

PORTFOLIO

AI 개발 지원자로서의 프로젝트 수행 경험 및 보유 역량을 전달하기 위해 포트폴리오를 제출합니다.



김경민 (Kim Gyeong-min)

- 부천고등학교 졸업('15년)
- 이메일: rlarudals_96@naver.com
- 휴대폰: 010-6375-2536
- 자격증: 정보처리기사
- Git: <https://github.com/Reon96/GyeongMin>

INDEX

1

Aumo-deto 서비스 개발 프로젝트

p. 03~04

- 주요 기술 적용 사항: Python, OpenVINO, YOLO 등

2

See웃 서비스 개발 프로젝트

p. 05~06

- 주요 기술 적용 사항: Python, OpenCV, GAN 등

3

Re:Play 서비스 개발 프로젝트

p. 07~08

- 주요 기술 적용 사항: Python, TF-IDF, Word2Vec

4

웹툰 장르 분류기 개발 프로젝트

p. 09~10

- 주요 기술 적용 사항: Python, LSTM, transfer-training

5

RC카 자율주행 시스템 개발 프로젝트

p. 11~12

- 주요 기술 적용 사항: C, STM32, C#, Unity 등

6

스마트 사육장 시스템 개발 프로젝트

p. 13~14

- 주요 기술 적용 사항: Python, RaspberryPi, STM32 등



Python

Python 기반 AI·머신러닝 데이터 수집·전처리·분석·시각화 전 과정 수행

Scikit-learn, OpenCV, PyTorch, OpenVINO 활용

분류·회귀·딥러닝 모델 구현 경험 보유

OpenVINO 기반 최적화로 엣지 환경에서 실시간 추론 성능을 확보



C

C 언어 기반 임베디드 시스템 및 저수준 하드웨어 제어 경험 보유

포인터·구조체·메모리 제어·인터럽트를 활용한

펌웨어 개발 및 드라이버 수준 제어 로직 구현 가능

STM32 등 MCU 기반 프로젝트 수행 경험 보유

Dev.
역량 요약



IDE

VSCode, STM32CubeIDE, Micro Studio, Unity 등

다양한 개발 환경 경험 보유

환경에 구애받지 않는 안정적인 개발 역량 보유

디버깅·버전 관리(Git)·확장 기능 활용을 통한 개발 생산성 향상



C#

C# 기반 Unity 게임 및 GUI 애플리케이션 개발 경험 보유

객체지향 설계 및 이벤트 기반 프로그래밍에 숙련

클래스·상속·인터페이스 활용으로 재사용성과 확장성이 높은

구조 구현 가능

프로젝트 개요

- **프로젝트 명**
 - 아우모-데토 (Aumo-deto)
- **목적**
 - 고령화와 귀농 인구 감소로 인한 농촌 인력 공백을 보완하고, 팔로잉카를 활용해 운반 효율을 높이는 무접점 제어 시스템 개발
- **목표 기능**
 - 제스처만으로 직관적 제어 (별도의 장치 불필요)
 - Depth Estimation 기반 거리 측정 및 사용자 추종
 - 특정 요청자만 인식하는 안전한 제어 기능
 - Edge 환경에서 실시간 동작 가능

구현 내용

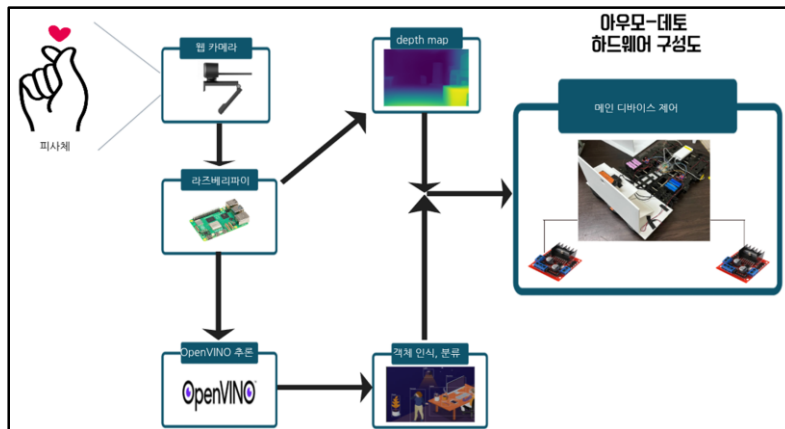
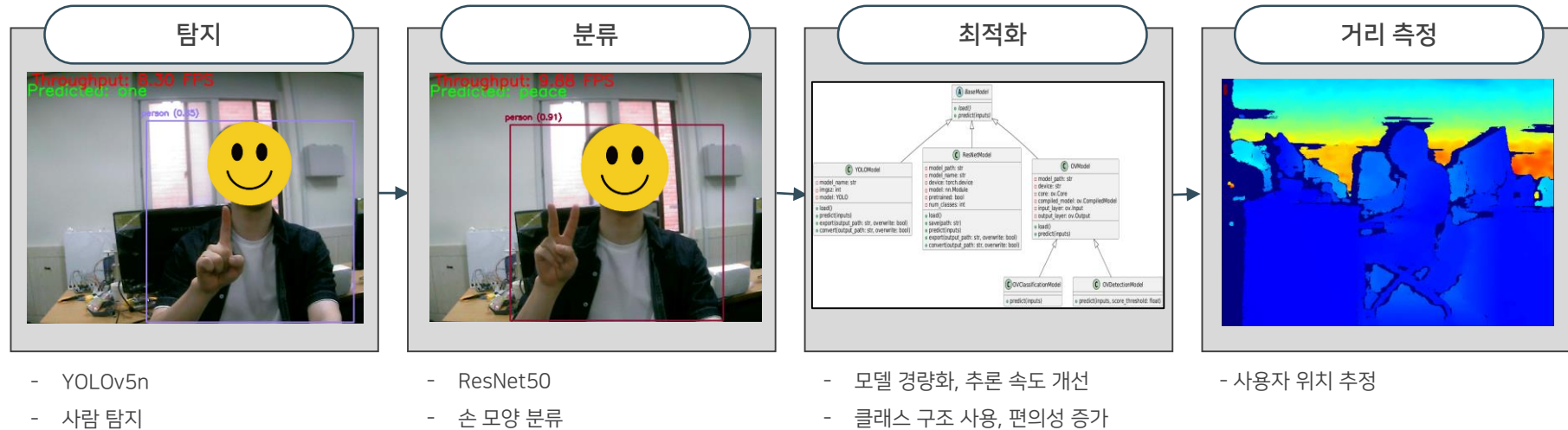
- **데이터 수집/전처리**
 - Hagrid Dataset에서 특정 손 모양만 필터링하여 라벨링
 - 이미지 폴더 구조화 및 클래스별 분류
- **모델 학습 및 최적화**
 - YOLOv5n: 사용자 탐지 → FPS 21.91 → OV 최적화 후 41.80
 - ResNet50: 손동작 분류 → OV
- **거리 측정 로직**
 - Depth Map → 노이즈 제거 → 실제 거리 매핑
 - IOU 기반 객체 고정 및 추종
- **제어 로직**
 - 특정 손동작으로만 동작하도록 제약
 - Sleep 모드 구현

역할 및 성과

- **역할**
 - 사람 인식 및 손동작 분류 모델 학습
 - OpenVINO 기반 모델 경량화 및 성능 최적화
 - 성능 벤치마킹 및 후처리 로직 개발
- **기술 스택**
 - AI: YOLO, ResNet50, OpenVINO
 - 임베디드: Raspberry Pi, Depth Camera
 - 프로그래밍: Python, OpenCV
 - 기타: Git, Linux
- **성과 및 개선점**
 - Edge 환경에서 실시간 인식 가능
 - Depth Map 품질 문제 → 매핑 기반 보완
 - PID 제어 도입 및 모터 엔코더 적용 필요

본 프로젝트를 통해 Edge에서 모델을 사용하기 위한 모델 경량화에 대해 경험하였습니다.
이 경험은 텍스트 추천·검색 기능을 서비스로 연결하는 실무 역량을 강화하는 기반이 되었습니다.

프로젝트 요약



- Raspberry Pi에서 YOLOv5n·ResNet50을 OpenVINO로 경량화해 실시간 제스처-팔로잉 제어를 구현했습니다(21.91→41.80 FPS).
- 실제 주행을 막은 Depth 한계를 원인분석하고 PID/엔코더 기반 정밀 제어와 캘리브레이션 고도화를 차기 과제로 정의했습니다.

프로젝트 개요

프로젝트 명

- Seeot - Virtual Try-On

목적

- 온라인 쇼핑 시 직접 착용이 불가능한 문제를 해결하기 위해 가상 피팅 서비스를 개발

목표 기능

- 포즈 기반 가상 피팅: OpenPose로 포즈의 key-points 탐지
- 신체 부위 인식: Human-Parsing으로 얼굴·목·팔 등 19개 부위 분할
- 이미지 합성: VTON(CP-VTON+) 기반 생성형 모델로 의류 이미지와 사용자 이미지를 결합

구현 내용

R&D

- 가상 피팅 구현에 필요한 모델 조사, Git 레포지토리 생성, 이미지 생성 과정 설계

모델 학습

- 의류 합성: CP-VTON+ 모델 학습
- 포즈 분석: OpenPose (pretrained)
- 신체 부위 분할: Human-Parsing (pretrained)

모델 추론 및 전처리

- 학습된 모델을 통해 사용자 이미지와 의류 이미지를 결합
- 포즈 및 의류 종류(상의, 하의, 신발, 장갑 등)에 따라 결과 이미지 생성

역할 및 성과

역할

- 프로젝트 전 과정 단독 수행
- 모델 구조 조사 및 학습 파이프라인 구성
- 데이터 전처리 및 추론 환경 구축
- 실행 환경 버전 충돌 해결 및 모델 재구성

기술 스택

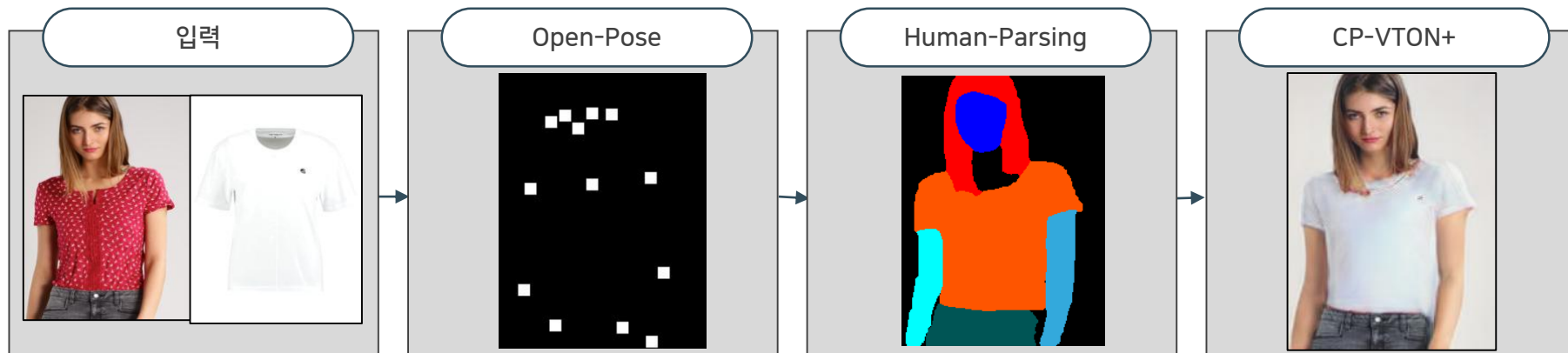
- AI: CP-VTON+, OpenPose, Human-Parsing
- 개발 환경: Python, PyTorch, CUDA
- 기타: Git, CMD 기반 환경 설정

성과 및 개선점

- 단독 프로젝트로 모델 학습·추론 파이프라인 완성
- 포즈 변화·의류 종류에 따른 가상 피팅 생성 성공
- 추후 GAN 기반 고도화 계획

본 프로젝트를 통해 데이터의 수집 및 전처리의 중요성, 모델작성요령에 대해 학습할 수 있었습니다. 이러한 역량을 인공지능의 개발 및 다양한 서비스로 사용자의 접근성에 활용하고자 합니다.

프로젝트 요약

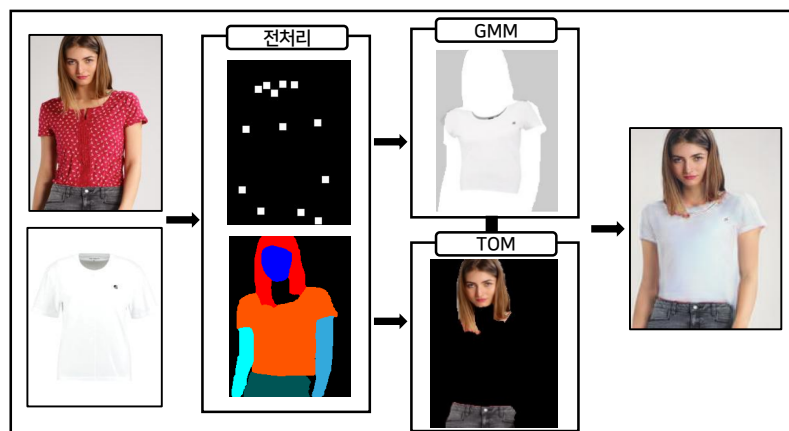


- 사용자 사진 입력
- 착용할 의류 사진 입력

- Keypoints 추출 및 시각화
- JSON 데이터 출력

- 얼굴, 목, 팔, 옷 등 19개 부위 파싱
- 의류 위치 및 신체 부위 인식

- 입력 사진 기반 가상 피팅 결과 생성



- OpenPose·Human-Parsing·CP-VTON+로 전신/의류 특성을 추출·변형·합성하여 가상 피팅을 구현했습니다.
- 단독으로 학습·추론 파이프라인을 구성하고 버전 충돌 해결까지 수행했습니다.

프로젝트 개요

- **프로젝트 명**
 - RE:PLAY
- **목적**
 - 영상 제작자 또는 방송인들이 쉽고 빠르게 저작권 무료 음악을 찾을 수 있도록 지원
- **목표 기능**
 - 유튜브 댓글·제목 기반 음악 추천
 - 키워드 기반 자동 추천
 - 플레이리스트 저장 및 불러오기

구현 내용

- **데이터 수집**
 - 유튜브 채널(NoCopyrightSounds) → 영상·댓글 크롤링
- **전처리 및 라벨링**
 - 번역(외국어→한국어) → 형태소 분석 → 불용어 제거
- **벡터화**
 - TF-IDF, Word2Vec 적용
- **UI/서비스**
 - 키워드·제목 검색 가능
 - 랜덤 추천 키워드 제공
 - 재생목록 저장/불러오기

역할 및 성과

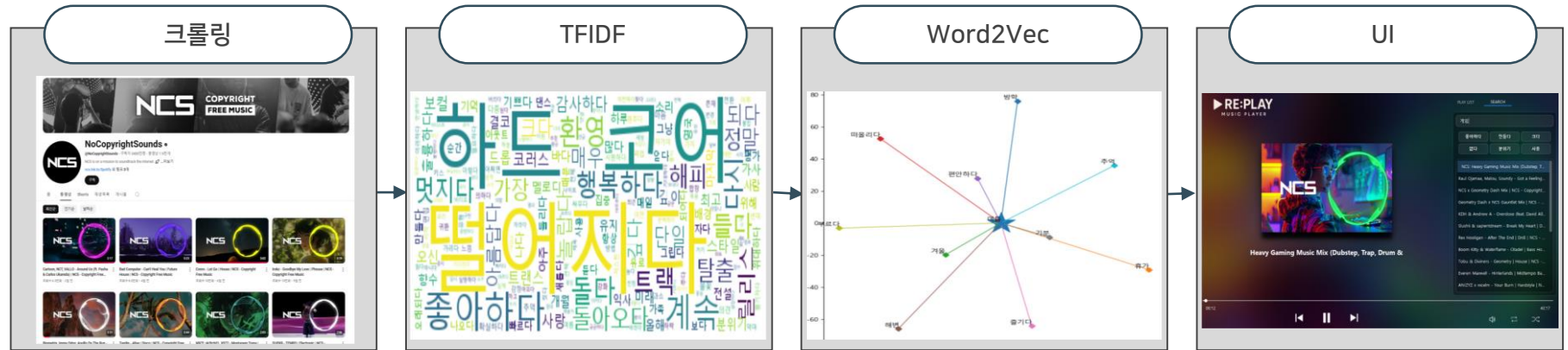
- **역할(팀장)**
 - 유튜브 데이터 크롤링 (영상 ID, 제목, 댓글)
 - NLP 전처리 및 벡터화
 - 시스템 코드 구현
- **기술 스택**
 - AI: TF-IDF, Word2Vec
 - 프로그래밍: Python, PyQt (Qt Designer)
 - 데이터 수집: Selenium
 - 기타: pandas, scikit-learn
- **성과 및 개선점**
 - 텍스트 기반 음악 추천 구현
 - 오류 발생 시 데이터 저장 기능 도입
 - 추천 키워드 갱신 기능 도입

본 프로젝트를 통해 NLP 전처리와 벡터화, 유사도 기반 추천 설계, **모델-UI 연동**을 한 번에 경험했습니다.
 이 경험은 텍스트 추천·검색 기능을 서비스로 연결하는 실무 역량을 강화하는 기반이 되었습니다.

E2E 파이프라인을 1주 내 구현, 사용자 기능까지 연결했습니다.
콘텐츠 기반 추천을 서비스 형태로 역량을 강화했습니다.

[Intel] Edge AI SW 개발자 양성과정
 팀 프로젝트 / '25년 5월 약 1주간

프로젝트 요약

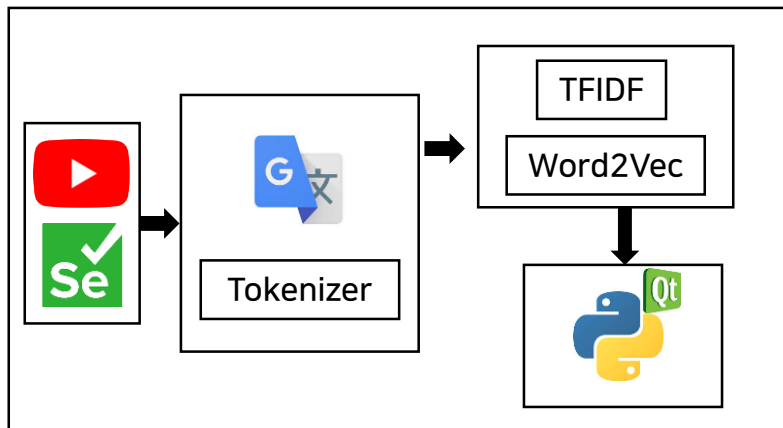


- 각 영상의 고유 ID 추출
- 추출 후 리스트에 저장

- 문서 내 단어의 중요도를 수치로 나타내는 통계적 방법

- 단어를 컴퓨터가 이해할 수 있도록 숫자 벡터로 표현하는 기법

- 썸네일, 제목, 재생목록, 검색창, 랜덤 추천 단어, 재생바, 다양한 버튼으로 구성



- 유튜브 NCS의 제목·댓글을 크롤링해 번역→형태소→불용어 제거→TF-IDF/Word2Vec 벡터화로 음악을 추천하고 PyQt UI와 연동했습니다.
- 사용자 관점에서 키워드/제목 검색·랜덤 추천·재생목록 저장을 지원하는 E2E 추천 툴로 완성했습니다.

프로젝트 개요

- **프로젝트 명**
 - DKS Webtoon
- **목적**
 - 다양한 플랫폼(Naver, Kakao, Lezhin) 웹툰 데이터를 활용해 자동 카테고리 분류 모델 개발
- **목표 기능**
 - 웹툰 제목과 줄거리 기반 카테고리 분류
 - 학습 전략 비교
(일반 학습, 점진적 학습+전이학습, 전이학습)

구현 내용

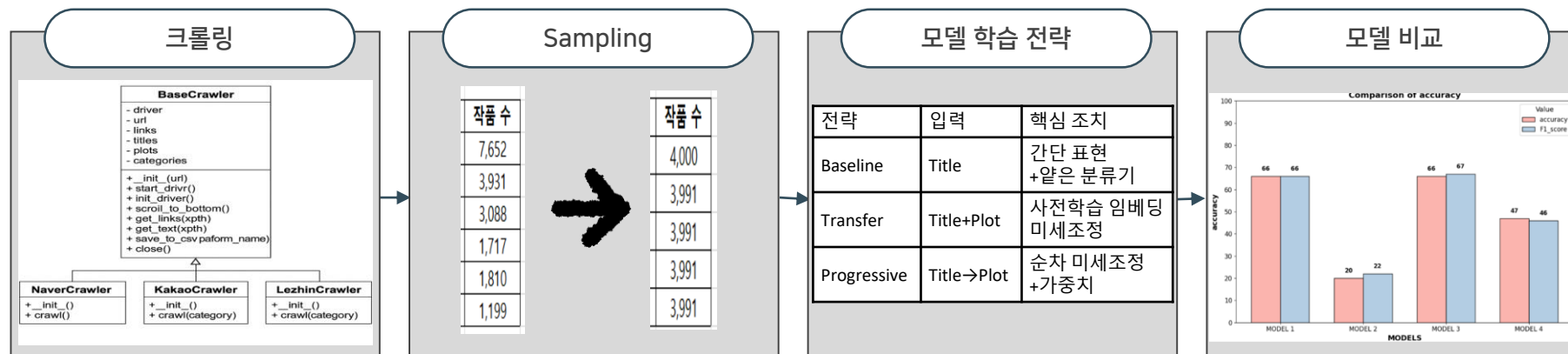
- **데이터 수집**
 - 각 플랫폼 크롤링
→ CSV 저장(utf-8-sig 인코딩)
- **데이터 전처리**
 - 카테고리 축소 및 데이터 불균형 조정
- **모델 학습 전략**
 - Title 단독, Plot 단독, Title+Plot 학습 비교
 - 점진적 학습+전이학습 방식으로 성능 향상 시도
- **결과 분석**
 - 학습 방식별 정확도·처리속도 비교
 - 불균형 데이터 문제와 장르별 언어적 차이 확인

역할 및 성과

- **역할(팀장)**
 - 프로젝트 기획 및 일정 관리
 - 모델 학습 및 예측 구현
 - 발표 및 결과 분석
- **기술 스택**
 - AI: Transformer, Attention Mechanism
 - 프로그래밍: Python, scikit-learn, PyTorch
 - 데이터 수집: Selenium
- **성과 및 개선점**
 - 전이학습 기반 모델에서 분류 정확도 향상
 - 카테고리 불균형 해소
 - Attention 및 Transformer 기반 고도화 추가 필요

멀티 플랫폼 크롤링과 CSV 파이프라인 표준화, 불균형 대응 및 학습 전략 비교를 경험하였습니다.
 이러한 역량은 콘텐츠 분류·추천 모델을 서비스에 연동하고 운영 품질을 개선하는 데 활용하고자 합니다.

프로젝트 요약

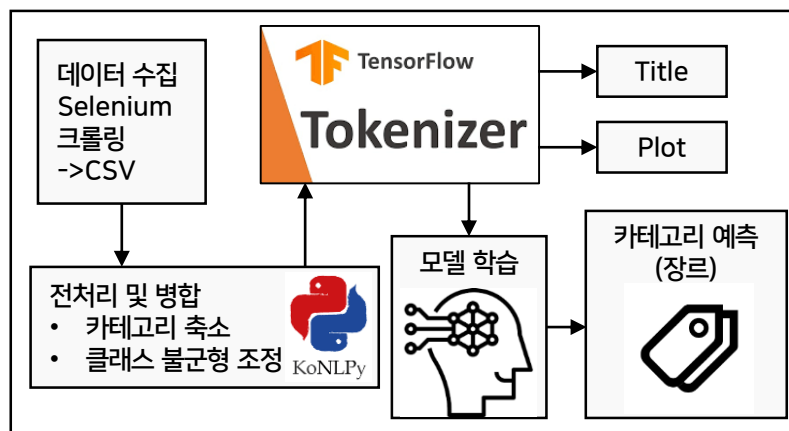


- 크롤링관련 부모 Class 구현
- 플랫폼별 상속받아 자식 class 구현

- 전체적인 작품수를 균일하게 맞춰 학습에 도움을 주기 위해

- 제목 과 줄거리를 따로 학습하여 전이 학습과 점진적 학습 진행

- 일반적 학습, 점진적 학습 + 전이학습, 전이학습 등



- 여러 플랫폼의 웹툰을 크롤링해 CSV 표준화 후, 제목/줄거리 토큰화를 거쳐 학습전략(일반·전이·점진+전이)을 비교했습니다.
- 불균형/장르 언어 차이를 다루며 분류 정확도 향상을 확인했고, Transformer/Attention 기반 고도화를 제안했습니다.

프로젝트 개요

- **프로젝트 명**
 - 초음파 RC카
- **목적**
 - 드라이버 모듈+DC모터 속도 제어 기반으로 Bluetooth 원격/자율주행 실험 및 장애물 회피 안정화
- **목표 기능**
 - 드라이버 모듈 + DC모터 속도 제어
 - Bluetooth + 스마트폰 앱 원격 제어
 - 초음파 센서를 통한 자율주행 구현

구현 내용

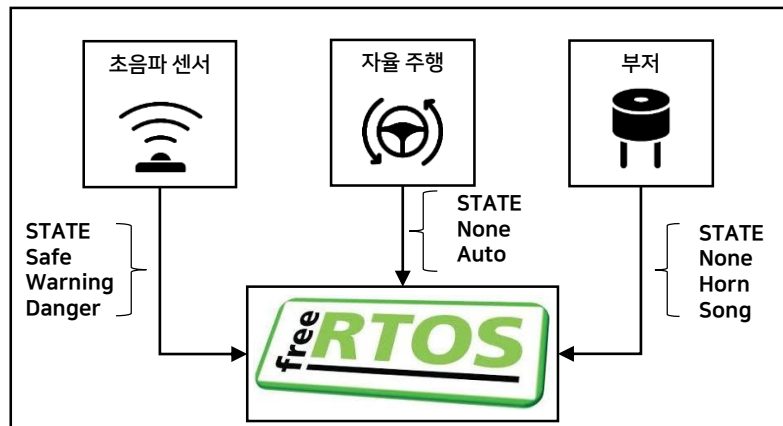
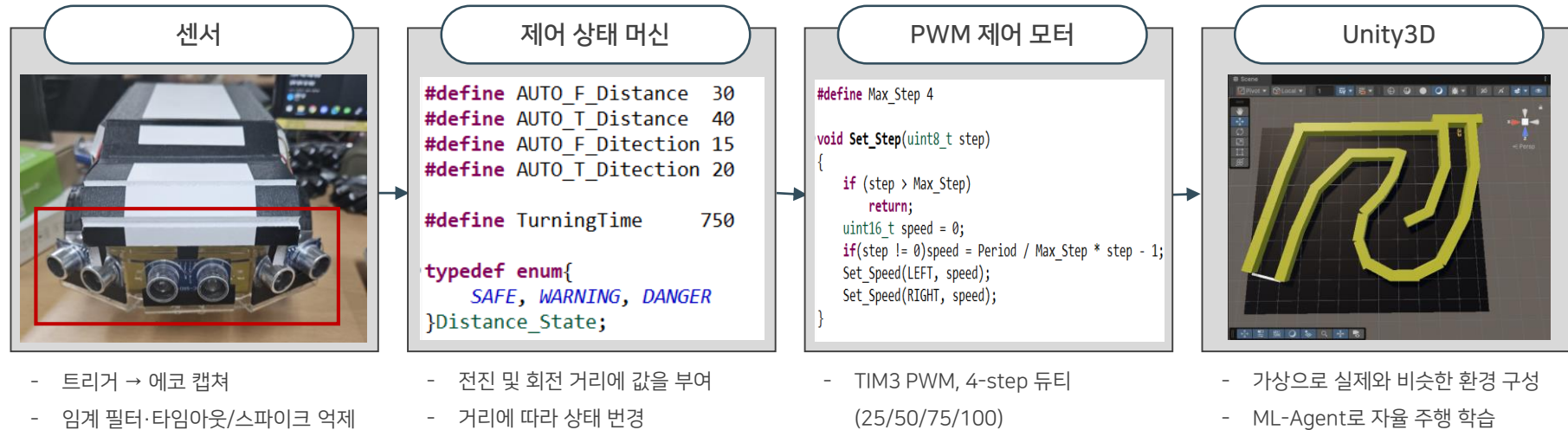
- **모터 제어(PWM)**
 - TIM3 PWM, 4-step 듀티
 - 좌/우 독립 제어(L298N)
- **RTOS 구조**
 - Ultrasonic / AutoRun / Buzzer 태스크
 - 큐·타이머 기반 논블로킹
- **센서 처리**
 - HC-SR04 ×3 캡처 인터럽트
 - 임계 필터·타임아웃으로 스파이크 억제
- **자율 로직**
 - 거리 임계값 기반 상태머신(RC/자율)
- **통신·안전**
 - Bluetooth(HC-06) 원격 제어
 - Backward-safe(후진+부저 알림)

역할 및 성과

- **역할**
 - FreeRTOS 태스크/큐 설계, 인터럽트 기반 초음파, PWM 모터 제어, 통신, 자율 로직
- **하드웨어**
 - MCU: STM32F411
 - 구동 : L298N, DC 모터×4
 - 센서 : HC-SR04(초음파 센서)×3
 - 통신 : HC-06(Bluetooth)
 - 부저 : 타이머 PWM 구동
- **성과 및 개선점**
 - 블로킹 코드 제거로 제어 루프 주기 안정화
 - 초음파 임계 필터 적용으로 회피 동작 향상
 - 모듈형 태스크 구조로 인코더/PID·Unity 시뮬 연동 등 고도화 준비 완료.

인터럽트·DMA 기반 논블로킹 태스크와 상태머신으로 RC/자율 주행 제어를 끝단까지 구현·검증하였습니다.
 이러한 역량은 AGV·모바일 로봇의 모터/센서 제어와 실시간 주행 소프트웨어에 활용하고자 합니다.

프로젝트 요약



- 초음파·자율 주행·부저를 FreeRTOS 상태머신으로 묶어 동작을 결정하는 논블로킹 제어 구조를 구현했습니다.
- Unity 가상환경을 구축해 ML-Agents로 자율주행 정책 학습을 시도하여, 실제 제어 로직 고도화를 위한 기반을 마련했습니다.

프로젝트 개요

■ 프로젝트 명

- 스마트 파충류 사육장

■ 목적

- 온습도를 실시간으로 측정·저장하고, 설정값을 바탕으로 환경을 제어하는 스마트 사육장 구축

■ 목표 기능

- DHT11를 이용한 온습도 측정
- 측정한 값을 블루투스 통신으로 STM32로부터 라즈베리파이에 전송
- 수신한 데이터를 라즈베리파이의 DB에 저장
- 온습도 데이터를 GUI 앱으로 시각화

※ 기본 기능 수행은 실시간으로 이루어져야 함.

구현 내용

■ 임베디드

- CubeIDE에서 GPIO(DHT11/팬/RGB), I²C(LCD), USART6(HC-06) 설정·펌웨어 구현

■ 라즈베리파이 서비스화

- Bluetooth 수신·DB I/O 모듈화 후 systemd 등록(부팅 자동 실행), MariaDB 스키마 설계

■ GUI/제어 로직

- PyQt 실시간 그래프·세트포인트 제어
- LCD 상태표시/팬·가습기 구동
- RGB LED로 온도 상태 표시

■ 신뢰성 보강

- RxEventCallback로 수신 안정화
- 멀티프로세싱 충돌→단일 프로세스로 통합
- 팬 저항값 재설계로 구동 문제 해결

역할 및 성과

■ 역할(팀장)

- Bluetooth 통신 기능 개발
- 임베디드-라즈베리파이-GUI E2E 통합
- 주요 이슈 트러블슈팅 총괄

■ 기술 스택

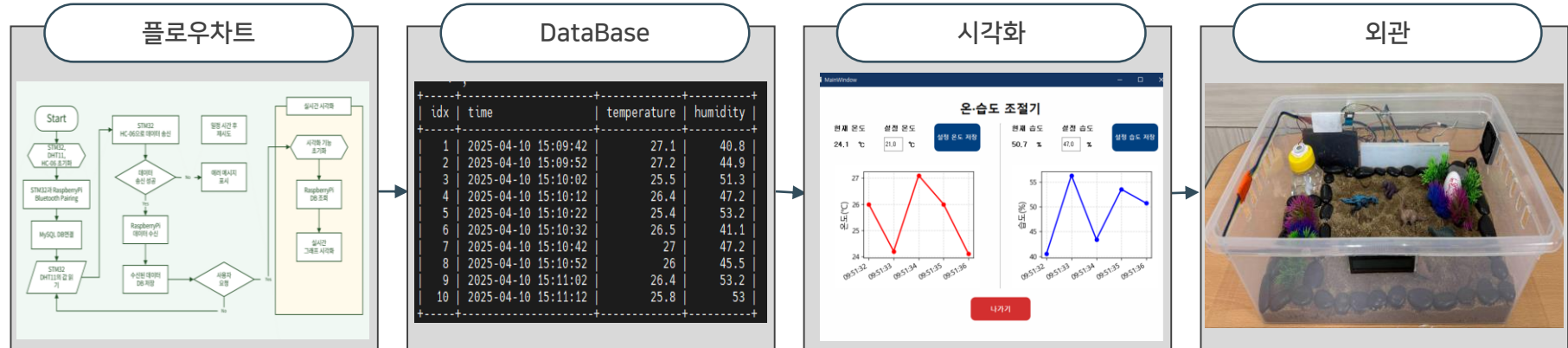
- STM32(CubeIDE), C, Python, PyQt, MariaDB/MySQL
- STM32F411, Raspberry Pi, DHT11, LCD/팬/가습/RGB LED, Bluetooth(HC-06)

■ 성과 및 개선점

- 센싱→전송→저장→시각화→제어의 실시간 파이프라인 구현
- DB 권한/접속 정책 정비
- 동시성·하드웨어 이슈 해결로 운영 안정성 확보

엣지 환경에서의 실시간 데이터 파이프라인 설계와 임베디드-서비스-UI의 통합 운영을 경험하였습니다.
 이러한 경험은 IoT/AI 서비스에서의 신뢰성 있는 데이터 흐름 설계와 운영 자동화에 활용하고자 합니다.

프로젝트 요약

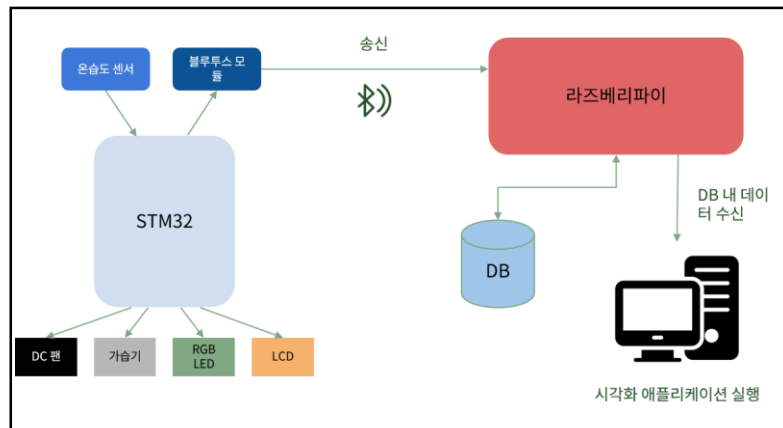


- DHT11→BLE→수신/검증
→DB→GUI

- temperature, humidity, status
- 매일 평균 저장 및 일일DB초기화

- PyQt 실시간 그래프, 온습도 세팅
- 가습 ON/OFF, 경고 시 RGB 작동

- STM32F4 + DHT11 + HC-06 +
팬/가습기/LCD/RGB 통합



- STM32가 온·습도를 측정해 BLE로 라즈베리파이 에 전송하고, 라즈베리파이는 DB 저장과 실시간 시각화를 담당합니다.
- 팬·가습기·RGB LED·LCD 제어는 STM32가 수행하며, 라즈베리파이 앱은 systemd(systemctl) 등록으로 재부팅 시 자동 실행됩니다.