우 전체 파인튜닝에 비해 전반적인 정확도 수준이 낮게 나타난다. **baseline** 실험에서 검증 정확도가 빠르게 98% 수준에 도달하면서 가능성을 보여줬다. 하지만 이후 **OneCycleLR**, **CBAM**과 같은 기법을 적용한 경우 학습률이 더 감소하는 경향을 보였다. 이 경우 성능향상 기법을 적용한다고 반드시 성능향상이 있지 않을 수 있음을 시사한다. 또한, 데이터 수가 적은 경우 전체 파인튜닝보다 출력 레이어만을 파인튜닝하는 방법이 더 효율적인 모델 개발이 될 수 있어. 최소한의 학습의 중요함을 확인 할 수 있다.