

# 포트폴리오

## 1. RAG를 활용한 멀티모달 효과적 비디오 검색 및 상호작용

Intel과 협력한 DeepLearning.AI의 Multimodal RAG: Chat with Videos 과정의 일환으로, 비디오 콘텐츠와 지능적으로 쿼리하고 상호작용할 수 있는 멀티모달 검색 증강 생성(RAG) 시스템을 개발했습니다. 이 시스템은 비디오 프레임 추출, 전사, 멀티모달 임베딩 및 검색을 포함한 멀티모달 AI 기술을 활용합니다.

### 기술적 전략:

- 비디오 전처리: 세 가지 다른 비디오 입력 유형(전사본이 있는 비디오, 오디오는 있지만 전사본이 없는 비디오, 오디오나 전사본이 없는 비디오)을 처리하는 비디오 전처리 스크립트를 개발했습니다. 이는 비디오 프레임 추출 및 OpenAI의 Whisper와 LLaVA 모델을 사용한 전사 생성을 포함합니다.
- 벡터 저장소 수집: BridgeTower 임베딩을 사용하여 LanceDB 벡터 데이터베이스로의 수집을 구현했습니다. 검색에 사용되는 데이터의 품질을 향상시키기 위해 조각화된 전사본을 보강하는 실험을 했습니다.
- RAG 시스템 구현: LangChain을 사용하여 멀티모달 검색 증강 생성(RAG) 시스템을 만들어 비디오 기반 쿼리 및 상호작용을 가능하게 했습니다. Python과 Gradio를 사용하여 텍스트 기반 및 웹 인터페이스 쿼리 시스템을 구현했습니다.

### 프로젝트 결과물:

- 비디오 프레임 추출: 비디오에서 프레임을 추출하고 전사본 및 비디오 세그먼트 정보와 같은 메타데이터와 연결하는 시스템을 개발했습니다.
- 멀티모달 데이터 수집: 비디오 및 전사 데이터를 처리하고 벡터 저장소에 수집하여 검색 가능하게 만들었습니다.
- RAG 시스템: 사용자가 비디오 프레임, 전사본 및 임베딩의 조합을 사용하여 비디오 콘텐츠에 대한 멀티모달 쿼리를 할 수 있도록 했습니다.

### 사용 도구:

- LangChain: 멀티모달 RAG 시스템 실행
- OpenAI Whisper: 전사용
- MoviePy: 비디오에서 오디오 추출
- LLaVA 모델: 비디오 프레임에서 캡션 생성
- LanceDB: 벡터 저장

### 사용된 API 키:

- OpenAI API: 자연어 처리용
- Prediction Guard API: 모델 예측용

### ML 알고리즘:

- BridgeTower 임베딩: 전사본 벡터화
- RAG 시스템: 비디오 데이터 쿼리

 [수료증 링크](#)

 [깃허브 링크](#)

## 2. LLaMA 고객 지원 어시스턴트: 파인튜닝된 대화형 AI

M1 아키텍처를 위한 Unsloth 최적화를 통해 LLaMA 3.2 모델을 파인튜닝하여 구축된 전문 고객 지원 AI 시스템입니다. 이 시스템은 상황적 이해와 구조화된 응답으로 고객 서비스 쿼리를 처리합니다.

기술적 전략:

- 2배 빠른 훈련을 위해 Unsloth를 사용하여 LLaMA 3.2 베이스 모델에 LoRA 파인튜닝을 구현했습니다.
- 그래디언트 체크포인팅과 효율적인 배치 처리를 통해 M1 Mac용 메모리 사용량을 최적화했습니다.
- 구조화된 고객 지원 대화를 위한 데이터셋 전처리 파이프라인을 구성했습니다.

프로젝트 기능:

- 고객 문의에 대한 자동 응답 생성
- M1 아키텍처에서 메모리 효율적인 훈련
- 주문 관련 쿼리의 상황적 이해
- 성능 시각화 및 메트릭 추적
- 빠른 응답을 위한 간소화된 추론 파이프라인

핵심 구성요소:

- LoRA 어댑터가 있는 파인튜닝된 LLaMA 3.2 모델
- 고객 서비스 컨텍스트를 위한 사용자 정의 데이터 처리
- Apple Silicon을 위한 최적화된 훈련 파이프라인
- 모델 평가를 위한 포괄적인 테스트 스위트

도구 및 기술: PyTorch, Transformers, Unsloth, Python 3.10, PEFT, Datasets, Accelerate, Apple M1 MPS 백엔드

데이터셋: 주문 관리, 취소 및 상태 문의를 다루는 Bitext의 27K 고객 지원 대화

결과:

- Unsloth 최적화를 통해 2배 빠른 훈련 달성
- 8GB RAM 미만의 효율적인 메모리 사용
- 고객 쿼리에 대한 자연스러운 응답 생성

 [깃허브 링크](#)

## 3. 어텐션 메커니즘을 활용한 신경망 대화형 챗봇

PyTorch에서 시퀀스-투-시퀀스 신경망과 어텐션 메커니즘을 사용하여 고급 챗봇을 개발했습니다. 챗봇은 영화 대화에서 220,000개 이상의 대화 쌍을 포함하는 Cornell Movie-Dialogs Corpus에서 훈련되었습니다.

기술적 전략:

- 양방향 GRU 인코더 + 어텐션 GRU 디코더

- Luong의 글로벌 어텐션 메커니즘
- 배치 처리, 500 hidden units, 2 레이어, dropout 0.1

도구 및 기술: PyTorch, Python, NLTK, NumPy, Pandas

 [깃허브 링크](#)

## 4. 다중 에이전트 아키텍처 기반 LLM 여행 추천 시스템

부산의 여행자를 위한 최적화된 일일 일정과 상황별 추천을 생성하기 위해 LangChain과 여러 AI 에이전트를 활용한 고급 여행 계획 시스템입니다. 시스템은 날씨 데이터와 관광지 정보를 처리하여 개인화된 여행 경험을 만듭니다.

기술적 전략:

- LangChain을 활용한 이중 에이전트 아키텍처:
- 일정 생성기: 관광지 방문 시간과 날씨 조건을 기반으로 일일 여행 일정을 최적화합니다.
- 추천 생성기: 상황별 여행 조언과 날씨별 주의사항을 제공합니다.
- 구조화된 관광지 및 날씨 데이터 처리를 위한 견고한 JSON 처리 파이프라인

핵심 기능:

- 온도, 공기질 및 조건에 맞춰 적응하는 날씨 인식 계획
- 최적의 여행 경험을 위한 동적 일정 조정
- 포괄적인 오류 처리 및 입력 검증
- 확장 가능한 기능을 위한 모듈식 아키텍처

도구 및 기술:

- Python 3.7+, PyTorch, LangChain, Ollama
- 지능적인 처리를 위한 LLaMA 3.2 모델
- 구조화된 데이터 처리를 위한 JSON

결과:

- 환경 인식을 통한 효율적인 일정 생성
- 복잡한 일정 시나리오의 견고한 처리

 [깃허브 링크](#)

## 5. PyTorch 2 마스터링

Mastering PyTorch 2 책을 통해 기초부터 고급 아키텍처까지 딥러닝 개념을 탐구했습니다. 제 학습에는 DenseNet, GoogLeNet, ResNet과 같은 CNN 아키텍처 구현, 추천 시스템 및 텍스트

생성 모델 구축, 트랜스포머 기반 애플리케이션을 위한 Hugging Face 통합이 포함되었습니다. 또한 Captum을 사용한 모델 해석가능성 작업도 했습니다. 주요 구현에는 MNIST 숫자 분류, AlexNet을 사용한 전이 학습, LSTM/RNN 애플리케이션 및 음악/텍스트 생성 모델이 포함되었습니다.

 [깃허브 링크](#)

## 6. 딥러닝과 전통적인 ML을 사용한 감정 분석

이 프로젝트는 현대적인 딥러닝 접근법과 전통적인 머신러닝 방법을 모두 사용하여 포괄적인 감정 분석을 구현합니다. 시스템은 IMDB 데이터셋의 영화 리뷰를 분석하여 감정을 긍정적 또는 부정적으로 분류하며, 정교한 LSTM 기반 RNN 아키텍처와 전통적인 분류기를 활용합니다.

기술적 전략:

- LSTM-RNN 아키텍처: 향상된 텍스트 이해를 위해 사용자 정의 가능한 레이어, 단어 임베딩, 드롭아웃 정규화 및 어텐션 메커니즘을 갖춘 양방향 LSTM을 사용하여 정교한 신경망을 구축했습니다.
- 전통적인 ML 통합: 비교 분석을 위해 구성 가능한 매개변수를 가진 다항 나이브 베이즈와 SGD 분류기를 구현하고 최적화했습니다.

프로젝트 기능:

- 다중 모델 감정 분석: 양방향 기능을 가진 딥러닝 LSTM-RNN과 전통적인 ML 접근법
- 성능 최적화: 패킹된 패딩 시퀀스와 드롭아웃 정규화를 통한 효율적인 시퀀스 처리
- 비교 분석: 다양한 모델 성능에 대한 포괄적인 평가 및 비교
- 대화형 인터페이스: 쉬운 실험과 결과 시각화를 위한 Jupyter 노트북 인터페이스

사용 도구: PyTorch, Scikit-learn, Python, NLTK, NumPy, Pandas, Matplotlib, Seaborn

ML 알고리즘: 양방향 LSTM 네트워크, 다항 나이브 베이즈, 확률적 경사 하강법, 단어 임베딩

 [깃허브 링크](#)

## 7. Stable Diffusion 이미지 생성

정밀한 제어를 통한 텍스트 가이드 이미지 변환

텍스트 가이드 이미지 생성 및 변환을 위한 Stable Diffusion 트랜스포머를 실험하고, 세밀한 제어 메커니즘과 효율적인 처리를 위한 GPU 가속을 구현했습니다.

사용 도구:

- Python 3.7+
- PyTorch
- Hugging Face Diffusers
- CUDA GPU 가속
- PIL (Python Imaging Library)
- Stable Diffusion v1-4 모델

 [깃허브 링크](#)

## 8. 배터리 수명 예측

이 프로젝트의 주요 목표는 다양한 회귀 알고리즘의 적용을 통해 배터리의 잔여 유용 수명(RUL)을 예측하는 것이었습니다. 주요 초점은 높은 정확도를 달성하는 것이었으며, RandomForestRegressor가 인상적인 99%를 달성했습니다. 이 프로젝트를 위해 Kaggle에서 얻은 데이터셋을 사용했습니다. 이 데이터셋은 25°C에서 1000회 이상 사이클된 14개의 NMC-LCO 18650 배터리로 구성되어 있으며, 방전 시간, 전압 동작 및 충전 시간을 특징으로 했습니다. 이 프로젝트는 탐색적 데이터 분석(EDA), 데이터 전처리, 모델 선택 및 평가, 특성 중요도 분석, 앙상블 모델링 단계를 포함합니다. 효과적인 데이터 시각화를 위해 Matplotlib과 Seaborn을 활용하여 통찰력 있는 시각화를 만들었습니다. 전반적으로 이 프로젝트는 배터리 RUL에 대한 예측 모델을 성공적으로 개발하여 특성 중요도에 대한 귀중한 통찰을 제공하고 RandomForestRegressor로 놀라운 정확도를 달성했습니다.

예측에 사용된 특성

배터리 RUL 예측을 위한 예측 특성은 다음과 같습니다:

- F1: 방전 시간(초)
- F2: 4.15V에서의 시간(초)
- F3: 정전류 시간(초)
- F4: 3.6-3.4V 감소(초)
- F5: 최대 방전 전압(V)
- F6: 최소 충전 전압(V)
- F7: 충전 시간(초)
- 총 시간(초)

가장 중요한 특성

모델에 따르면 RUL 예측에 가장 중요한 특성은 다음과 같습니다:

1. F1: 방전 시간(초)
2. F4: 3.6-3.4V 감소(초)

머신러닝 알고리즘

Scikit-learn을 사용하여 다양한 회귀 알고리즘을 구현했습니다:

- RandomForestRegressor
- AdaBoostRegressor
- GradientBoostingRegressor
- BaggingRegressor
- SVR

- DecisionTreeRegressor
- ExtraTreeRegressor
- LinearRegression
- SGDRegressor
- KNeighborsRegressor

사용 도구: Python, Numpy, Matplotlib, Seaborn, Pandas, Scikit-learn, Time 모듈 등

 [깃허브 링크](#)

 [Kaggle 데이터셋](#)

## 9. 실시간 쓰레기봉투 탐지 및 추적 (YOLOv8)

NetVision 회사의 쓰레기봉투 수거 과정을 최적화하기 위한 고급 AI 프로젝트를 설계했습니다. YOLOv8의 힘을 활용하여 우리 팀은 실시간 비디오 스트림 내에서 쓰레기봉투를 감지, 분류, 분할, 추적 및 크기 추정할 수 있는 정교한 시스템을 만들었습니다. 제 역할은 연구 논문을 읽고 YOLOv8 모델을 훈련시키는 것이었습니다. 이 프로젝트로 우리는 우송대학교 2023년 캡스톤 경진대회에 참가했습니다. 29개 학과에서 200개 이상의 팀을 이기고 우송대학교 총장상을 받았습니다.

기술적 전략: YOLOv8 통합: 최첨단 객체 감지 모델인 YOLOv8을 사용하여 라이브 비디오 스트림 내에서 쓰레기봉투를 효과적으로 식별하고 분류했습니다. 실시간 처리: 비디오 데이터를 즉시 분석하여 쓰레기봉투를 신속하고 정확하게 식별하고 추적하는 알고리즘을 구성했습니다. 정밀한 분할: 고급 분할 기술을 활용하여 비디오 프레임에서 쓰레기봉투를 정밀하게 분리하여 상세한 분석을 했습니다.

프로젝트 결과물: 감지 및 분류: 비디오 피드 내에서 다양한 유형의 쓰레기봉투(예: 재활용 가능, 재활용 불가능)를 정확하게 식별하고 분류했습니다. 분할 및 추적: 효율적인 모니터링을 위해 비디오 스트림 전체에서 쓰레기봉투를 정밀하게 분리하고 지속적으로 추적했습니다. 크기 추정: 최적의 수거 계획과 자원 할당을 위해 쓰레기봉투 크기 추정을 제공했습니다.

사용 도구: PyTorch, OpenCV, YOLOv8, Ultralytics

ML 알고리즘: YOLOv8 객체 감지

 [프로젝트 시연](#)

 [총장상 수상](#)

 [축제 영상](#)

## 10. PyTorch와 OpenCV를 사용한 실시간 얼굴 마스크 감지

이 프로젝트는 PyTorch, OpenCV 및 얼굴 감지용 MTCNN을 활용하여 마스크 착용 여부를 식별합니다. 모델은 라벨링된 데이터에서 훈련되어 이미지나 실시간 비디오 피드에서 얼굴 마스크 착용 여부를 정확하게 감지합니다.

사용 도구: PyTorch, OpenCV, MTCNN 모델

ML 알고리즘: 전이 학습을 통한 ResNet 18, ResNet 50

 [깃허브 링크](#)

 [발표자료](#)

## 11. PyTorch와 CNN을 사용한 다중 클래스 날씨 분류

PyTorch와 합성곱 신경망을 사용하여 이 프로젝트는 이미지를 다양한 날씨 조건으로 분류합니다. Kaggle 데이터셋의 사용자 정의 날씨 데이터셋을 활용하고 데이터 증강 및 전이 학습과 같은 기술을 사용하여 모델의 정확도와 성능을 향상시켰습니다.

사용 도구: PyTorch, Python

ML 알고리즘: CNN, ResNet 9

 [깃허브 링크](#)


 [Kaggle 데이터셋](#)

## 12. EDA 웹 앱 - StatFlow (Streamlit)

StatFlow EDA 앱은 사용자가 업로드한 데이터셋에서 탐색적 데이터 분석(EDA)을 수행하는 웹 기반 애플리케이션입니다. Streamlit과 데이터 사이언스 라이브러리를 사용하여 만들어졌으며, Streamlit 클라우드에서 호스팅됩니다. 기술 통계, 데이터 정리, 시각화 및 상관 분석과 같은 기능을 제공합니다.

사용 도구: Streamlit, Pandas, Numpy, Matplotlib, Seaborn, Pandas Profiling

 [깃허브 링크](#)

 [실행 링크](#)

## 13. IoT 기반 공기질 모니터링

이 프로젝트는 혁신적인 IoT 기반 실내 공기질 모니터링 시스템에 전념합니다. ESP Wi-Fi 모듈과 ThingSpeak 클라우드 플랫폼을 활용하여 다양한 센서를 통합하여 실시간 공기질 매개변수를 추적하고 표시합니다. 또한 실내와 실외 공기질을 비교하기 위해 OpenWeather API의 데이터를 통합합니다.

시스템의 주요 기능은 다음과 같습니다:



- IoT 통합: ESP Wi-Fi 모듈과 다양한 센서를 활용하여 실내 공기질 데이터를 수집합니다.
- ThingSpeak 클라우드: 다양한 센서에서 받은 실시간 공기질 매개변수를 저장하고 처리합니다.
- 실시간 대시보드: 대화형이고 사용자 친화적인 대시보드에서 공기질 지표를 표시합니다.
- OpenWeather API 통합: 실내 공기질 지표와 비교하기 위해 실외 공기질 데이터를 가져옵니다.

사용 도구: Arduino, C 언어, ThingSpeak 클라우드, ESP Wi-Fi, 센서, Streamlit

 [깃허브 링크](#)

 [발표자료](#)

 [실행 링크](#)

## 14. 신장 질환 자동 진단 (X-ray)

ResNet50, VGG16, VGG19 및 U-Net과 같은 아키텍처를 사용하여 이 프로젝트는 신장의 의료 X-ray 이미지 내에서 낭종, 결석, 종양을 포함한 신장 질환의 감지를 자동화합니다. U-Net은 분할에 사용되지만 신경망의 마지막 레이어를 변경하여 분류 작업에 사용했습니다. U-Net 오토인코더 아키텍처를 사용하여 더 높은 정확도와 정밀도를 달성했습니다. 모델은 정확한 신장 식별을 위해 딥러닝과 전이 학습 기술을 활용합니다.

사용 도구: PyTorch, Python

ML 알고리즘: 전이 학습을 통한 CNN, ResNet 50, VGG16, VGG19, U-Net

 [깃허브 링크](#)

 [발표자료](#)

 [보고서 링크](#)

## 15. 보험료 예측 (PyTorch)

이 프로젝트는 PyTorch와 머신러닝 알고리즘을 사용하여 보험료 예측에 초점을 맞춥니다. 관련 데이터셋에서 훈련된 모델은 다양한 요인을 기반으로 보험료를 예측합니다.

사용 도구: Numpy, PyTorch

ML 알고리즘: 신경망, 선형 회귀

 [깃허브 링크](#)



## 16. 주택 가격 예측 (PyTorch)

PyTorch와 머신러닝을 활용하여 이 프로젝트는 Kaggle의 관련 데이터셋을 기반으로 주택 가격을 예측합니다. 모델은 다양한 특성을 사용하여 주택 가격을 예측하도록 훈련되었습니다.

사용 도구: Numpy, PyTorch

**ML 알고리즘:** 신경망, 선형 회귀



[깃허브 링크](#)



[Kaggle 데이터셋](#)