

# 프로젝트 경력 기술서

## 📌 프로젝트명

피부암 Binary Classification AI 시스템

## 📄 프로젝트 설명

ResNet50 기반 Transfer Learning을 활용하여 피부경(dermatoscope) 이미지를 분석하고 피부 병변의 양성/악성을 자동 분류할 수 있는 딥러닝 AI 시스템 개발. 의료 접근성이 낮은 지역에서 조기 스크리닝 도구로 활용 가능하며, 의료진의 진단 보조 시스템으로 실제 배포 가능한 웹 애플리케이션을 구현.

## 👛 주요 업무

### 1. 데이터 전처리 및 분석

- HAM10000 데이터셋(10,015장) 분석 및 Binary 분류 체계 설계
- 클래스 불균형 문제 해결: 80:20 → 1:1 완전 균형 샘플링 (3,908장)
- OpenCV 기반 전처리 파이프라인 구축 (CLAHE, ImageNet Normalization)
- Data Augmentation 기법 적용 (Flip, Rotation, Zoom, Brightness)
- Train/Test Split 후 정규화 적용으로 데이터 누수 방지

### 2. 딥러닝 모델 개발

- ResNet50 기반 Transfer Learning 아키텍처 설계 및 구현
- 2단계 학습 전략 수립: Freeze Learning → Fine-tuning
- Custom Classification Head 설계 (Dense 256/128 + Dropout 0.5/0.3)
- Adam Optimizer, Binary Crossentropy Loss 최적화
- Kaggle GPU 환경에서 모델 학습 및 하이퍼파라미터 튜닝

### 3. 성능 최적화

- 목표 성능 초과 달성: Accuracy 77.62%, Recall 81.84%, AUC 0.8585
- 의료 AI 특성 고려한 Recall 우선 전략 채택 (악성 종양 검출력 강화)
- 전처리 순서 오류 해결 (EfficientNet → ImageNet Normalization)
- Learning Rate Scheduling 및 Dropout을 통한 과적합 방지

### 4. 웹 애플리케이션 개발 및 배포

- Streamlit 기반 실시간 예측 웹 애플리케이션 개발

- 사용자 친화적 UI/UX 설계 (이미지 업로드, 예측 결과 시각화, CSV 내보내기)
- 다중 이미지 동시 예측 및 통계 대시보드 구현
- TensorFlow 2.17 모델 서빙 및 전처리 파이프라인 일관성 유지

## 5. 시스템 이식성 및 유지보수

- 다중 PC 환경 배포 문제 해결 (Python 버전, TensorFlow 호환성, Streamlit 버전 충돌)
- Conda 가상환경 기반 패키지 의존성 관리
- 상세한 설치 가이드 문서화 (QUICK\_START, SETUP\_GUIDE)
- GitHub Releases를 통한 모델 버전 관리

---

## 🏆 주요 성과

### 정량적 성과

- **Accuracy 77.62%** 달성 (목표 75% 초과)
- **Recall 81.84%** 달성 (악성 종양 검출력 우수)
- **AUC 0.8585** 달성 (우수한 분류 성능)
- 클래스 불균형 데이터를 완전 균형(1:1) 샘플링으로 개선
- 3,908장 데이터로 24M+ 파라미터 모델 학습 성공

### 기술적 성과

- **End-to-End AI 시스템 구현:** 데이터 수집 → 전처리 → 모델 개발 → 웹 배포까지 전 과정 수행
- **Transfer Learning 실무 적용:** ImageNet 사전학습 모델 활용 및 의료 이미징 도메인 전이
- **실패 경험 극복:** 전처리 오류, 클래스 불균형, 세션 타임아웃 등 다양한 문제 해결
- **실제 배포 가능한 시스템:** 로컬 및 다중 환경에서 실행 가능한 이식성 확보

### 문서화 및 재현성

- 상세한 프로젝트 기획서 작성 (MD, PDF)
- 설치 및 실행 가이드 문서화
- requirements.txt를 통한 의존성 명시
- GitHub 코드 및 모델 관리

---

## 🔧 개선 사항

### 문제 인식 및 해결 과정

#### 1. 전처리 파이프라인 오류 해결

- **문제:** EfficientNet preprocessing 함수 사용 시 값 범위 이상 (-1~1 초과)으로 학습 실패

- **원인:** Train/Test Split 전 잘못된 정규화 적용
- **해결:** ImageNet Normalization으로 변경 및 Split 후 정규화 적용
- **학습:** 전처리 순서의 중요성 및 데이터 누수 방지 원칙 체득

## 2. 클래스 불균형 문제 해결

- **문제:** 원본 데이터 양성 80.5%, 악성 19.5%로 심한 불균형
- **시도:** Focal Loss, Class Weight, SMOTE, Oversampling 등 다양한 기법 실험
- **해결:** 완전한 1:1 샘플링으로 근본적 해결 (각 1,954장)
- **학습:** 도메인 특성에 맞는 데이터 전략의 중요성

## 3. 모델 선정 과정 개선

- **문제:** 초기 EfficientNetB0 사용 시 전처리 호환성 문제 발생
- **해결:** 안정적이고 검증된 ResNet50으로 변경
- **학습:** 의료 이미징 분야에서 널리 사용되는 아키텍처 선택의 중요성

## 4. 버전 호환성 관리

- **문제:** TensorFlow 2.17로 저장한 모델이 2.15 환경에서 로드 불가
- **에러:** `Could not deserialize class 'Functional'` 발생
- **해결:** 정확한 버전 명시 (requirements.txt) 및 compile=False 옵션 활용
- **학습:** ML 프로젝트의 의존성 관리 및 버전 일치의 중요성

## 5. 학습 환경 제약 극복

- **문제:** Kaggle 세션 타임아웃(12시간)으로 학습 중단 및 모델 손실
- **해결:** 주기적 체크포인트 저장 및 학습 재개 전략 수립
- **학습:** 클라우드 환경의 제약사항 이해 및 대응 전략

---

## 💡 핵심 역량

### 딥러닝 기술 역량

- Transfer Learning 실무 적용 능력
- 데이터 전처리 및 증강 기법
- 모델 최적화 및 하이퍼파라미터 튜닝
- 의료 AI 특수성 이해 (Recall 우선 전략)

### 소프트웨어 개발 역량

- Python 기반 End-to-End 시스템 개발

- Streamlit 웹 애플리케이션 개발
- 버전 관리 및 의존성 관리 (Git, Conda)
- 크로스 플랫폼 이식성 확보

### 문제 해결 역량

- 체계적인 디버깅 및 문제 원인 분석
- 다양한 실패 경험을 통한 학습
- 대안 탐색 및 최적 솔루션 선택
- 문서화를 통한 지식 공유

### 프로젝트 관리 역량

- 명확한 목표 설정 및 단계별 진행
- 상세한 문서화 및 재현 가능성 확보
- 실무 적용을 고려한 설계
- End-user 관점의 UX 고려

---

## 기술 스택

### Deep Learning

- TensorFlow 2.17, Keras
- ResNet50, Transfer Learning
- Adam Optimizer, Binary Crossentropy

### Data Processing

- OpenCV 4.8.0
- NumPy 1.24.3, Pandas 2.0.3
- PIL/Pillow, CLAHE

### Web Development

- Streamlit 1.28.0
- Python 3.11

### Development Tools

- Kaggle Notebooks (GPU: NVIDIA P100)
- Conda, Git, GitHub
- Jupyter Notebook, VS Code

- HAM10000 (10,015 images)
  - 7-class → Binary classification
- 

## 프로젝트 기간 및 규모

- 기간: 2024년 11월 (약 1개월)
  - 역할: 개인 프로젝트 (데이터 분석, 모델 개발, 웹 배포 전 과정 수행)
  - 데이터: 3,908장 (학습 데이터),  $224 \times 224 \times 3$  RGB 이미지
  - 모델: 24M+ parameters, 94MB
  - 코드: 약 500줄 (전처리, 학습, 웹 앱)
- 

## 프로젝트를 통한 성장

### 기술적 성장

- 의료 AI 도메인 지식 및 실무 경험 습득
- 전체 ML 파이프라인 구축 능력 배양
- Transfer Learning의 실전 적용 노하우 체득
- 모델 배포 및 서빙 경험

### 소프트 스킬 성장

- 문제 정의부터 솔루션 배포까지 전 과정 경험
  - 실패를 통한 학습 및 개선 능력
  - 기술 문서 작성 및 커뮤니케이션 능력
  - 사용자 관점의 제품 개발 마인드
- 

## 관련 링크

- GitHub: <https://github.com/parkahjin/skin-cancer-classification>
  - 프로젝트 기획서: [PDF/MD 파일]
  - 데모 영상: [Streamlit 앱 실행 화면]
  - 발표 자료: [PPT/PDF]
- 

본 프로젝트는 AI 플랫폼 기업 취업을 위한 포트폴리오로 제작되었으며, 실제 의료 현장 적용을 위해서는 임상 검증 및 규제 승인이 필요합니다.