

1. 데이터 전처리 및 증강

- **AutoAugment / RandAugment**
 - 이미지 증강 기법으로, 모델 일반화 성능 향상에 기여.
- **정규화(Normalization)**
 - 학습 안정성을 위해 데이터셋 평균 및 표준편차 계산 후 정규화 적용.
- **Mixup & CutMix**
 - 서로 다른 두 이미지를 혼합하여 학습, 결정 경계를 더 일반화시킴.
- **Label Smoothing**
 - 라벨을 부드럽게 처리해 과적합 방지 및 학습 안정화.

2. 모델 구조

- **EfficientNetV2-M (tf_efficientnetv2_m)**
 - 성능과 연산 효율성을 동시에 갖춘 CNN 모델 구조.
- **Dropout / DropPath**
 - 과적합 방지를 위해 뉴런을 임의로 제거하거나 경로를 랜덤 차단.

3. 최적화 및 학습 전략

- **Optimizer: Adam**
 - 일반적으로 빠른 수렴과 안정성을 보장하는 최적화 기법.
- **Learning Rate Scheduler: CosineAnnealingWarmRestarts**
 - 학습률을 코사인 형태로 주기적으로 감소시켜 최적화 안정성 향상.
- **EMA (Exponential Moving Average)**
 - 가중치의 지수 이동 평균을 저장해 더욱 안정적인 검증 성능 확보.
- **AMP (Automatic Mixed Precision)**
 - 연산을 FP16과 FP32 혼합으로 수행하여 학습 속도 및 메모리 효율 향상.

4. 평가 및 저장

- **평가지표: Accuracy, F1-score (macro)**
 - 정확도 외에도 클래스 불균형에 민감한 F1-score 사용.
- **Best Model 저장 조건**
 - 검증 정확도가 가장 높을 때만 모델 저장 (best_val_acc 기준).

5. 기타 성능 최적화

- **DataLoader 최적화**
 - pin_memory, persistent_workers, prefetch_factor 사용으로 데이터 로딩 병목 감소.
- **Seed 고정**
 - 재현성을 위해 random, numpy, torch 시드 고정.