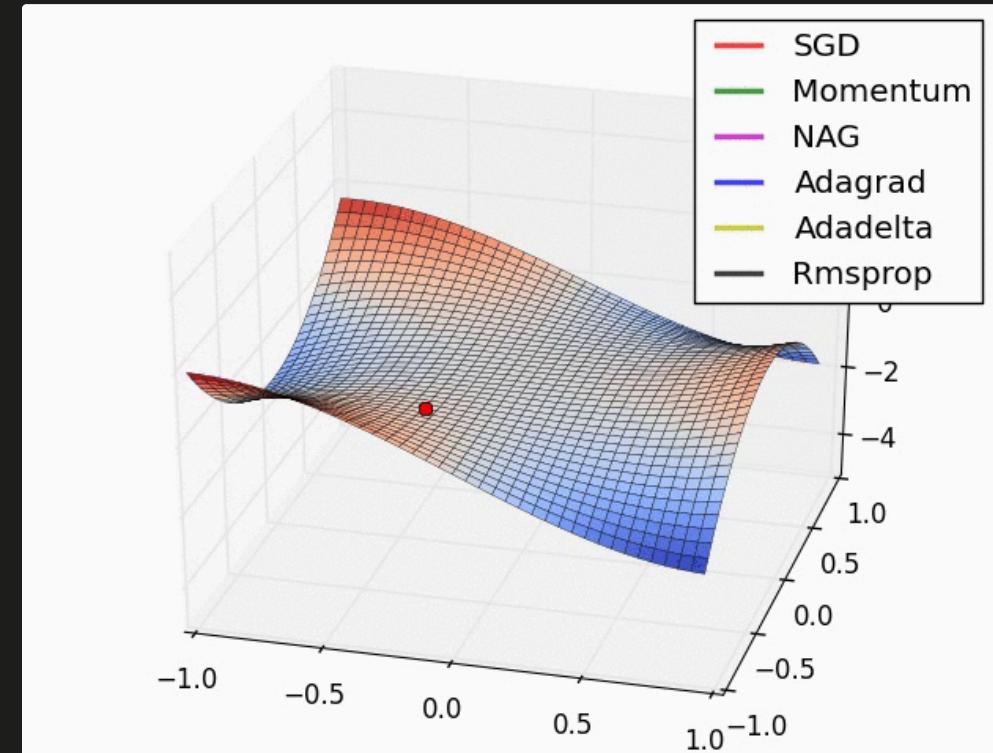
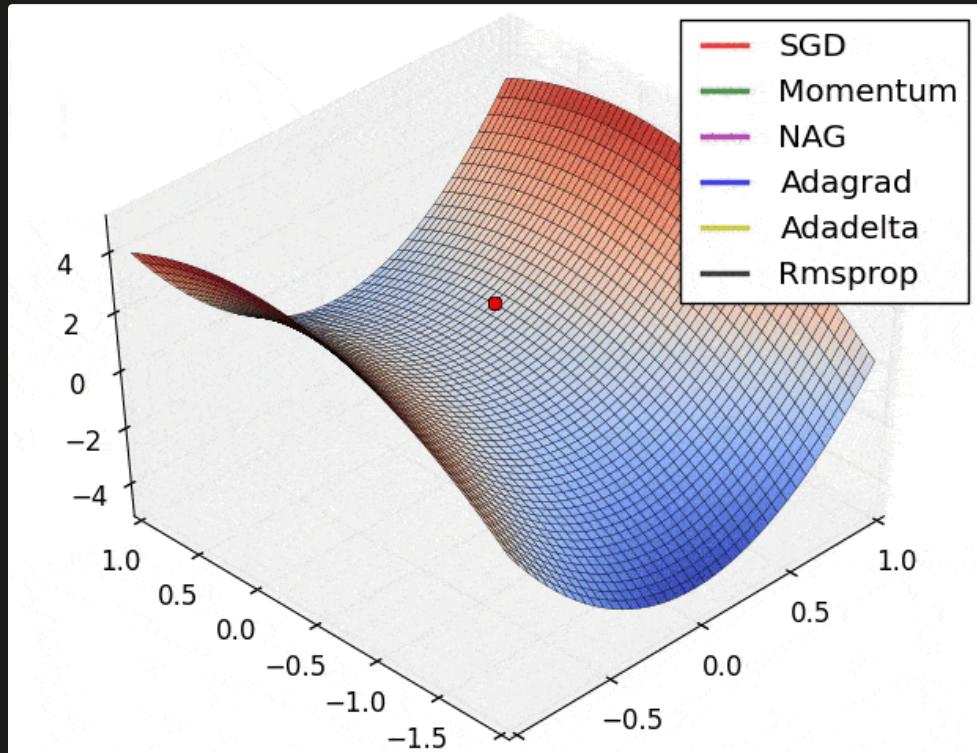




해파리 이미지 분류기: 하이퍼파라미터 튜닝의 영향 분석

처음엔 옵티마이저 최적화였다?



위 그래프는 옵티마이저들이 어떻게 오차의 최저점을 찾아가는지 형상화한 것입니다. 각 옵티마이저는 서로 다른 방식으로 손실 함수의 최소값을 향해 이동하며, 그래프를 통해 각 알고리즘의 특성과 수렴 속도를 비교할 수 있습니다.

옵티마이저 토너먼트



1 참가자: 옵티마이저

각각의 옵티마이저가 해파리 이미지 분류 성능을 겨릅니다.

2 목적: 최고의 성능 거두기

정확도와 효율성을 겨루는 토너먼트 방식으로 진행됩니다.

3 결과: 우승 옵티마이저

최종적으로 가장 뛰어난 성능을 보인 옵티마이저가 우승합니다.

목표 수정 이유는?

1

옵티마이저 심층분석의 한계

수학적 접근의 복잡성과 어려움

2

목표 변경

하이퍼파라미터 영향 심층 분석

3

최종 목표

하이퍼파라미터 튜닝을 통한 성능 개선 메커니즘 이해



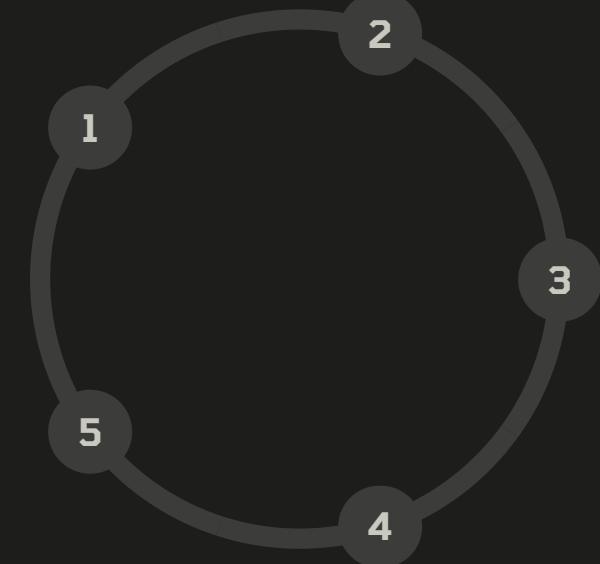
캐글 데이터셋



6개 해파리 종

테스트 4%

40장 성능평가용



979장 이미지

종별 약 163장씩 분포

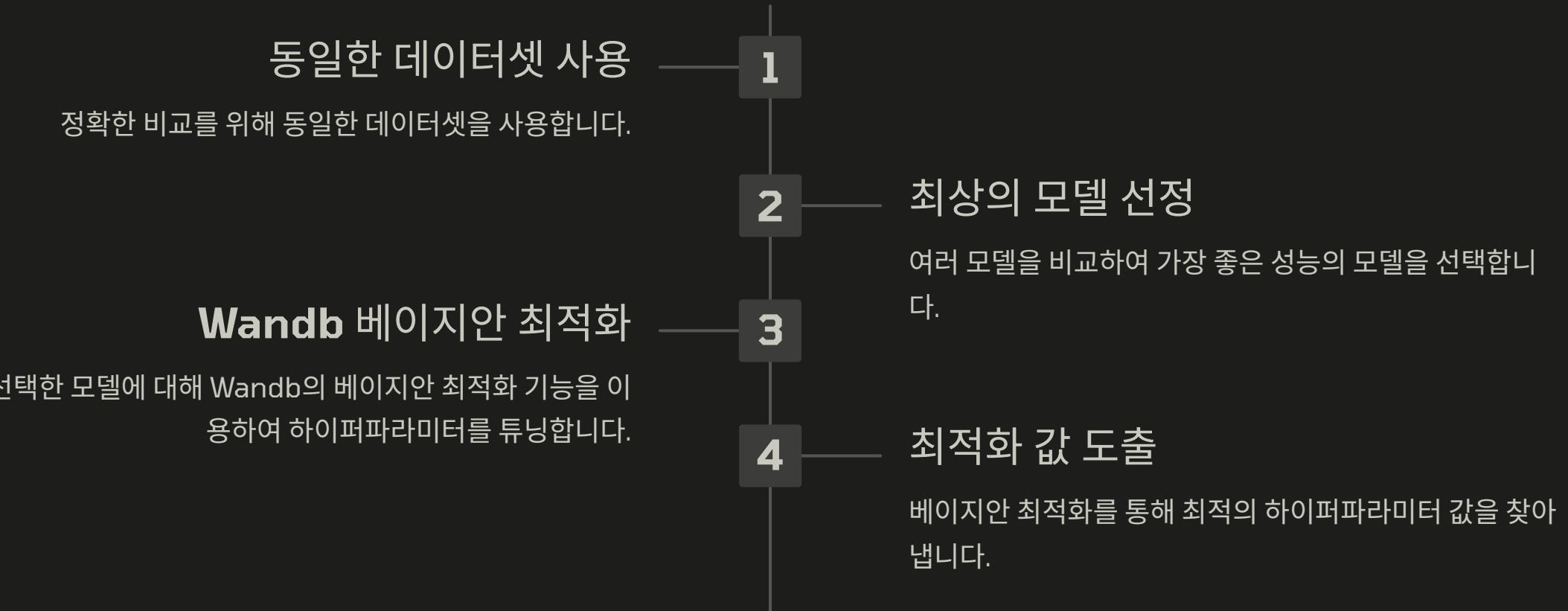
훈련 데이터 **92%**

900장 학습용: 종별 150장씩 균등 분포

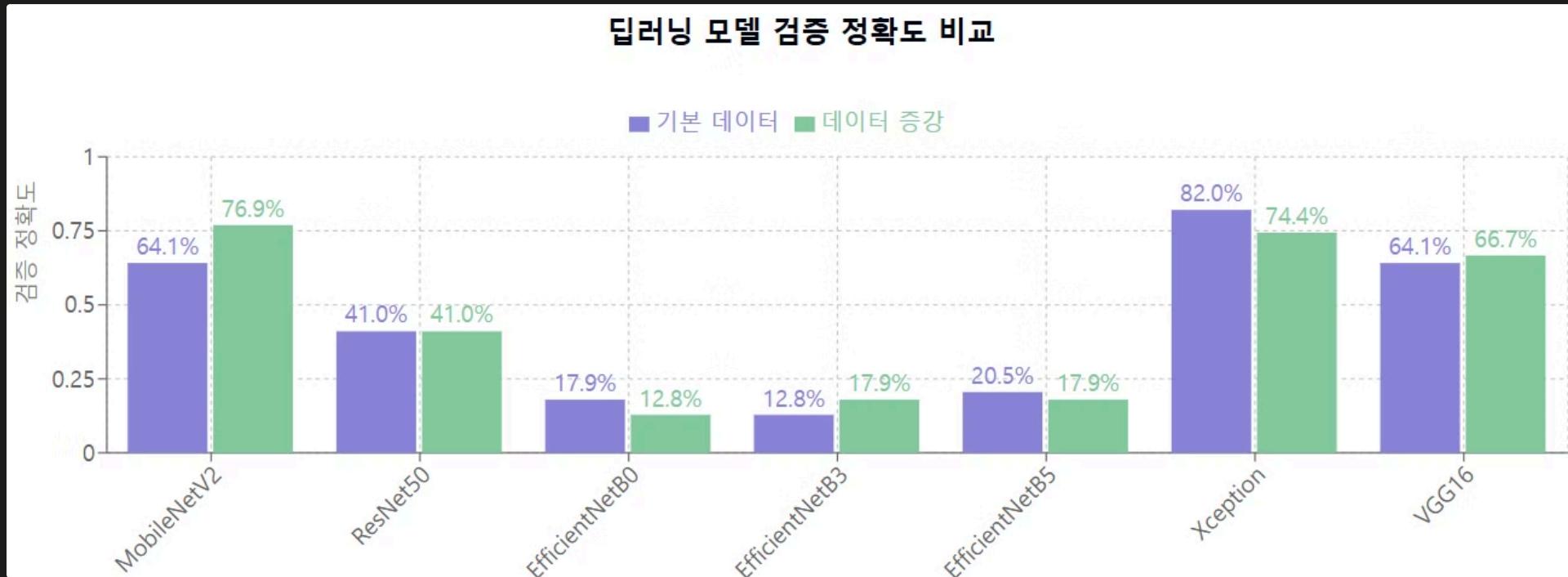
검증 **4%**

39장 검증용

하이퍼파라미터 최적화 프로세스



모델 선택: Xception, MobileNetV2



경량화 아키텍처



높은 연산 효율성



뛰어난 성능



전이학습 용이

하이퍼파라미터 튜닝 전략

그리드 서치

모든 가능한 조합 실험

- 철저한 공간 탐색
- 계산 비용 높음
- 최적점 보장

랜덤 서치

무작위 샘플링 기반 탐색

- 효율적 공간 탐색
- 적은 실험 횟수
- 계산 효율성 높음

베이지안 최적화

확률 모델 기반 탐색

- 이전 결과 활용
- 지능적 파라미터 선택
- 빠른 수렴 속도

베이지안 최적화를 선택 이유

기존 방법론의 한계를 극복하고 효율적인 하이퍼파라미터 최적화를 달성하기 위함입니다.



탐색 효율성

이전 실험 결과를 활용해 가능성 높은 영역에 집중합니다.



계산 시간 절약

그리드 서치보다 적은 실험으로 최적값에 더 빠르게 수렴합니다.



지능적 탐색

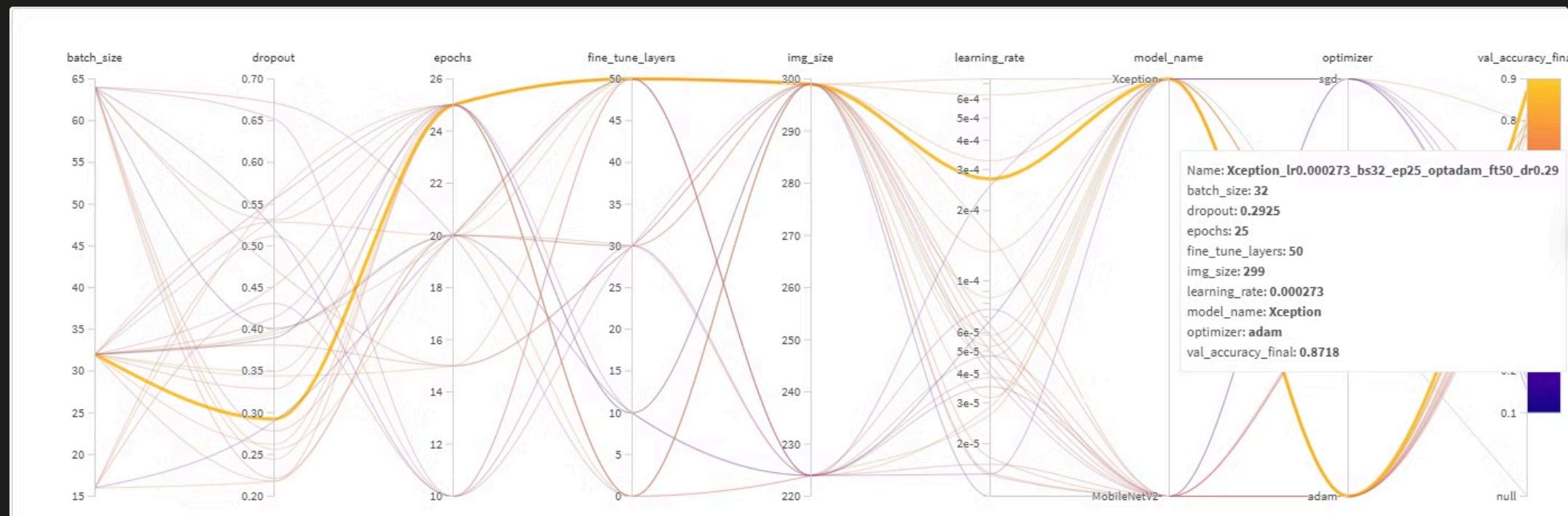
학습 모델 기반으로 다음 실험 지점을 지능적으로 선택합니다.



hparam_optimizer_journey_star.mp4



튜닝 실험 #1 (케글 데이터셋)



튜닝 실험 #1 (케글 데이터셋)

파라미터	범위	최적화값
모델	Xception, MobileNetV2	Xception
learning_rate	min: 1e-5, max: 1e-3	0.000273
batch_size	16, 32, 64	32
dropout	min: 0.2, max: 0.7	0.2925
optimizer	adam, rmsprop, sgd	adam
fine_tune_layers	0, 10, 30, 50	50
img_size	224, 299	299
epochs	10, 15, 20, 25	25
val_acc		0.8718

```
sweep_config = {
    'method': 'bayes',
    'metric': {'name': 'val_accuracy', 'goal': 'maximize'},
    'parameters': {
        'model_name': {'values': ['Xception', 'MobileNetV2']},
        'learning_rate': {'distribution': 'log_uniform_values',
                         'min': 1e-5, 'max': 1e-3},
        'batch_size': {'values': [16, 32, 64]},
        'dropout': {'min': 0.2, 'max': 0.7, 'distribution': 'uniform'},
        'optimizer': {'values': ['adam', 'rmsprop', 'sgd']},
        'fine_tune_layers': {'values': [0, 10, 30, 50]},
        'img_size': {'values': [224, 299]},
        'epochs': {'values': [10, 15, 20, 25]}
    },
    'early_terminate': {
        'type': 'hyperband',
        'min_iter': 3,
        'eta': 3,
        's': 2
    }
}
```

데이터셋이 달라진다면?

데이터셋을 변경하면 모델 성능과 하이퍼파라미터 최적화 결과에 어떤 영향을 미칠까?

▼ Optimizer도 달라질까?

데이터셋이 변경되면 모델이 빠르게 수렴하는 방식도 달라질 수 있습니다. 이전에 효과적이었던 Optimizer가 새로운 데이터에서는 적합하지 않을 수 있습니다.

▼ Learning Rate는 그대로 써도 될까?

데이터의 스케일이 달라지면 최적의 Learning Rate도 변경될 필요가 있습니다. 이전에 잘 작동했던 Learning Rate가 새로운 데이터에서는 적절하지 않을 수 있습니다.

▼ Dropout은 얼마나 줄여야 할까?

데이터의 복잡도에 따라 과적합 위험도가 달라지므로, 적절한 Dropout 비율도 변경해야 할 수 있습니다.

▼ 몇 층을 고쳐야 할까? (fine_tune_layers)

새로운 데이터가 이전 데이터보다 더 일반적이거나 특수한 경우, 모델 깊이를 조절해야 할 필요가 있습니다.

▼ 검증 정확도(val_accuracy)의 기준도 달라지는 건 아닐까?

클래스의 수나 분포가 달라지면 동일한 검증 정확도 수치도 다른 의미를 가질 수 있습니다.



데이터 분포 변화

새로운 데이터셋의 통계적 특성이 하이퍼파라미터 최적값을 변화시킵니다.



모델 일반화 능력

다양한 데이터셋에서 일관된 성능을 보이는 모델이 더 강건합니다.



재최적화 필요성

새 데이터셋에 맞춰 하이퍼파라미터를 다시 최적화해야 합니다.

튜닝 실험 #2 (증강 데이터셋)

데이터 증강 적용

원본 해파리 이미지에 다양한 증강 기법을 적용했습니다.

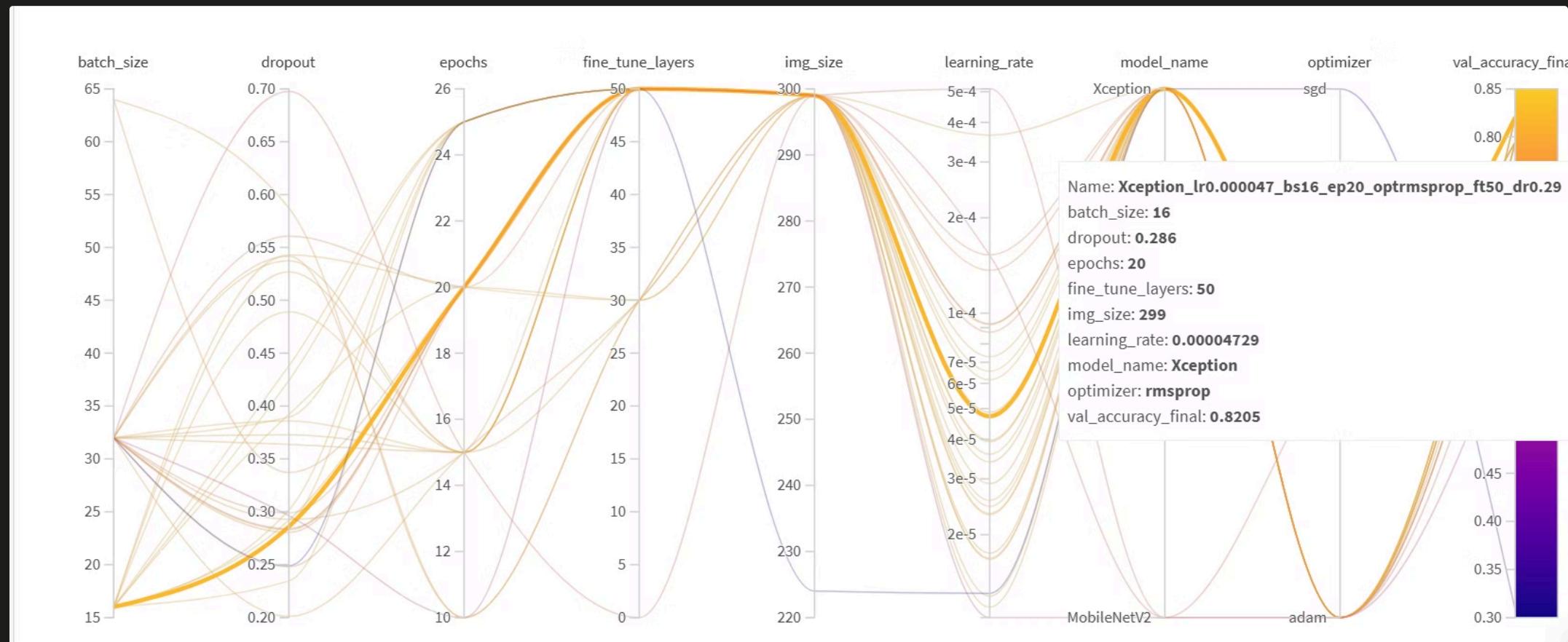
- 회전, 뒤집기 (수평, 수직)
- 이미지 이동, 크기 조정, 회전 (랜덤하게)
- 밝기, 대비, 색상 변경
- 가우시안 노이즈 추가
- 이미지 흐림 효과 (블러)
- 원근감 왜곡, 그리드 왜곡, 탄성 변형
- **CLAHE**: 대비 향상

증강 효과

데이터 증강으로 학습 데이터의 다양성이 크게 향상되었습니다.

- 각 클래스 당 500개 총 3000개의 학습데이터
- 클래스 불균형 완화 예측
- 모델 일반화 능력 강화 예측

튜닝 실험 #2 (증강 데이터셋)

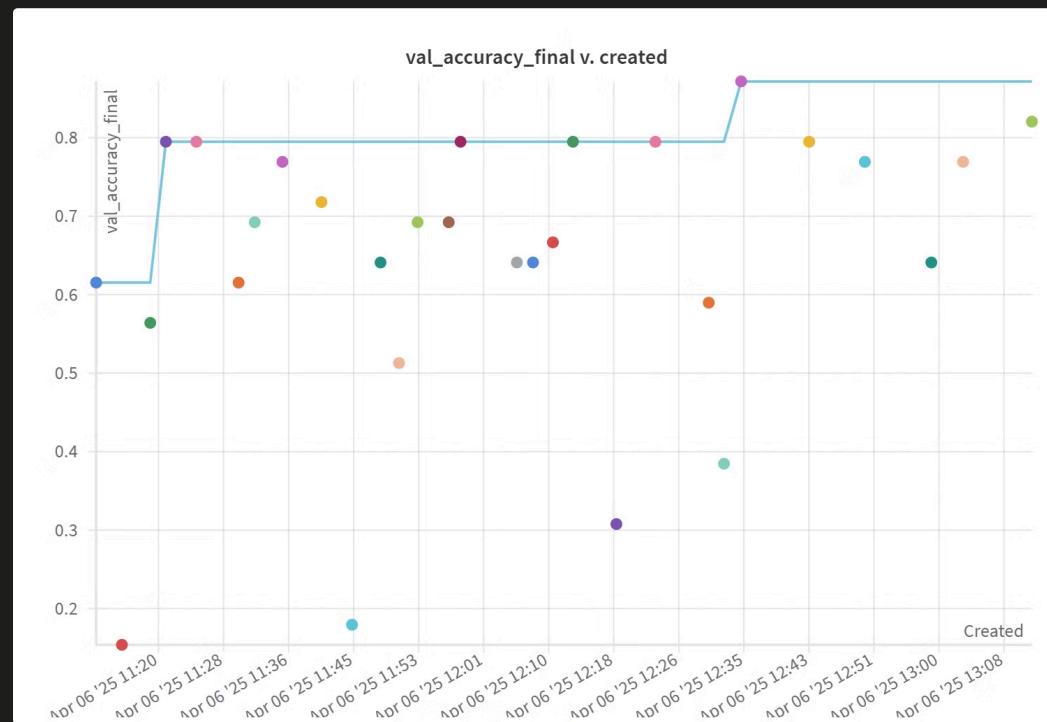


튜닝 실험 #1 vs #2

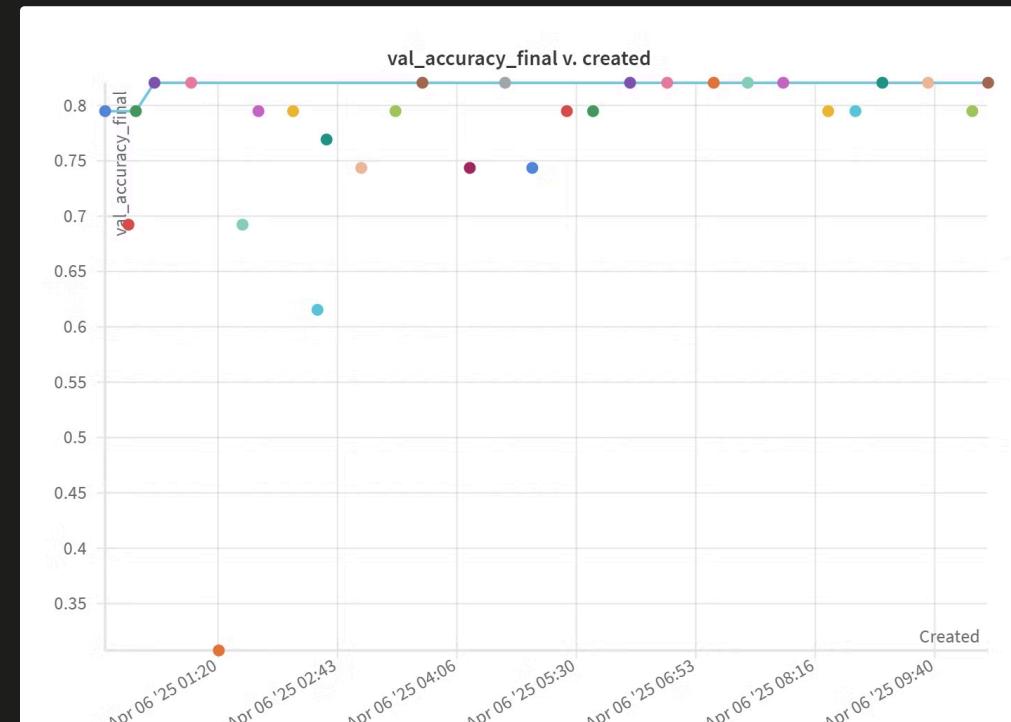
	캐글 데이터셋	증강 데이터셋
파라미터	최적화값	최적화값
모델	Xception	Xception
learning_rate	0.000273	0.00004729
batch_size	32	16
dropout	0.2925	0.286
optimizer	adam	rmsprop
fine_tune_layers	50	50
img_size	299	299
epochs	25	20
val_acc	0.8718	0.8205

튜닝 실험 #1 vs #2

#1 캐글 데이터 : 성능 불안정적, 가끔 80% 후반대로 고성능



#2 증강 데이터 : 성능 안정적, 평균적으로 80% 초반대



캐글 데이터셋

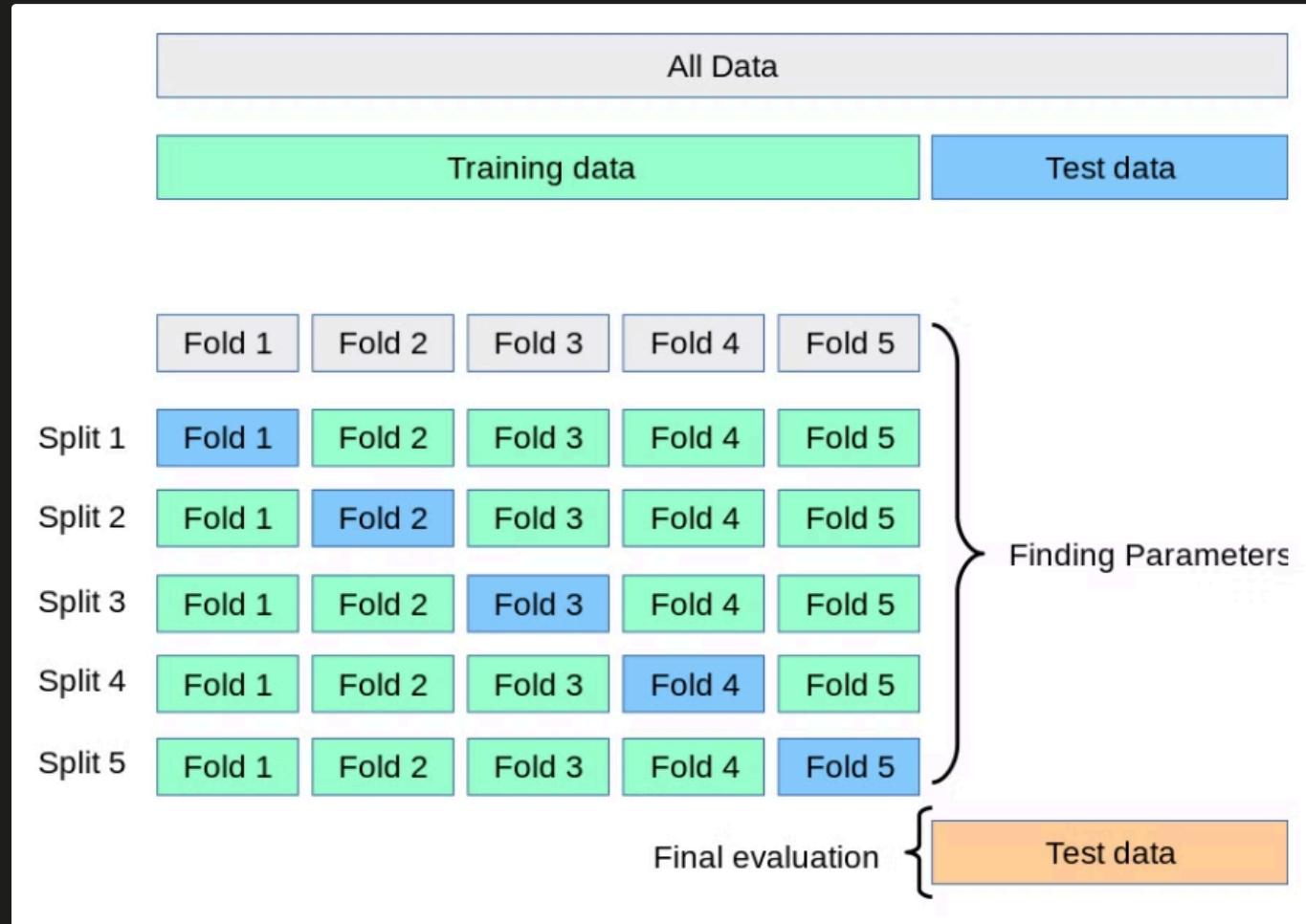
증강 데이터셋

불안정성

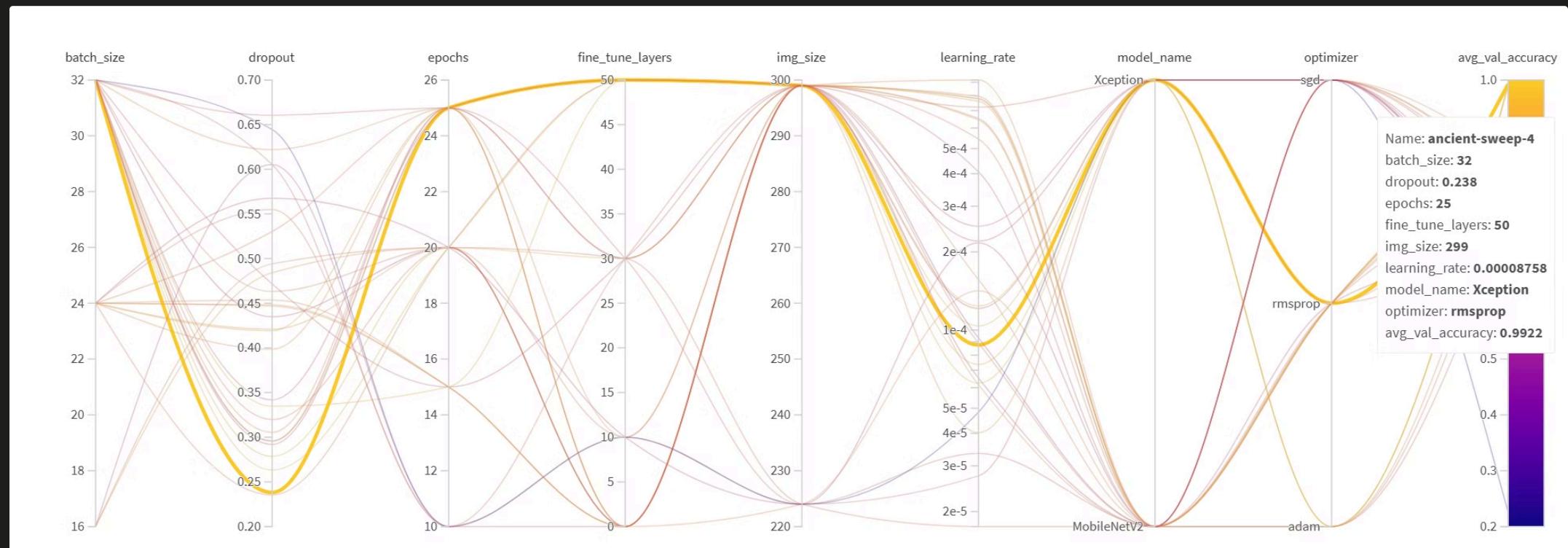
?

낮은 성능

K-fold 교차검증



튜닝 실험 #3 [캐글 데이터셋]

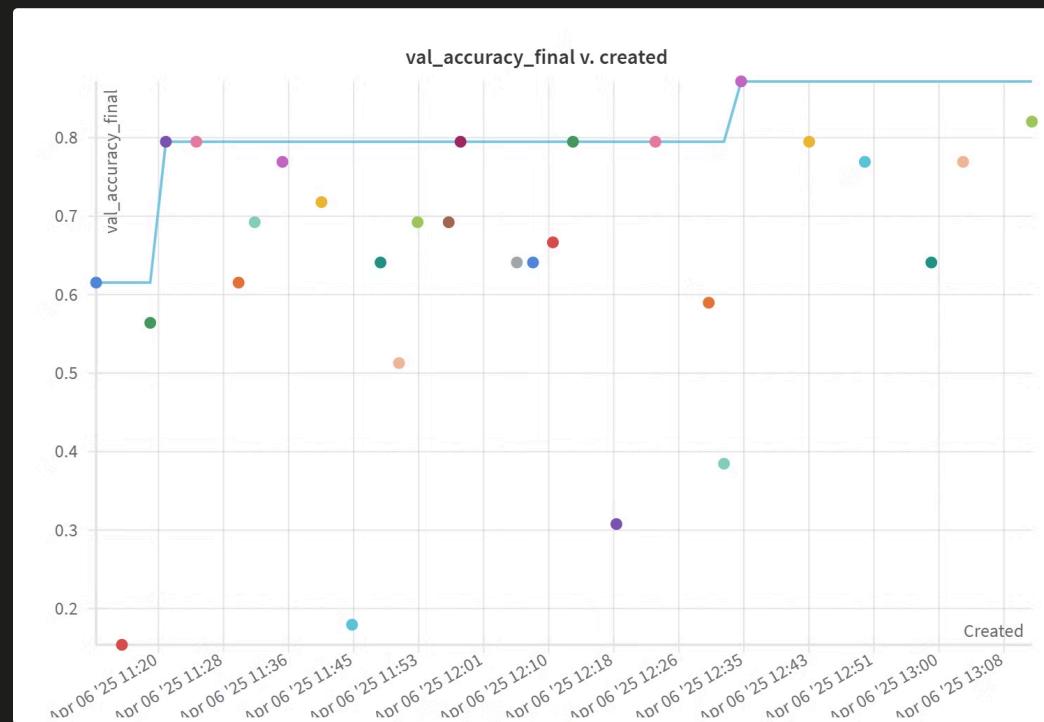


튜닝 실험 #1 vs #3

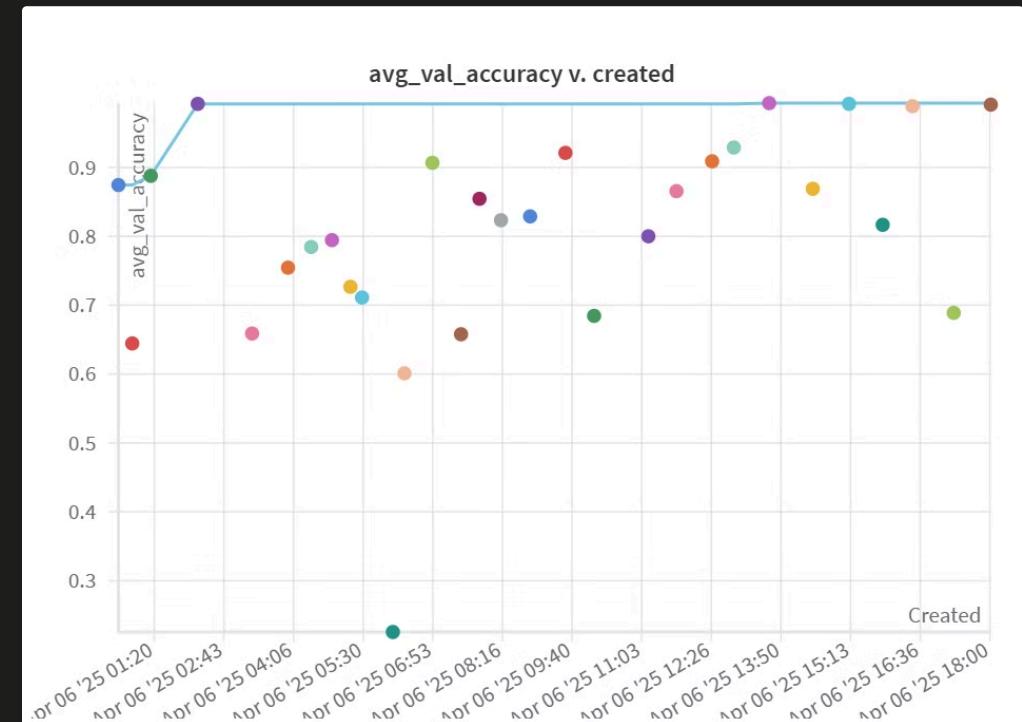
	캐글 데이터셋	캐글 데이터셋
파라미터	최적화값	최적화값
모델	Xception	Xception
learning_rate	0.000273	0.00008758
batch_size	32	32
dropout	0.2925	0.2841
optimizer	adam	rmsprop
fine_tune_layers	50	10
img_size	299	299
epochs	25	25
val_acc	0.8718	0.9922

튜닝 실험 #1 vs #3

#1 캐글 데이터 : 성능 불안정적, 가끔 80% 후반대로 고성능



#3 k-fold 교차 검증 : 안정적, 99% 고성능



하이퍼파라미터 튜닝 실험으로 알게된 점

- 패턴을 찾기는 어려움
- wandb를 사용하면 최적의 하이퍼파라미터 조합을 쉽게 찾아줌
- 데이터가 적은 상태에서는 k-fold 교차검증이 안정적임(운빨 타는 데이터 split을 평균 내서 완화 가능)

모델 변경 vs 하이퍼파라미터 튜닝

정확도의 표준편차



정확도의 범위 (최대-최소)





실험의 한계점



데이터셋

데이터 양이 절대적으로 부족. 라벨링 오류



하이퍼파라미터 범위 설정

베이지안 최적화의 한계



시간 제약

적은 비용으로 효율을 높이는 방법

플러터 시연

Google Docs

Screen_recording_20250407_094357.webm

