

임베디드 시스템 설계 및 실험

004 분반 – 2 조 13 주차

실험 보고서

TinyML 2

실험자	202055606 주우성
	202055623 허치영
	202255632 벌드 바타르 아마르투브신
	201724637 오치어 자미안푸레브
	202055629 밧툴가 바잘삿
실험날짜	2024-11-28
제출날짜	2024-12-04

1. 실험 개요

이번 주 연습에서는 ESP 기반 MCU 펌웨어 포팅과 TensorFlow Lite for Microcontrollers를 사용하여 머신 러닝 모델을 배포하는 방법에 대해 다룹니다. 이 과정은 개발 환경 설정, 펌웨어 컴파일 및 ESP32-S3 모듈에 플래시, 그리고 모션, 음성, 비전과 같은 센서 입력을 위한 머신 러닝 모델 적용을 포함합니다.

2. 목표

이번 실험의 주요 목표는 다음과 같습니다:

- ESP 기반 마이크로컨트롤러에 펌웨어 포팅 이해
- ESP-IDF 프레임워크 사용법 익히기
- TensorFlow Lite와 TensorFlow Lite for Microcontrollers에 대한 이해

3. 필요한 환경

이 실험을 위한 환경 설정은 다음과 같습니다:

- 운영 체제: Windows
- IDE: Visual Studio Code (최신 버전 권장) 1
- 도구: ESP-IDF Extension for VS Code, Docker, 보드를 관리하기 위한 Python 라이브러리

4. 실험 과정

1. 환경 설정

- 1) Visual Studio Code와 ESP-IDF Extension 설치
- 2) ESP32-S3에 맞는 설정 및 확장 기능 구성
- 3) Windows의 장치 관리자에서 COM 포트를 확인하여 시리얼 통신 설정

2. 예제 0: 'Hello World' 프로젝트

1) 펌웨어 빌드:

- hello world 예제 디렉토리로 이동하여 다음 명령어 실행:

idf.py build

2) 펌웨어 플래시:

- 다음 명령어를 사용하여 펌웨어를 업로드하고 출력 모니터링:

ldf.py flash monitor

3. 과제 작업

다음 프로젝트 중 하나를 선택하여 실습합니다:

- pnu_exp_motion: 가속도계를 사용하여 움직임을 감지하고 분류 (예: 왼쪽, 오른쪽, 앞으로, 뒤로).
- pnu_exp_speech: 마이크를 사용하여 음성 명령을 분류 (예: 시작, 중지).
- pnu_exp_vision: 카메라를 사용하여 조명 조건에 따라 이미지를 분류.

4. 문제 해결

일반적인 문제는 플래시 설정, 컴파일 오류 또는 시리얼 포트 구성 오류입니다. 설정을 초기화하고 다시 빌드하려면 다음 명령어를 사용합니다:

ldf.py clean

ldf.py fullclean

5. 모델 배포 및 테스트

TinyWebTrainer를 사용하여 모델을 훈련한 후, .tflite 모델을 .c 파일로 변환하려면 xxd 유틸리티를 사용합니다:

```
xxd -i modelfile.tflite > modelfile.c
```

변환된 모델을 ESP32-S3에 플래시하고, 관련 센서를 사용하여 모션, 음성 또는 비전 분류 테스트를 수행합니다.

5. 결론

이번 실험은 TinyML 프로젝트를 설정하는 과정에서 환경 설정, 데이터 수집, 모델 훈련, 배포 및 문제 해결에 대해 배우는 기회를 제공했습니다. ESP32-S3 모듈과 TensorFlow Lite for Microcontrollers를 사용한 머신 러닝 모델 배포 경험을 통해 임베디드 시스템에서의 모델 활용을 이해할 수 있었습니다.