

DEEP LEARNING TEAM PROJECT

얼굴 감정 분류 AI 마스터 파이프라인

RAF-DB 데이터셋과 EfficientNet-B0를 활용한 고정밀 감정 인식 솔루션

 채환, 서현, 명수

 Version 1.0.4

CHAPTER 01

Introduction

프로젝트 배경, 목적 및 팀워크 구성을 통해 AI 감정 분류의 핵심 아젠다를 설정합니다.

WHY FACIAL EMOTION AI?

비언어적 데이터의 힘

인간 의사소통의 **55% 이상**은 시각적인 표정을 통해 전달됩니다. 이를 디지털화하는 것은 차세대 AI 상호작용의 필수 요소입니다.

- ✓ 교육: 학습자의 집중도 및 이해도 측정
- ✓ 의료: 비대면 상담 중 환자의 심리 상태 분석
- ✓ 보안: 이상 행동 및 위협 감정 조기 감지



RAF-DB 데이터셋 통계



Class Distribution (N=9,816)

- 행복 (Happiness): 38.9%
- 중립 (Neutral): 20.6%
- 슬픔 (Sadness): 16.1%
- 놀람 (Surprise): 10.5%
- 혐오 (Disgust): 5.8%
- 분노 (Anger): 5.7%
- 공포 (Fear): 2.4%

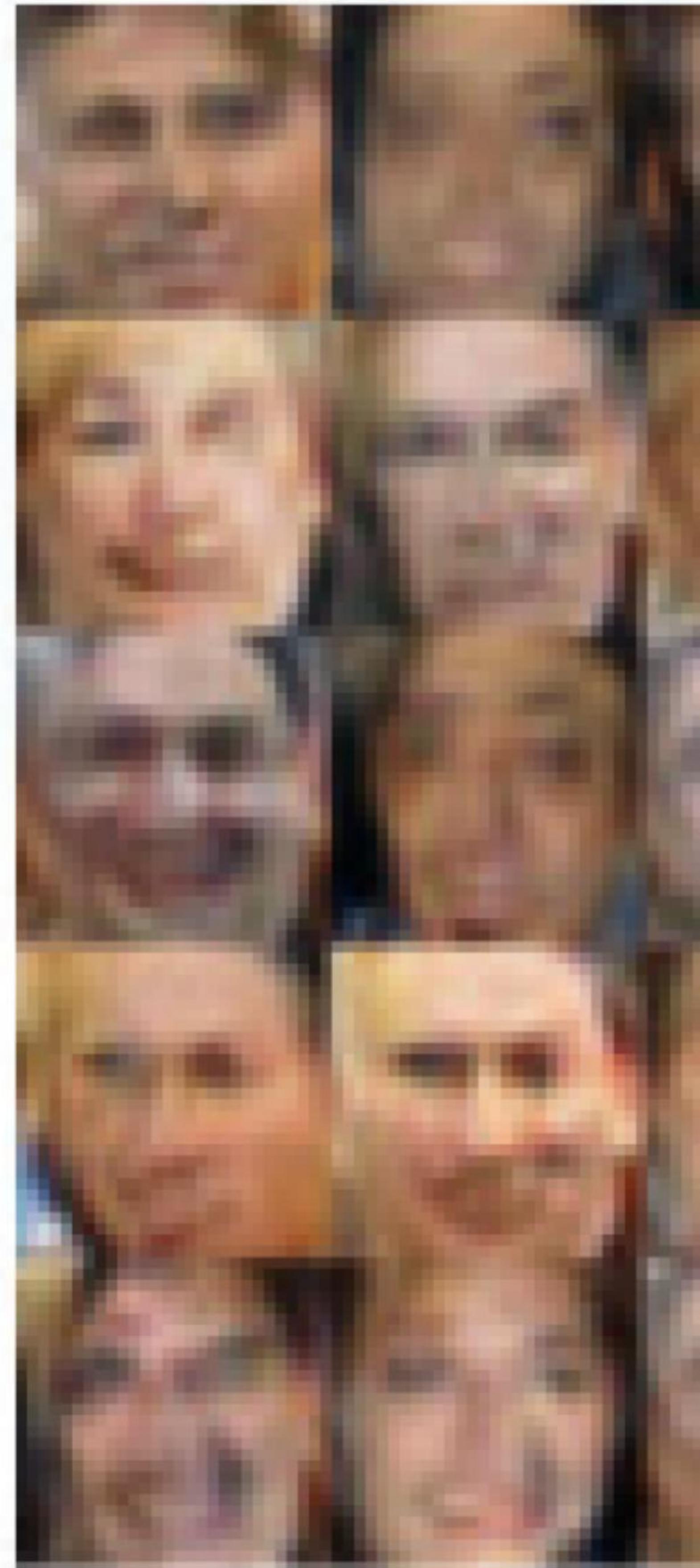
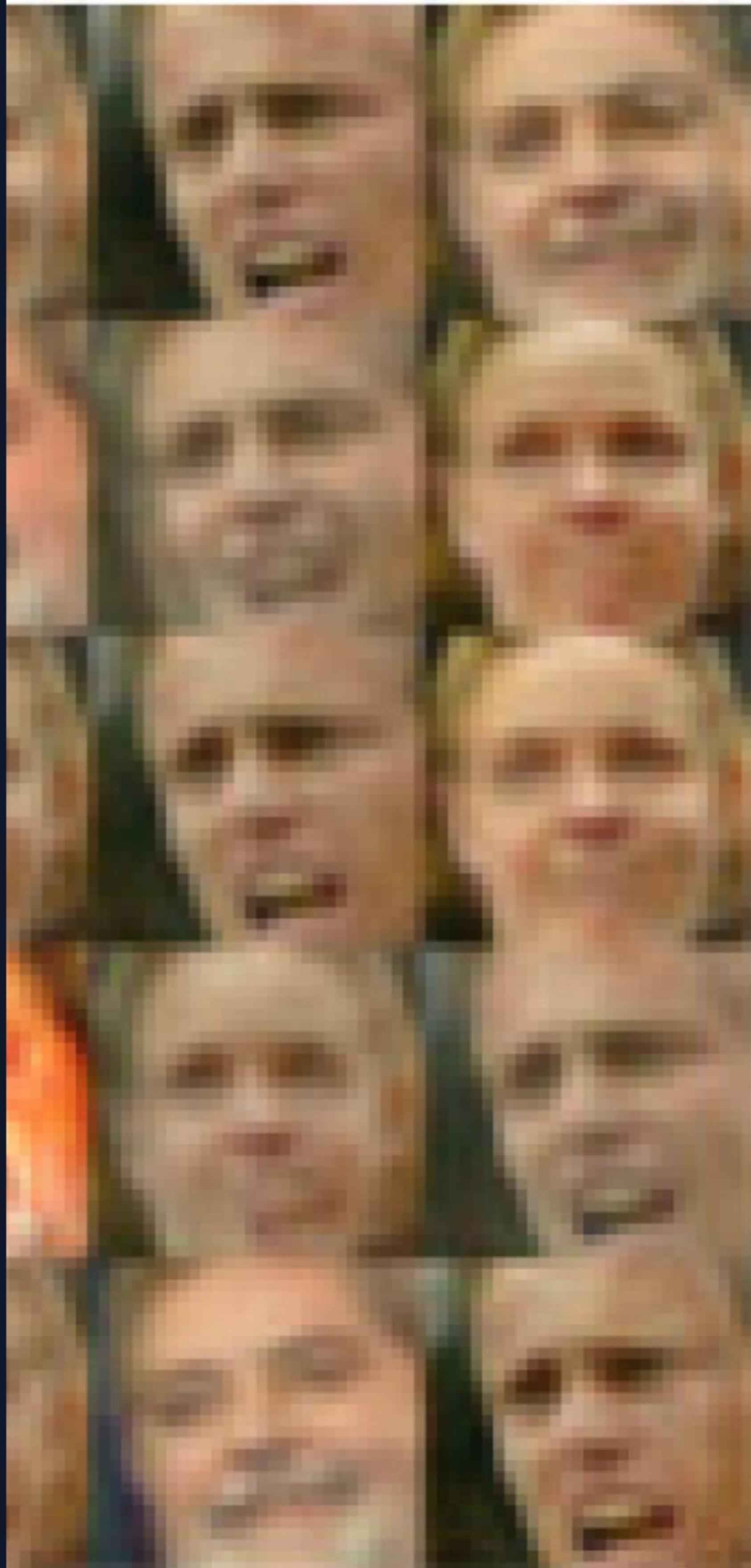
* 데이터 불균형 해결이 모델 성능의 핵심 과제

PART 01

Data Engineering

담당자: 채환

AI의 원재료인 데이터를 정제하고 가공하여 학습 최적
의 파이프라인을 구축합니다.



ALBUMENTATIONS 증강 전략

모델이 각도나 조명 변화에 흔들리지 않도록 **강건한 (Robust)** 데이터 증강 기법을 적용했습니다.

```
train_transform = A.Compose([
    A.HorizontalFlip(p=0.5),
    A.Resize(224, 224),
    A.Normalize(),
    ToTensorV2()
])
```

주요 증강 포인트

- **Horizontal Flip:** 표정의 방향성에 관계없이 동일 감정을 학습하도록 유도
- **Standardization:** ImageNet 통계 기반 정규화로 전이 학습 효율 극대화
- **Bilinear Resize:** 224x224 해상도 통일로 배치 처리 가속화

CUSTOM DATASET & LOADER

Dataset Architecture

OpenCV와 PyTorch Dataset 클래스를 결합하여 고성능 데
이터 로딩 로직을 구현했습니다.

- cv2.COLOR_BGR2RGB: 정확한 색상 채널 변환
- pin_memory=True: GPU 메모리 전송 속도 최
적화
- stratify_split: 라벨 비율을 유지하며 8:2
분할

16

OPTIMIZED BATCH SIZE

메모리 효율과 그래디언트 안정성을 고려한 최적의 배치 사
이즈 설정

PART 02

Model Development

담당자: 서현

최신 아키텍처를 기반으로 강력한 학습 엔진과 최적화
알고리즘을 설계합니다.

BACKBONE: EFFICIENTNET-B0

기존 CNN 대비 연산량은 줄이면서 성능을 비약적으로 높인
Compound Scaling 기술을 채택했습니다.

Feature Extraction 특징

마지막 풀링 레이어에서 추출된 **1280차원**의 풍부한 시각 특징을 활용하여 복잡한 표정 근육의 변화를 효과적으로 캡처합니다.

$$C = d^\alpha \cdot w^\beta \cdot r^\gamma$$

Compound Scaling Formula

Depth, Width, Resolution을 균형 있게 확장하여 감정 분류의 미세한 패턴 파악에 유리합니다.

OOP-BASED TRAINER CLASS

-  **Encapsulation:** 학습, 검증, 평가 로직을 단일 클래스로 캡슐화하여 코드 재사용성 극대화.
-  **Best Model Saving:** 매 에포크마다 val_acc를 모니터링하여 최적의 가중치 상태를 자동 보존.
-  **Tqdm Integration:** 학습 진행 상황을 실시간으로 시각화하여 모니터링 편의성 제공.

```
# Trainer.fit() Logic Flow
For epoch in range(epochs):
    train_loss, acc = train_epoch()
    val_loss, val_acc = validate_epoch()
    If val_acc > best_acc:
        self.best_model_state = model.state_dict()
```

학습 최적화 및 평가 지표



Loss Function

CrossEntropyLoss: 7개 클래스 간의 확률 분포 차이를 최소화하며 소프트맥스 연산을 포함하여 안정적인 그래디언트 생성.



LR Scheduler

ReduceLROnPlateau: 검증 손실이 3회 이상 정체될 경우 학습률을 50%로 자동 삭감하여 세밀한 전역 최적해 도달 유도.



Dual Metrics

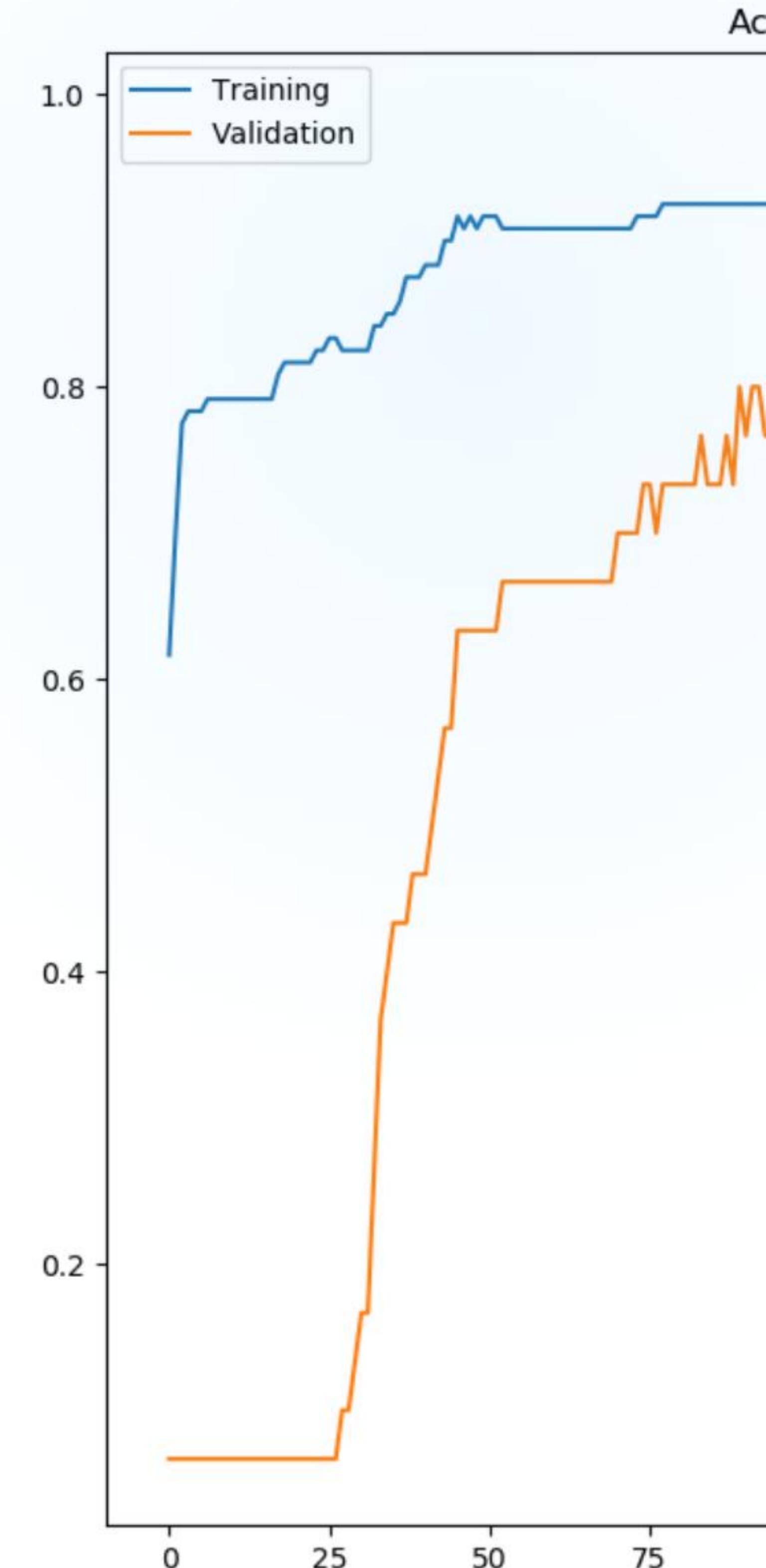
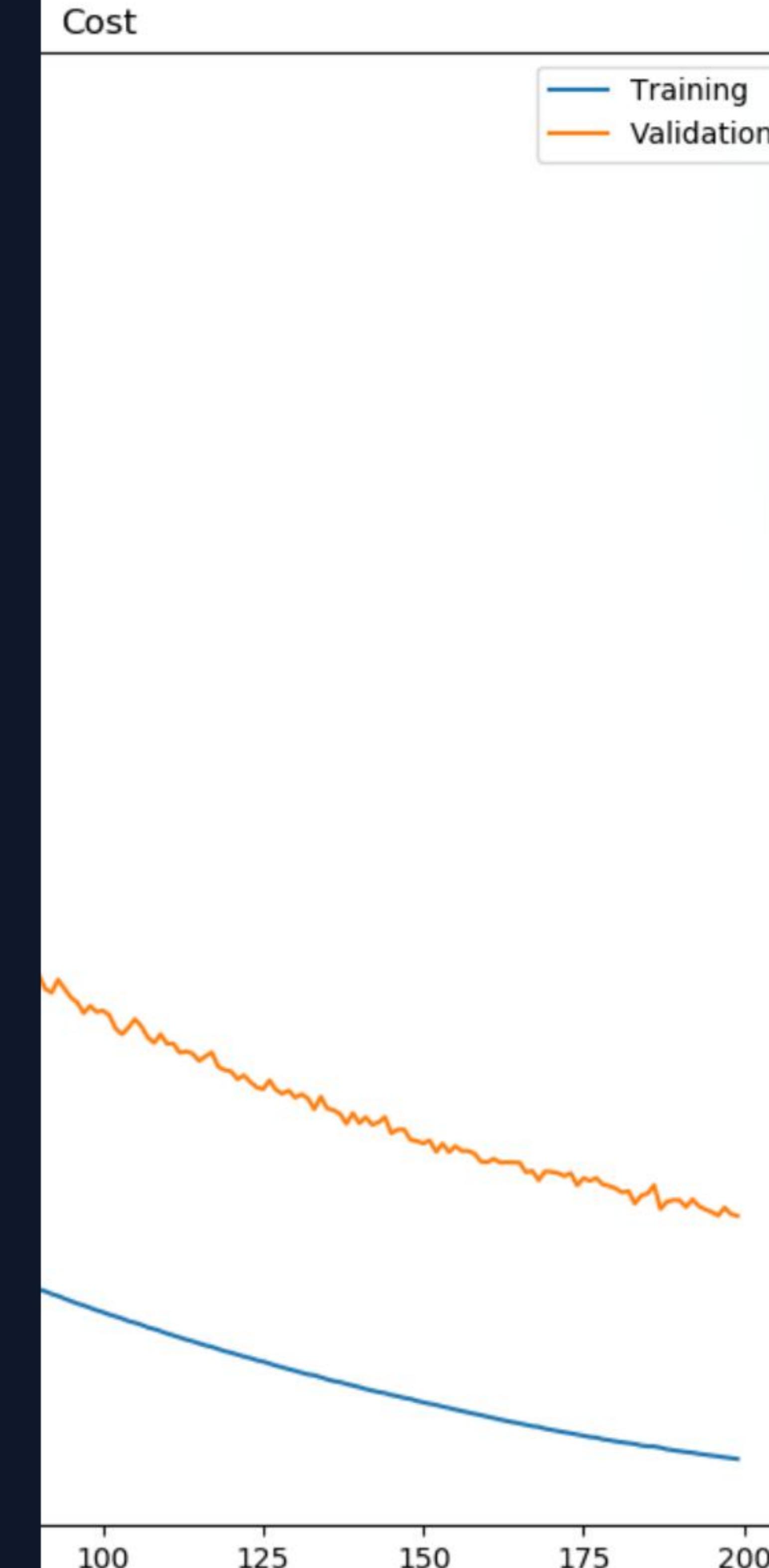
단순 정확도(Accuracy)뿐만 아니라 모델의 판별 능력을 측정하는 **AUROC** 지표를 병행 사용하여 성능을 입체적으로 평가.

PART 03

Prediction & Evaluation

담당자: 명수

학습된 모델을 실제 서비스에 적용하고 성능을 정량적으로 분석하여 신뢰성을 검증합니다.



EMOTIONPREDICTOR 클래스

End-to-End Inference

Raw 이미지를 입력받아 최종 감정 라벨까지 출력하는 통합 엔진을 구현했습니다.

```
def predict(self, image_path):
    img = self.transform(image=rgb_img)['image']
    logits = self.model(img.unsqueeze(0))
    probs = F.softmax(logits, dim=-1)
    return labels[argmax(probs)]
```

- ⚡ **CPU/GPU 유연성:** 하드웨어 환경을 자동 감지하여 최적의 추론 장치 할당.
- softmax 도출: 단순 점수가 아닌 0~1 사이의 확률 값으로 변환하여 신뢰도 제공.
- ▣ **Pre-processing 호환:** 학습 시와 동일한 Transform 필터 적용.

결과 시각화: 확률 분포 분석

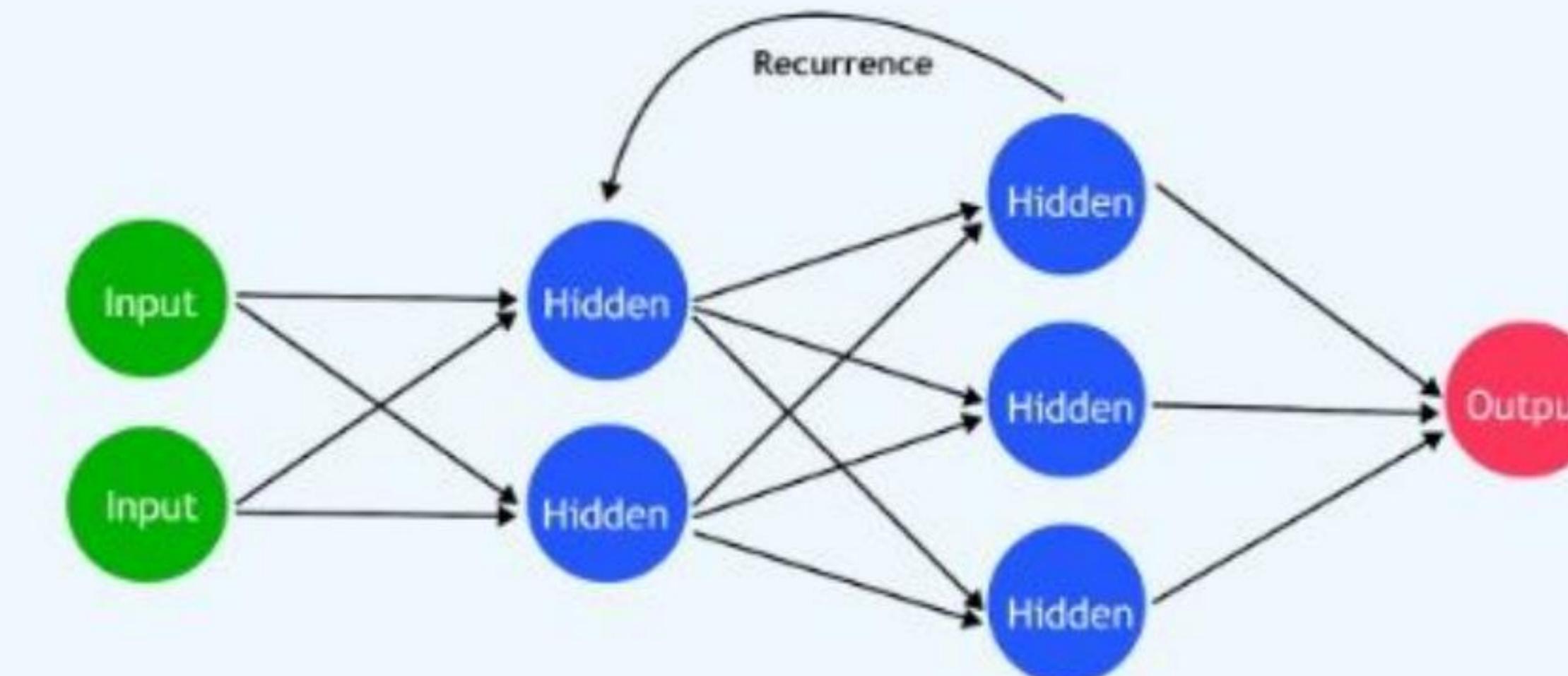
Horizontal Bar Chart (plt.barh)

단순히 하나의 감정만 맞추는 것이 아니라, 모델이 판단하는
감정 후보군들의 확률 분포를 한눈에 보여줍니다.

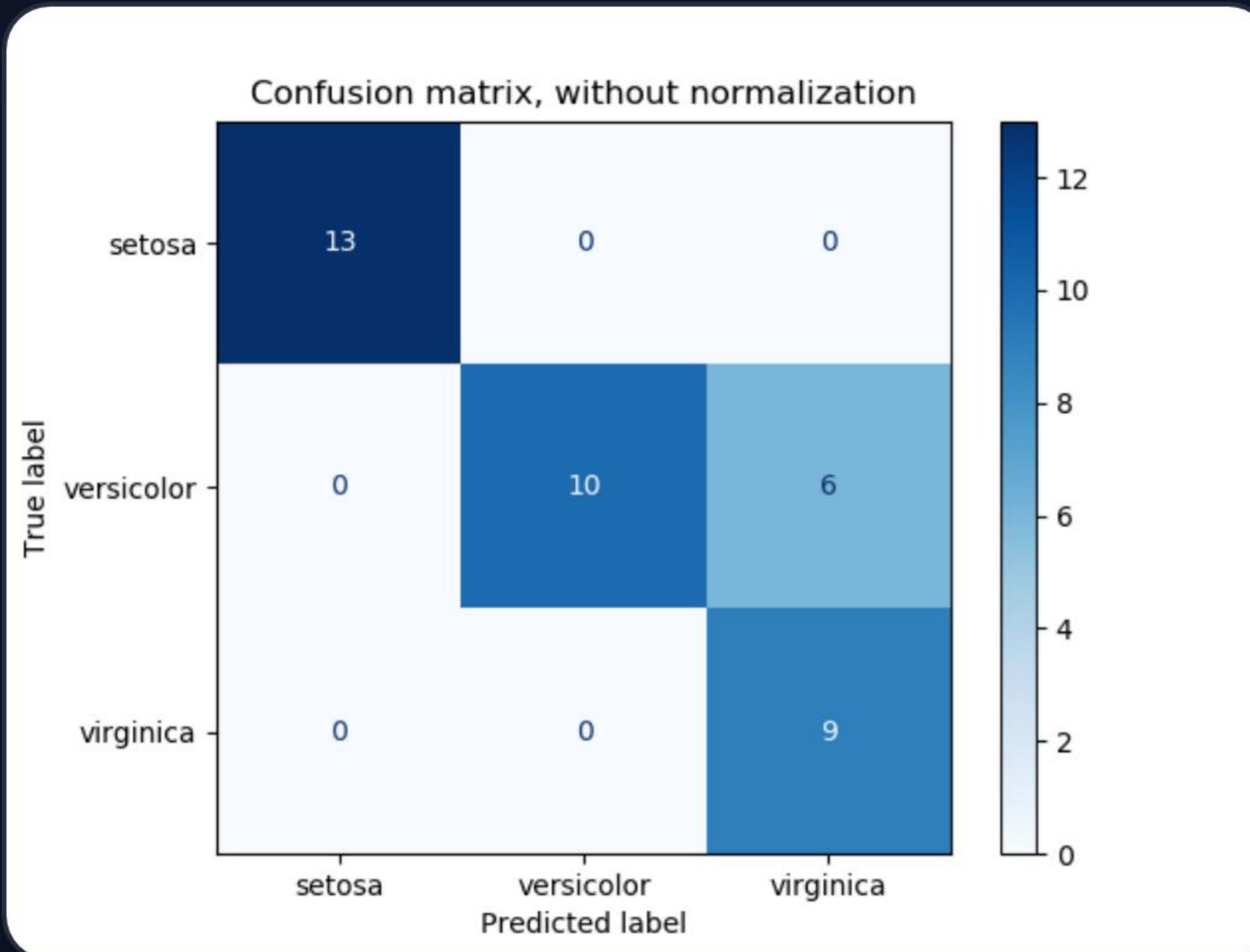
시각화 효과

상위 예측값은 강조 색상(Royal Blue)으로 표시하고, 나머지
감정은 회색으로 감쇄시켜 사용자 직관성을 극대화했습니다.

Recurrent Neural Networks



CONFUSION MATRIX & METRICS



정량적 평가 결과

Test Accuracy: 81.2%

- **최고 성능:** Happiness (F1 0.93) - 표정이 매우 뚜렷하여 모델이 완벽히 식별함.
- **개선 필요:** Fear / Disgust - 유사한 표정 근육 사용으로 인해 미세한 혼동 발생.
- **안정성:** Neutral / Sadness 등 미묘한 감정 변화도 80% 이상의 높은 재현율 기록.

결론 및 향후 발전 방향



성공적 구축

실제 환경 데이터를 극복한 고성능 감정 AI 구현



불균형 해소

Weighted Loss 적용으로 소수 클래스 보완 예정



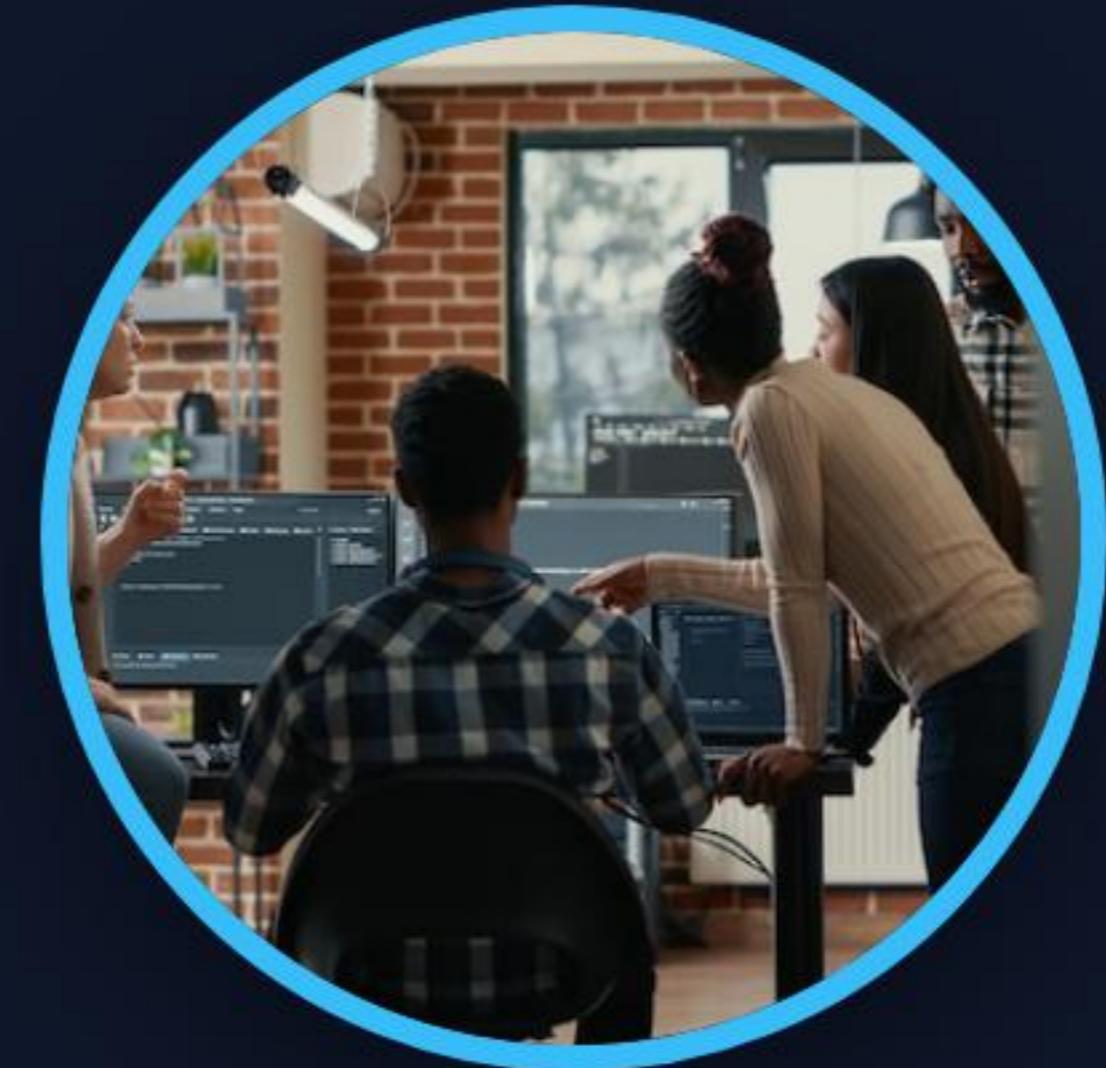
경량화

ONNX 변환을 통한 실시간 모바일 서비스 최적화



실시간 대응

프레임별 시계열 분석을 통한 감정 변화 추적



Q & A

발표를 경청해 주셔서 감사합니다.
질문이 있으시면 자유롭게 말씀해 주세요.

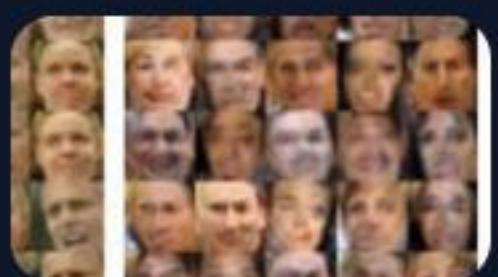
Contact: ai_project_team@example.com

IMAGE SOURCES



https://miro.medium.com/v2/resize:fit:700/1*UghBE0Itc0NhXkbbMPKMRA.jpeg

Source: www.comet.com



https://www.mdpi.com/information/information-13-00370/article_deploy/html/images/information-13-00370-g004.png

Source: www.mdpi.com



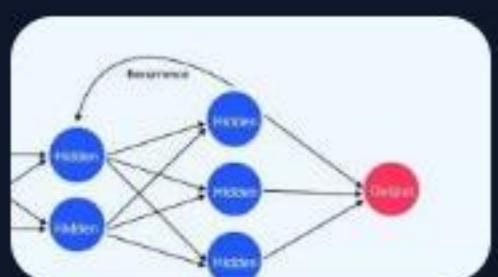
<https://viso.ai/wp-content/uploads/2024/03/EfficientNet-Architecture-diagram.png>

Source: viso.ai



<https://i.sstatic.net/Y9bj3.png>

Source: stackoverflow.com



https://ik.imagekit.io/upgrad1/abroad-images/imageCompo/images/recurrent_neural_networkX0WI24.jpg

Source: www.upgrad.com



<https://i.sstatic.net/g7s9f.png>

Source: stackoverflow.com

IMAGE SOURCES



https://img.freepik.com/free-photo/coders-discussing-about-source-code-compiling-discovers-errors-asks-rest-team-explanations-front-multiple-screens-running-algorithms-software-developers-doing-teamwork_482257-41856.jpg

Source: www.freepik.com