

해파리 감식반 1.0

MobileNetV2 기반 AI 해파리 특화 실시간 분류 서비스

DeepSea2

“스치면 병원행”…대형 독성 해파리가 몰려온다!



여름철 해수욕객 증가 시기(6~8월)
에 집중적으로 발생하며, 특히 독성
해파리로 인한 심각한 사례 증가 추세



기후변화와 양쯔강 하구 환경 변화
로 노무라입깃해파리 등 독성 해파
리 개체 수 크게 증가



데이터 부족으로 예측 정확도 한계,
현재 신고 앱 서비스 중단으로 실시
간 대응 어려움



해파리를 발견하고, 경험치를 쌓고, 바다를 지켜라

해파리 감식반 GO는 심각해지는 해파리 쏘임 사고에 대응하기 위한 모바일 애플리케이션입니다.

사용자가 발견한 해파리를 AI로 실시간 분류하고, 위험도와 대처법을 안내하며, 게임적 요소를 통해 지속적인 참여를 유도합니다. 수집된 데이터는 해파리 분포 모니터링과 안전 정책 수립에 활용됩니다.

📷 AI 해파리 인식

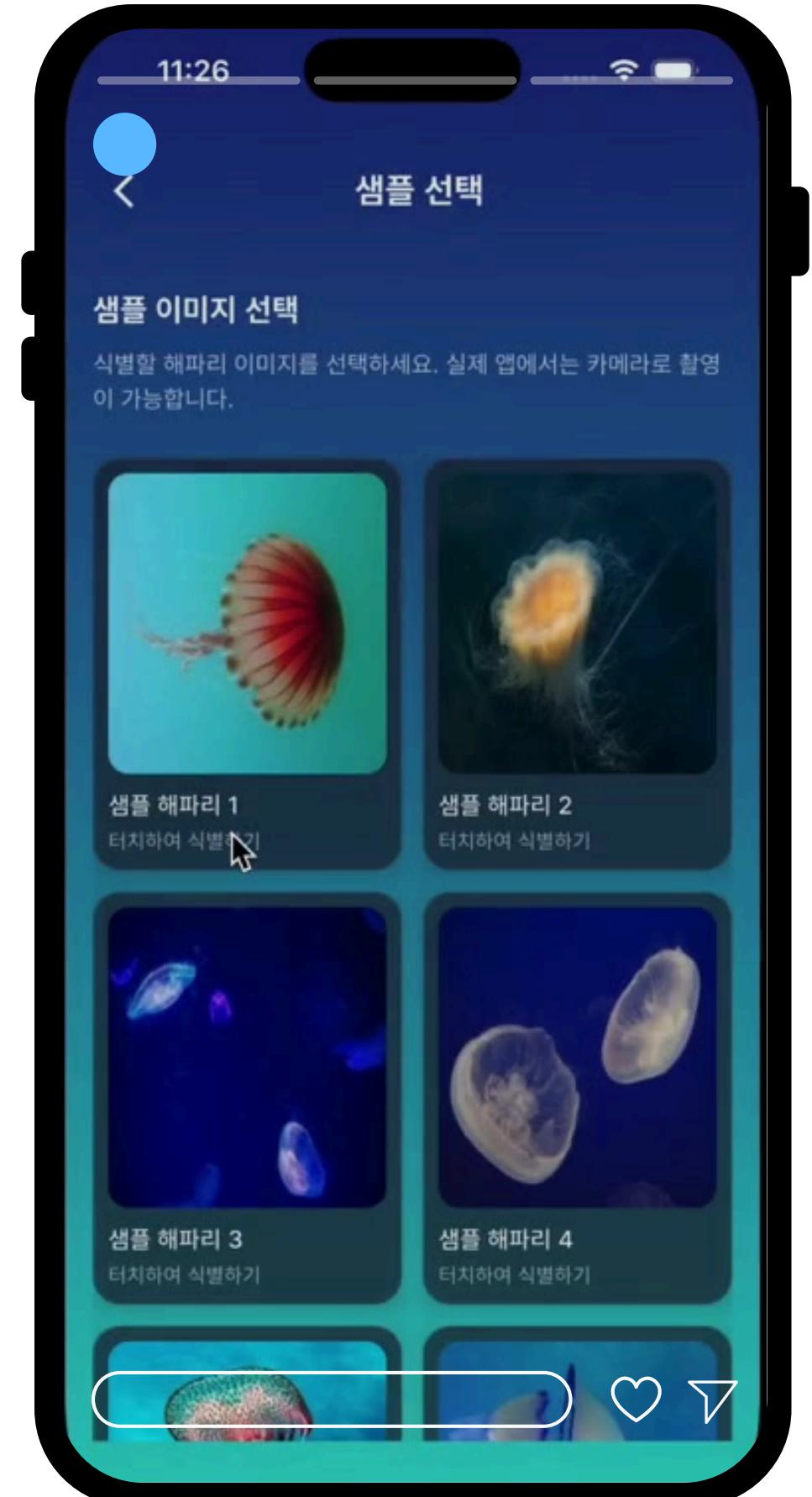
- 6종 해파리 실시간 분류 (Moon, Barrel, Blue, Compass, Lion's Mane, Mauve Stinger)
- 전이학습 기반 MobileNetV2 모델 (**평균 정확도 98%**)
- 양자화(TFLight)로 모바일 디바이스 최적화 (1.1mb)
- 5겹 교차검증으로 모든 데이터 셋에서 안정적인 성능 검증

⚠️ 안전 정보 제공

- 해파리별 위험도 표시 (저-중-고)
- 즉각적인 대처법 및 응급처치 정보
- 위치 제보를 통한 실시간 대응 강화

🎮 게이미피케이션

- 해파리 도감 수집 (6종 컬렉션)
- 레벨업 시스템 (초보 감식반 → 해파리 박사)
- 퀴즈와 돌발 이벤트로 지식 향상 및 참여 유도





MobileNetV2

- 경량 모바일 네트워크 아키텍처
- Inverted Residual 구조 도입
- 얇고 긴 계층 사용 (Thin and Long)
- 연산량과 모델 크기 최적화
- 모바일/엣지 디바이스에 최적화
- ImageNet 전이학습 가능
- 정확도와 계산 효율성 사이 우수한 균형

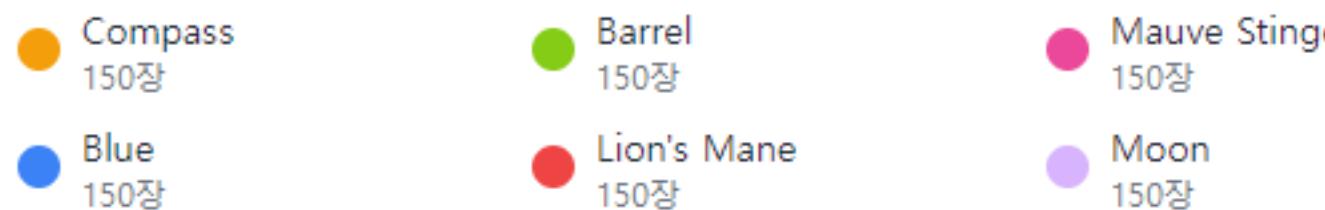
MobileNetV3Small

- V2 대비 더욱 경량화된 버전
- Neural Architecture Search (NAS) 기반 설계
- 하드웨어 실행 효율성 극대화
- Squeeze-and-Excitation 블록 통합
- 동적 계산 최적화
- 모바일 디바이스 전용 아키텍처
- 전력 소비 최소화
- 최소한의 성능 손실로 고효율 달성

EfficientNetB0

- 스케일링 기반 모델 아키텍처
- Compound Scaling 방법론 적용
- 깊이, 너비, 해상도 동시 최적화
- 모델 크기와 성능의 최적 균형
- AutoML 기반 아키텍처 설계
- ImageNet 성능 벤치마크 우수
- 다양한 크기의 EfficientNet 패밀리 존재
- 범용적 성능과 확장성

클래스별 비율



클래스별 비율



각 클래스 150장씩, 균등 분포

이미지 크기 불일치

이미지 크기 분포

● 224×224 ● 179×179

632장 268장



224x 224 이미지 리사이징

작은 데이터셋, 학습/평가 불균형

총 이미지 수: 979장

● Train ● Valid ● Test

900장 39장 40장



1. 데이터 증강 다변화

다양한 변환 기법을 적용하여 기존 이미지에서 새로운 학습 데이터를 생성

- 기하학적 변환: 회전, 뒤집기, 이동, 확대/축소
- 색상 변환: 밝기 조정, 대비 변화, 색조 변경
- 노이즈 추가: 수중환경 대응을 위한 노이즈 증가
- 필터 적용: 블러, 샤프닝
- 종별 특성 강화: 해파리 종별 특성강화 전략
- 혼합 증강: 여러 변환의 조합을 통한 다양성 증대

2. K-폴드 교차검증

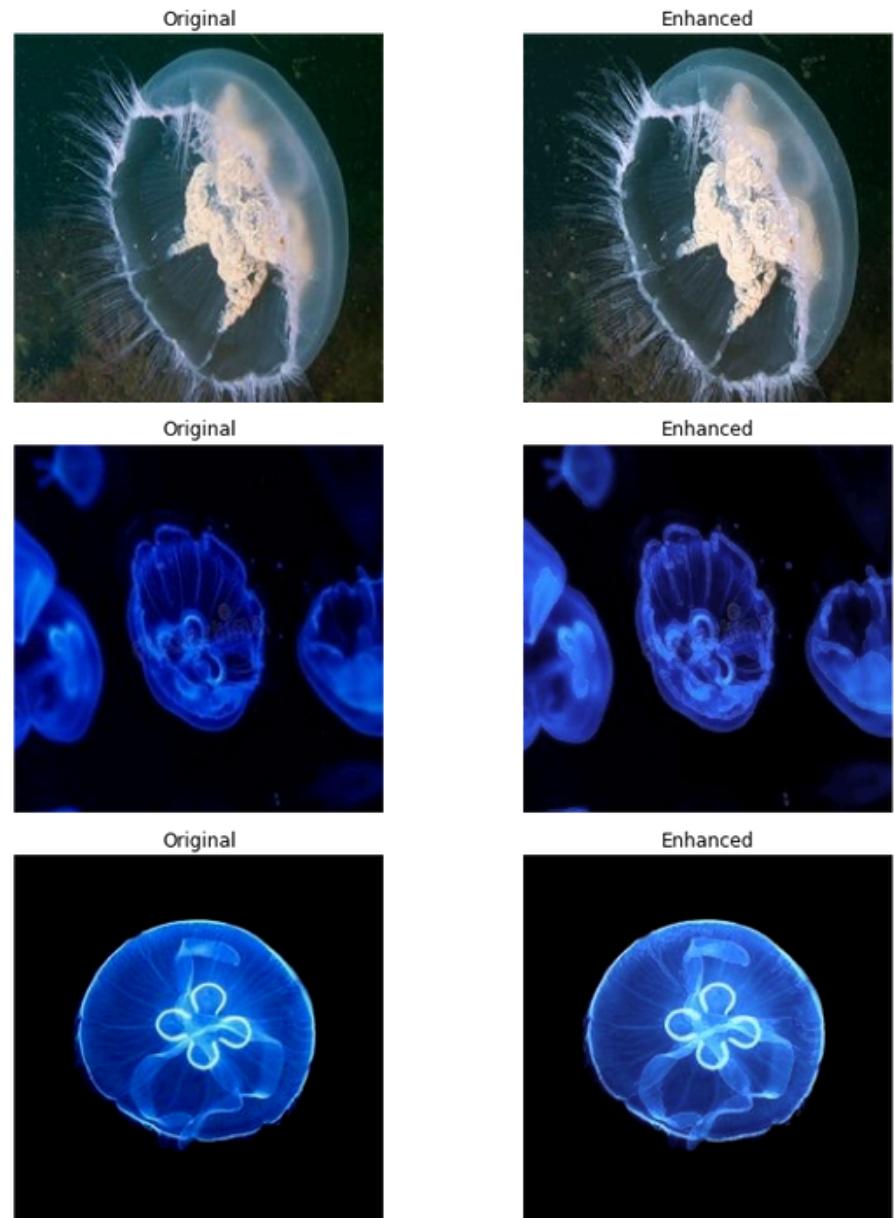
데이터 검증 신뢰성을 높이기 위해 5-폴드 교차검증을 구현

- 전체 데이터셋을 5개 부분으로 나눔
- 각 반복에서 4개 부분은 학습, 1개 부분은 검증에 사용
- 모든 데이터가 학습과 검증에 각각 한 번씩 사용됨
- 최종 모델 성능은 5개 모델의 평균 성능으로 평가

• 종별 특성강화

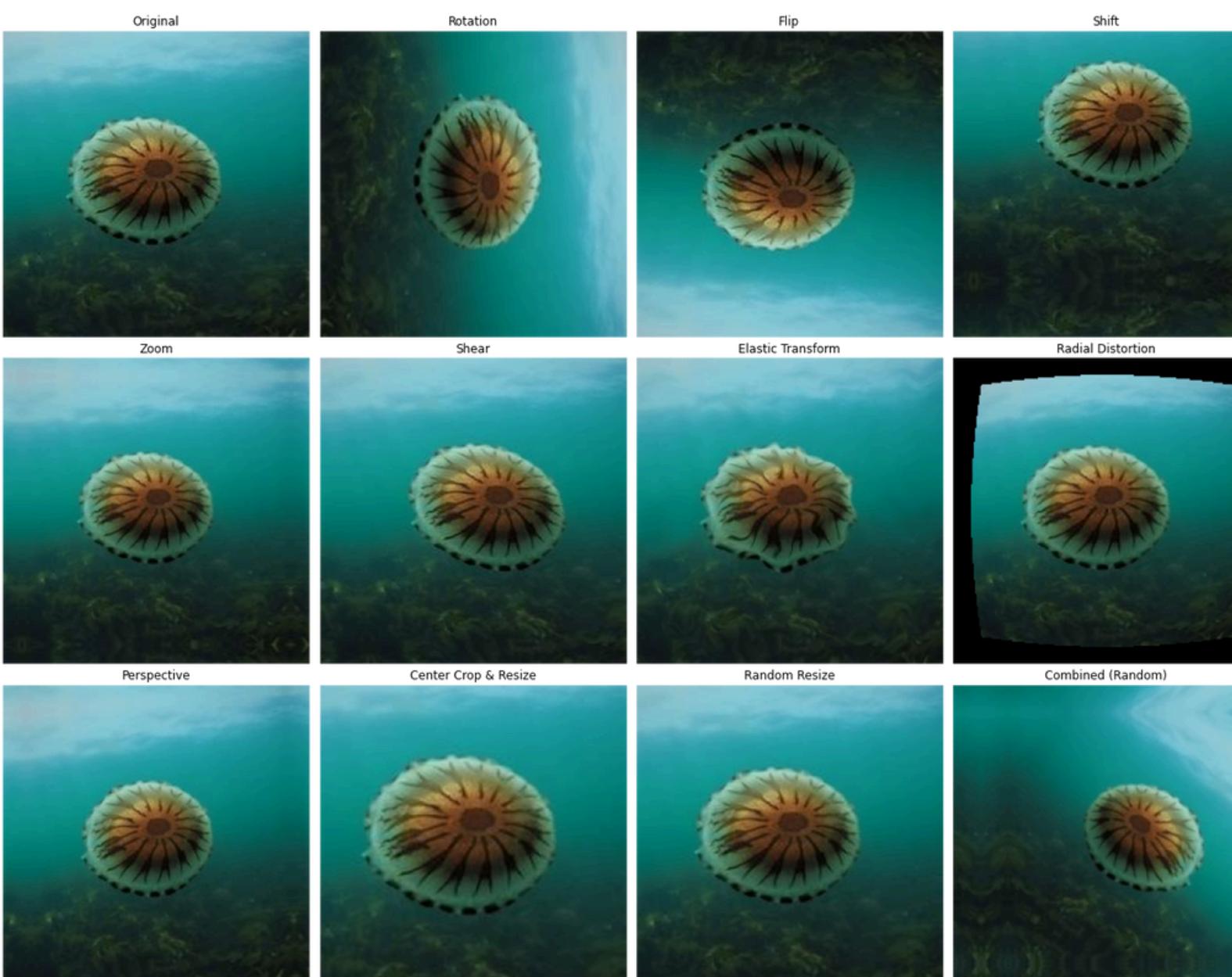
- Moon jellyfish: 투명도와 엣지 강조
- Barrel jellyfish: 대비와 선명도 증가
- Blue jellyfish: 파란색 채널 강조
- Compass jellyfish: 갈색 패턴 강조
- Lion's mane jellyfish: 크기와 붉은 색조 강조
- Mauve stinger: 엣지와 중간 밝기 강조

Moon jellyfish (*Aurelia aurita*) - Before / After



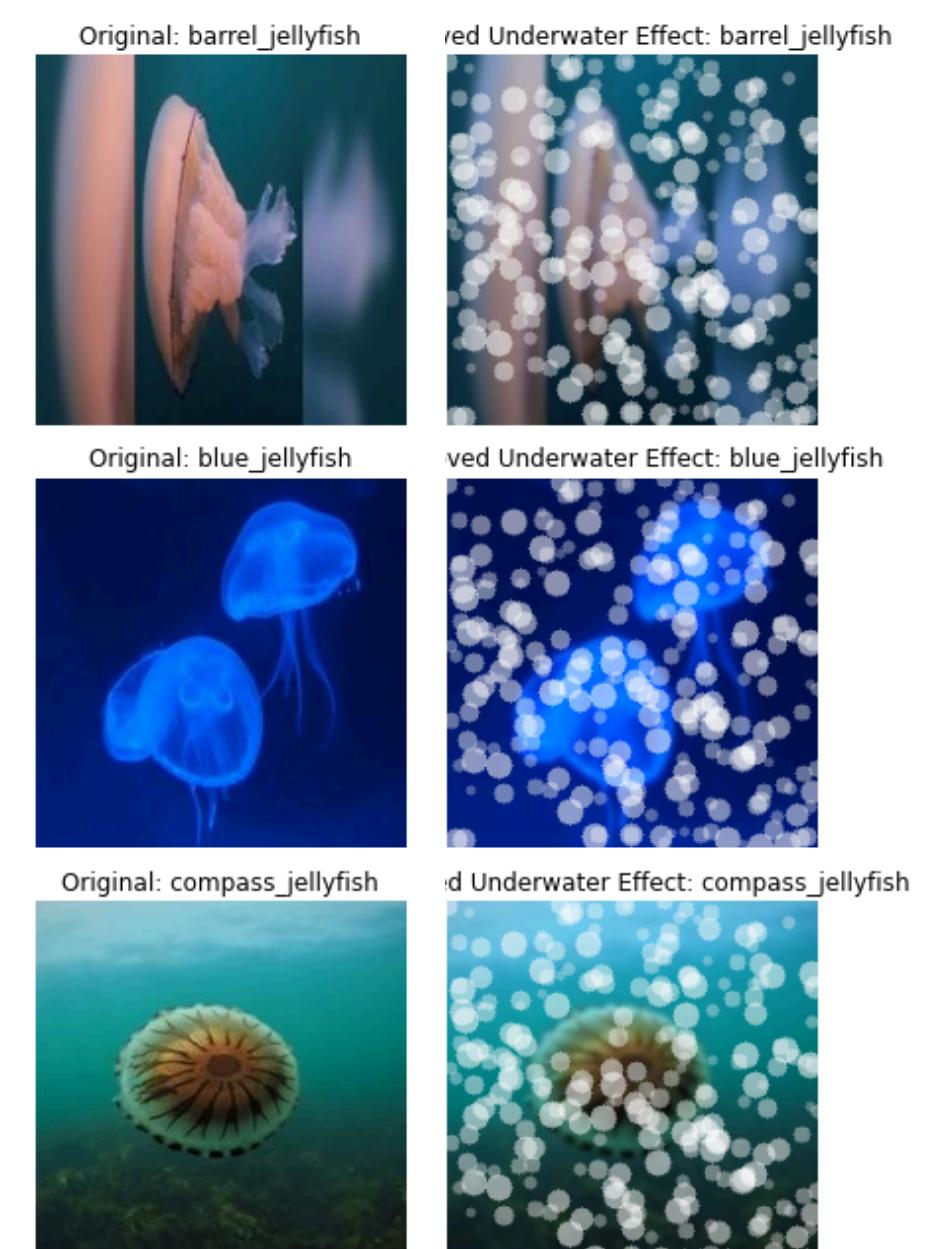
• 기하학적 증강

- 회전 (rotate)
- 이미지 뒤집기 (flip)
- 이동 (shift)
- 확대/축소 (zoom)
- 전단 변환 (shear)
- 탄성 변환 (elastic_transform)
- 방사형 왜곡 (radial_distortion)
- 원근 변환 (perspective_transform)
- 중앙 크롭 및 리사이즈 (center_crop_and_resize)
- 랜덤 리사이즈 (random_resize)



• 수중 환경 효과 시뮬레이션 전략

- 블루 틴트 적용 – 수중에서 빨간색 채널 감소
- 가우시안 블러
 - 수중 탁도(turbidity) 시뮬레이션
 - 이미지에 부드러운 흐림 효과 적용
- 버블 효과
 - 랜덤한 위치에 버블 생성
 - 다양한 크기와 투명도의 버블





TensorFlow Hub EfficientNet

- Train Acc 95.7%
- Val Acc: 76.9%
- Test Acc: 85.0%
- Test Loss: 0.426

양호한 정확도,
but, 낮은 검증 정확도

Keras EfficientNetB0

- Train Acc 98.3%
- Val Acc: 76.9%
- Test Acc: 95.0%
- Test Loss: 0.426

높은 테스트 정확도,
but, 낮은 검증 정확도

Keras / 전이학습 MobileNetV2

- Train Acc 52.9%
- Loss: 1.083

모바일환경 최적화 경량모델
전이학습으로 보수적 접근
(20%동결)

K폴드 교차검증 MobileNetV3 Small



MobileNetV3 Small

양상블모델 (M:5/E:5)

EfficientNetB0

양상블 (M:5/E:4/J:1)

과적합 징후

CNN (jellyfish 특화모델)

C 모델 개발위한 시도 기록

활성화 함수 / ReLU
손실함수 / categorical_crossentropy
옵티마이저 / Adam

		1차 (TensorFlow Hub/EfficientNet)	2차 (EfficientNetB0)	3차 (MobileNetV2)
기본 모델 아키텍처	베이스 모델	EfficientNet	EfficientNetB0	MobileNetV2
	가중치	명시x	ImageNet 가중치	ImageNet 가중치
	훈련가능성	명시x	전체 모델 훈련 가능	마지막 20개 레이어만 훈련 가능
	풀링방식	명시x	GlobalAveragePooling2D	GlobalAveragePooling2D
Dense 구조	온닉층	512 → 128 → 클래스 수	512 → 128 → 클래스 수	256 → 클래스 수
	규제화	BatchNorm + Dropout(0.2, 0.1)	BatchNorm + Dropout(0.4, 0.3) + L2(0.0001)	BatchNorm + Dropout(0.4)
하이퍼 파라미터	학습률률	0.001	0.0003	0.001
	드롭아웃비율	0.2, 0.1	0.4, 0.3	0.4
	l2규제	없음	0.0001	없음

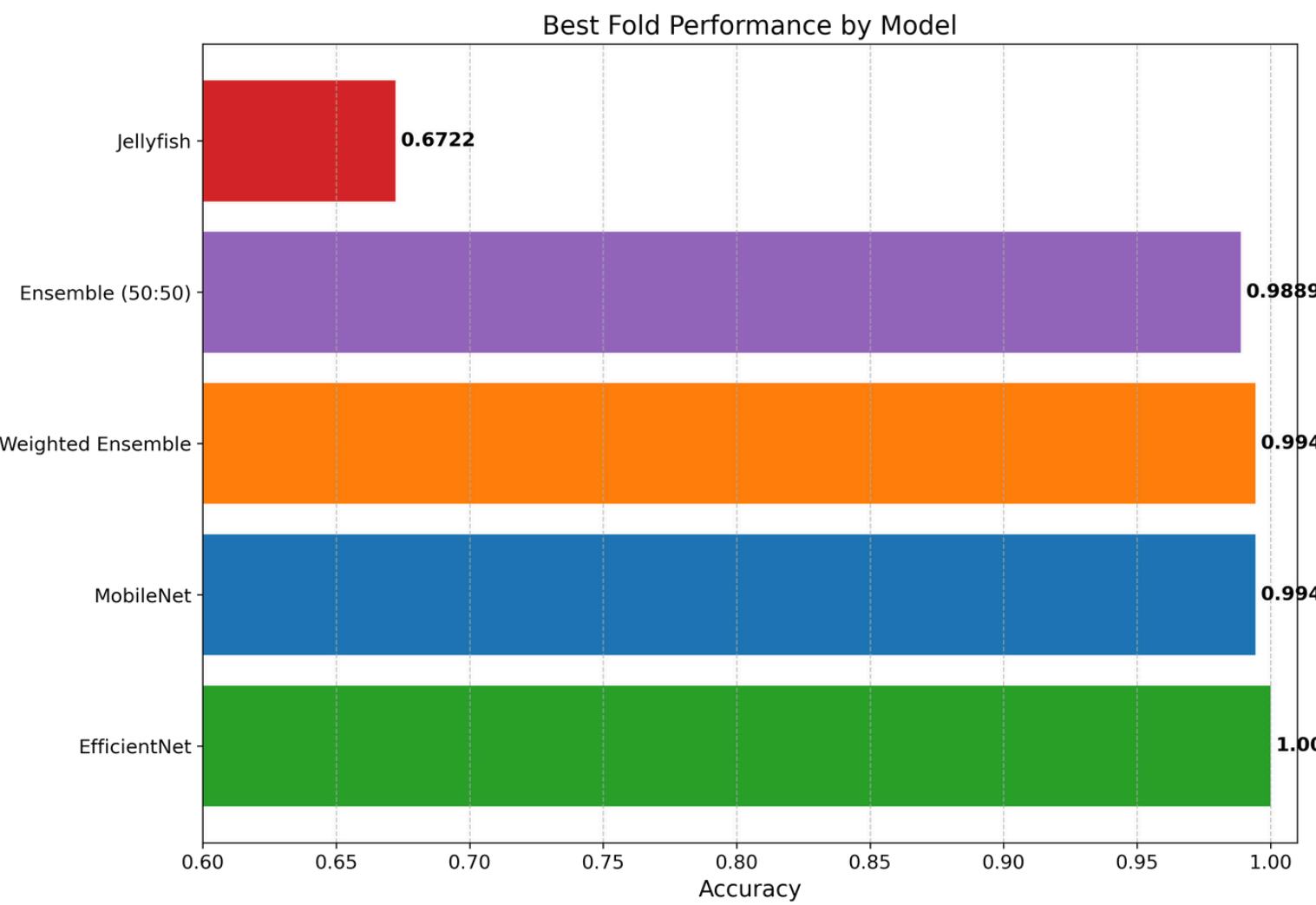


대부분 94% 이상의 높은 검증 정확도 달성

EfficientNet은 일부 fold에서 100% 성능을, MobileNet과 Weighted Ensemble은 최고 99.44%의 정확도를 달성

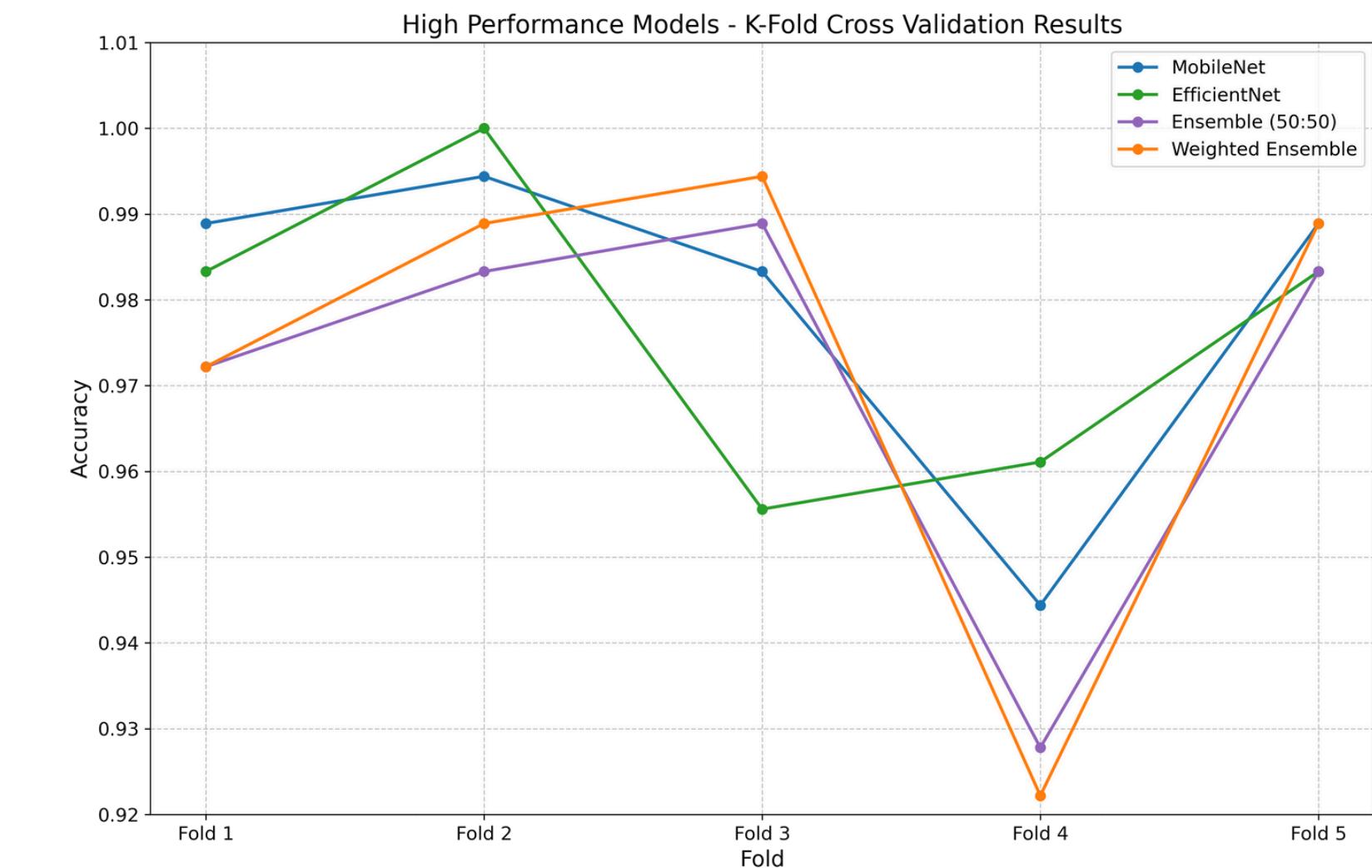
- 모델 성능 종합 평가

- 대부분의 모델이 94% 이상의 탁월한 정확도 달성
- EfficientNet은 일부 fold에서 완벽한 100% 정확도 기록
- Weighted Ensemble과 MobileNet은 최고 99.44% 정확도 시현
- Jellyfish 모델은 유일하게 67.22%의 상대적으로 낮은 정확도 기록



- 교차 검증(Fold) 성능 분석

- 각 fold별 미세한 성능 변동 관찰
- 2번 fold가 모든 모델에서 가장 안정적이고 일관된 고성능 입증
- 4번 fold는 상대적으로 더 큰 성능 편차 존재

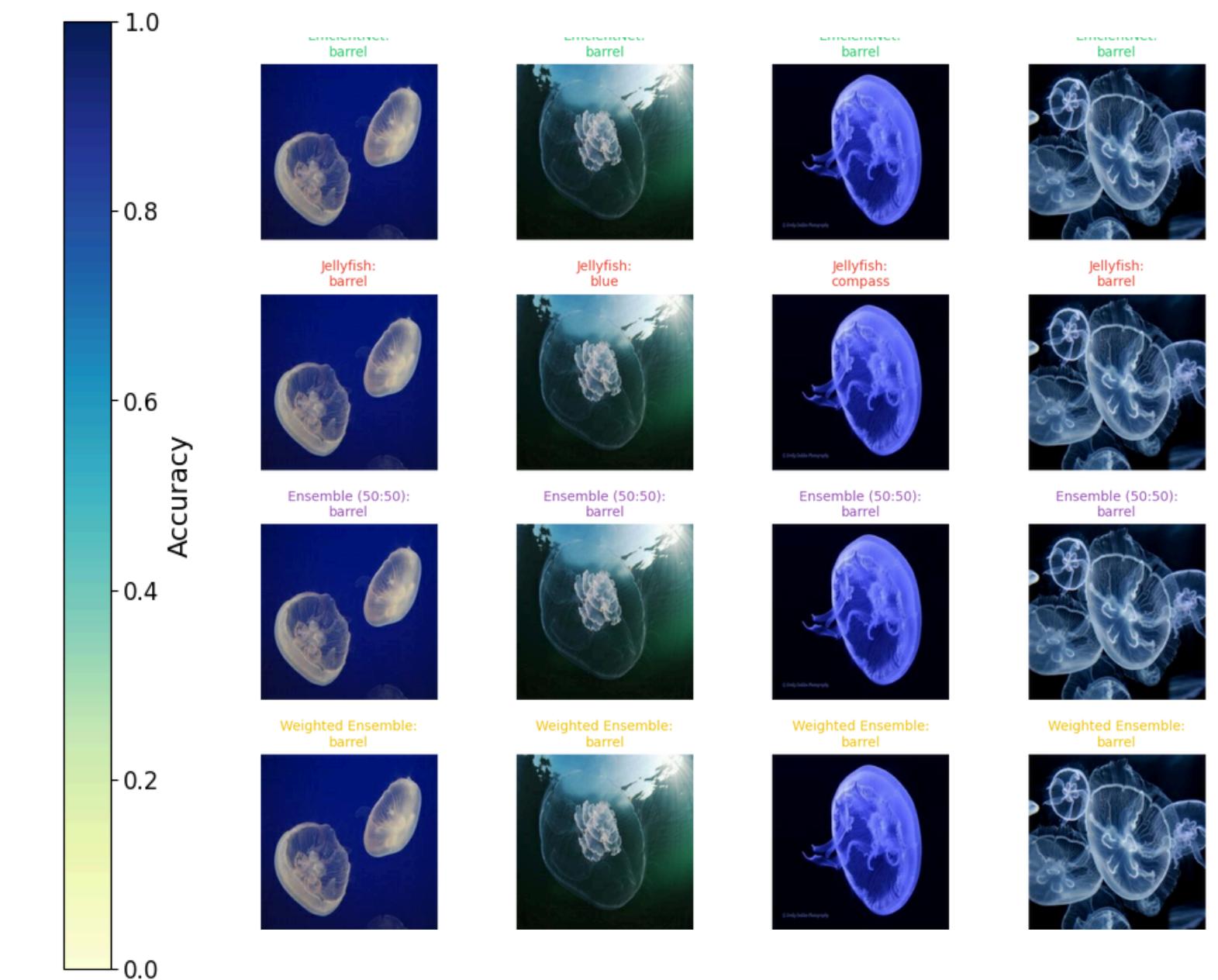
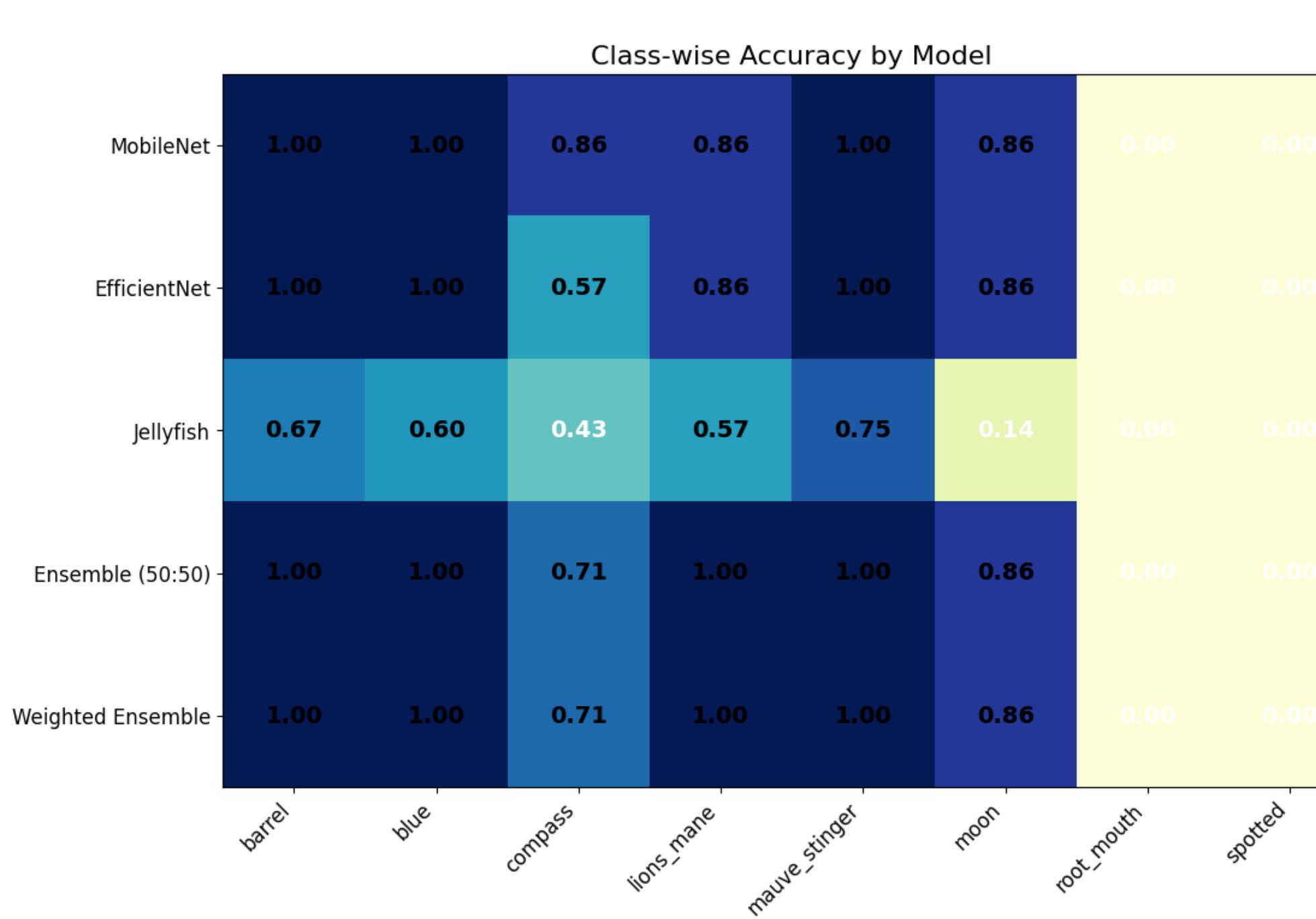


• 모델별 클래스 분류 강점

- MobileNet과 Weighted Ensemble: 4개 클래스에서 86-100% 안정적 성능
 - EfficientNet: 대부분 높은 정확도, 'compass' 클래스에서는 57%로 상대적 취약점 존재
 - Jellyfish 모델: 전반적으로 가장 낮은 정확도 (43-75% 범위)
 - Ensemble(50:50)과 Weighted Ensemble: 대부분의 클래스에서 일관되고 안정적인 성능 개별 클래스 분류의 불확실성을 효과적으로 감소

• 클래스별 도전 과제

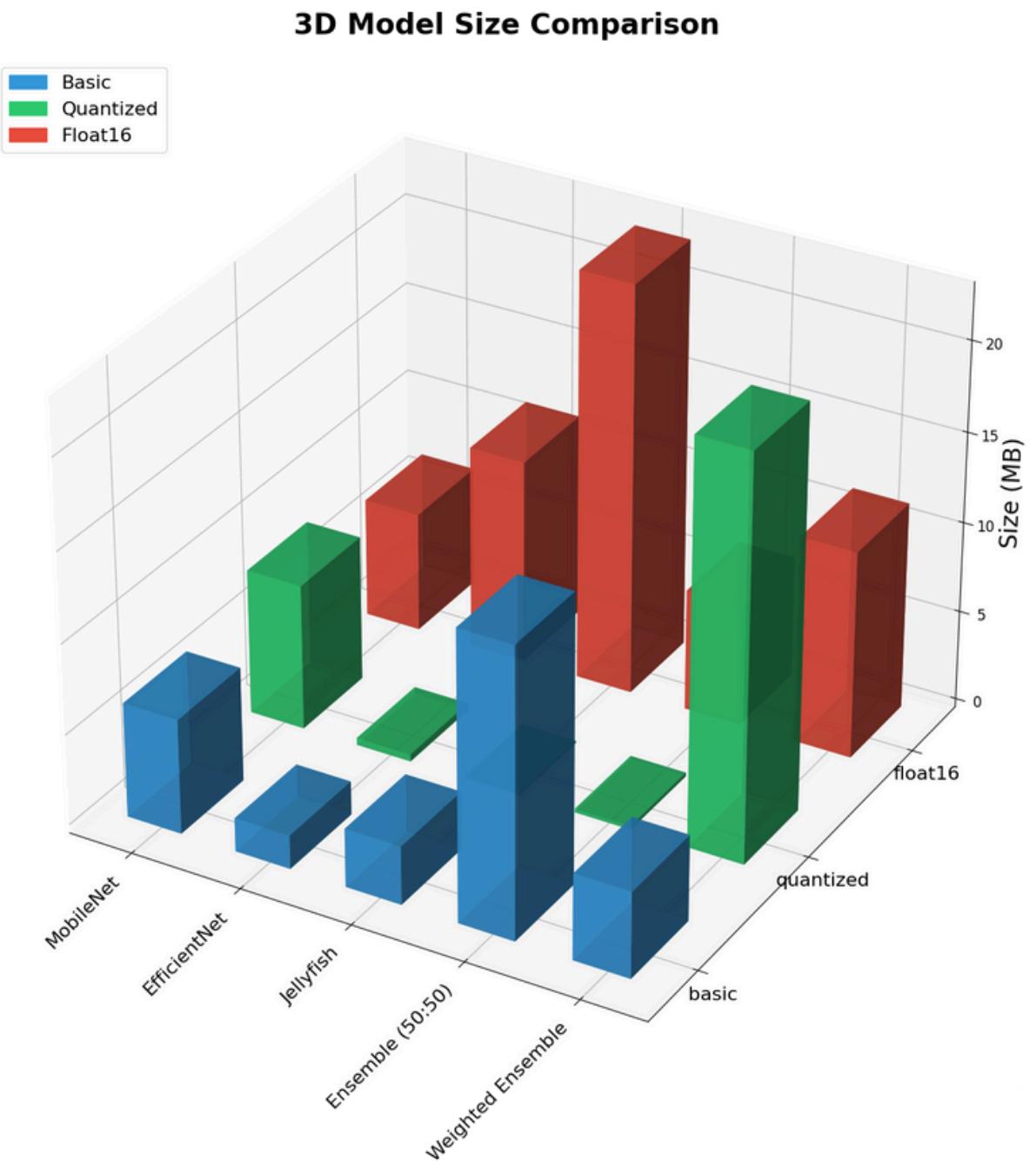
- root_mouth/stopped 클래스: 모든 모델에서 가장 낮은 분류 정확도
 - 추가 데이터 수집 또는 모델 개선 필요성 시사



모델크기가 작고, 안정성이 높은 모델 → MobilenetV3 전이학습 모델

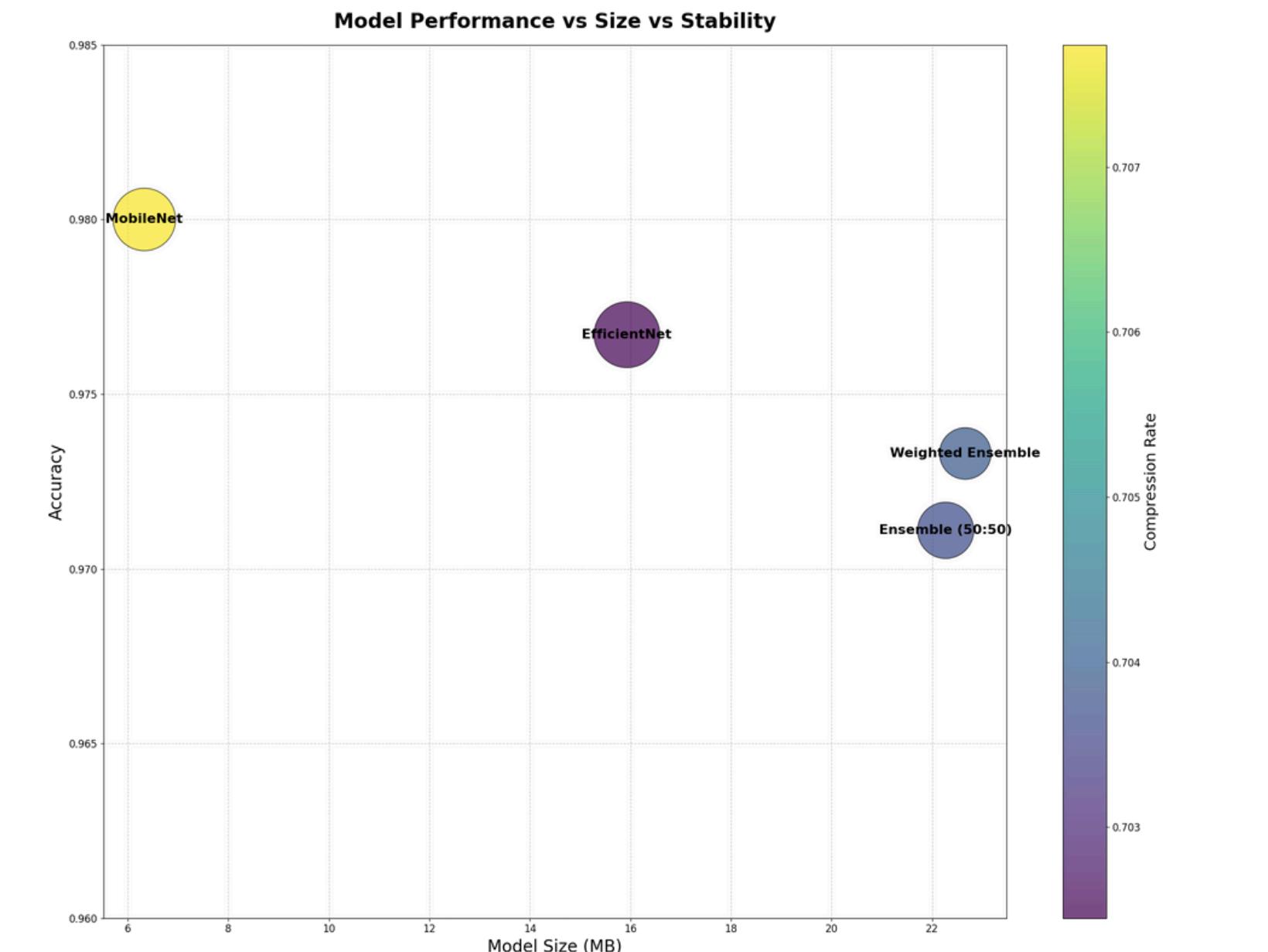
- 모델 크기 분석

- 모든 모델에서 Float16 버전이 가장 큰 모델 크기
- Weighted Ensemble이 가장 큰 모델 크기, MobileNet이 가장 작은 모델 크기
- Quantized 모델은 Basic 모델 대비 약 2배 크기
- Float16 모델은 Basic 모델 대비 약 3-4배 크기



- 모델 크기-성능-안정성 상관관계

- MobileNet: 모델 크기가 작으면서 최고 성능
- EfficientNet: 중간 규모 모델, 높은 성능과 안정성
- Weighted Ensemble: 대규모 모델, 우수한 일반화 능력
- Jellyfish 모델: 모델 크기 대비 현저히 낮은 성능



MobilenetV3 전이학습 모델 기반 서비스 개발

성능 측면

- MobileNet이 98.0%로 최고 성능 달성
- 상위 4개 모델(MobileNet, EfficientNet, 양상블 모델들)은 97% 이상 정확도 유지
- Jellyfish 모델은 64.3%로 현저히 낮은 성능 보임

안정성 관점

- MobileNet과 EfficientNet은 $\pm 1.6\text{~}1.8\%$ 의 매우 낮은 표준편차
- 이는 다양한 데이터셋에서도 일관된 성능을 의미
- 양상블 모델들은 $\pm 2.2\text{~}2.7\%$ 수준의 성능 편차

모델 경량화 관점

- MobileNet은 0.85MB로 최소 모델 크기
- 고성능 유지하면서 최소한의 모델 크기 확보
- 모바일/엣지 컴퓨팅 환경에 최적화

효율성 측면

- MobileNet이 53.0으로 최고 효율성
- 성능 대비 계산 복잡도가 가장 낮음
- 제한된 컴퓨팅 자원에서도 우수한 성능

종합 평가

- MobileNet: 성능, 안정성, 경량화 모든 측면 최적
- EfficientNet: 안정적이고 신뢰할 수 있는 대안
- 양상블 모델: 높은 일반화 능력
- Jellyfish 모델: 전면적 재설계 필요

