프로젝트 개요

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **프로젝트명** | (국문) 무디시 | | | |
| (영문) MOODISH | | | |
| **프로젝트**  **요약**  **(abstract)** | 사용자의 감정, 냉장고 재료, 조리 시간 등을 고려하여 최적의 요리 레시피와 관련 유튜브 영상을 추천하는 애플리케이션. | | | |
| **팀명** | 무디 | | | |
| **Team page** | https://github.com/may0611/2024\_MOODISH | | | |
| **팀원(1)** | 성 명 | 이채은 | 학 번 | 202237783 |
| email | [khce03](mailto:khce03@gachon.ac.kr)@gachon.ac.kr | 연락처 | 010-4935-2382 |
| github | https://github.com/may0611 | | |
| **팀원(2)** | 성 명 | 임동현 | 학 번 | 202035377 |
| email | [ehd9608@gachon.ac.kr](mailto:ehd9608@gachon.ac.kr) | 연락처 | [0](file:///Users/maylee/Downloads/123-456-7890)10-8191-9608 |
| github | https://github.com/saru033 | | |
| **팀원(3)** | 성 명 | 임세은 | 학 번 | 202035378 |
| email | sunnyliml@gachon.ac.kr | 연락처 | 010-9281-8366 |
| github | https://github.com/KWLEJ | | |
| **팀원(4)** | 성 명 | 성낙연 | 학 번 | 202035342 |
| email | sny7942@gachon.ac.kr | 연락처 | 010-3065-5369 |
| github | https://github.com/sny7942 | | |

프로젝트 기획(안) 및 수행계획서

|  |  |
| --- | --- |
| **프로젝트명** | (국문) 무디시 |
| (영문) MOODISH |

# **1. 수행계획서**

# 가. 프로젝트 개요 (요약)

|  |
| --- |
| MOODISH는 감정 기반으로 최적의 요리 레시피와 관련 유튜브 요리 영상을 추천하는 맞춤형 애플리 이션입니다. 현대인의 삶에서 음식은 단순히 생존을 위한 수단을 넘어서 스트레스 관리, 감정 전환에도 중요한 역할을 하고 있습니다. 그러나 기존의 요리 추천 시스템은 사용자의 감정 상태는 고려하지 않기 때문에, 그러한 점에서 차이점을 두고 더욱 좋은 서비스를 제공할 수 있는 애플리케이션을 제작하고자 하였습니다. MOODISH는 사용자가 냉장고 속 재료, 기분, 조리 가능 시간, 보유 조리 기구 정보 등을 입력하면 AI 기술(추천 시스템)을 활용하여 최적의 요리를 추천합니다. 이 애플리케이션은 여러가지 기능을 제공합니다. 감정 기반 요리 추천, 냉장고 재료 관리(이미지 인식 or 영수증 텍스트 인식), 감정 일기 및 기록 저장, 개인화된 데이터 기반으로 맞춤 레시피를 제공하고, 마지막으로는 AI로 분석된 유튜브 요리 영상을 추천해주는것입니다.  이 프로젝트는 음식과 감정 관리를 통해 개인의 정신 건강 증진과 자원 활용 극대화를 목표로 합니다. 또한 사용자의 데이터를 지속적으로 학습하여 더 정교하고 맞춤화된 추천 서비스를 제공합니다. MOODISH는 이를 통해 식재료 낭비를 줄이고, 사용자의 삶의 질을 높이며, 요리 자체를 하나의 즐거운 경험으로 변화시키는 데 기여할 수 있다고 생각합니다. |

# 나. 목표 및 필요성

|  |
| --- |
| -감정을 기반으로한 레시피 추천, 남은 재료 소진  - 감정에 따라 먹고싶은 것이 달라지고, 이에 따라 그저 음식을 넘어서 감정에 따라 반영할 수 있는 음식을 추천을 할 수 있습니다.  - 냉장고 속 남은 재료의 활용도를 높이기 때문에 식자재 낭비를 감소시킬 수 있습니다. (자원 절약)  스크린샷, 텍스트, 도표, 폰트이(가) 표시된 사진  자동 생성된 설명   * **가정에서 음식물 쓰레기의 대부분이 나오고있음.** |

# 다. 세부 개발 내용 및 방법

|  |
| --- |
| 도표, 텍스트, 라인, 기술 도면이(가) 표시된 사진  자동 생성된 설명  **프로젝트 설계: 사용자의 감정 및 냉장고 재료에 기반하여 맞춤형 레시피를 추천**  **<UI/UX 설계>**  Figma로 앱 와이어프레임 제작, 사용자 경험을 고려한 직관적인 화면 설계 프로젝트 구현: 프론트엔드(React Native로 앱 개발)   1. 상태 관리 React Navigation을 사용하여 화면 간 전환 및 상태 관리 2. 디버깅 및 테스팅 React Native Debugger와 Jest를 사용하여 앱의 동작을 점검하고 단위 테스트를 통해 안정성 보장 3. 최종 빌드 및 배포 최종 앱 빌드를 완료한 후 App Store 및 Google Play에 배포 준비   **<감정 인식 시스템>**   1. 데이터 흐름   입력 - 앱에서 사용자 이미지나 실시간 카메라 스트림으로 촬영  처리 - FER 오픈소스를 통해 감정을 인식  출력 - 감정 분석 결과와 확률 값   1. 기술 스택   Python을 통해 감정 인식 처리를 할 예정입니다.  **구현 단계**   1. FER 오픈소스를 이용하여 백엔드를 구현합니다. 2. React Native로 프론트엔드와 연결을 합니다. 3. 앱 화면에 감정 상태 표시를 해줍니다.   **테스트 및 배포**   1. 테스트   다양한 표정 데이터로 감정 인식 정확도를 검증할 예정입니다.  앱에서 카메라 입력의 품질과 처리 속도도 테스트할 예정입니다.   1. 배포   테스트 완료 후 배포를 한 다음, 정말 앱을 만들고 나서 피드백을 또 받고,  그에대한 개선을 할 예정입니다.  **<추천 시스템>**  Content- based filtering 과 SVD를 이용한 하이브리드 추천 시스템  Content-based에서 각 레시피를 분석해서 조리 난이도, 소요시간, 재료, 조리기구, 등등을 feature로 해서 회원 가입시 작성한 유저 profile에 감정 + 조리시간을 입력받아 content-based로 우선 100개 정도의 사용자에게 맞는 레시피를 추천하고 ,그 100개에서 svd를 사용해서 자신의 선호도와 맞는 레시피를 최종적으로 3개 정도 추천.  선호도에서 매번 레시피의 평가를 요구하면 편의성이 떨어지므로 레시피를 클릭해보거나, 실제로 사용해 본 레시피에 +1점을 부여하여 rating을 대체. 매번 레시피를 누를 때 마다 데이터베이스에 연동하면 프로그램이 느려질 수 있으므로 캐시 처럼 프로그램 내에 user-recipe-rating 형태의 table을 만들어서 여기에 정보를 저장해 두었다가 프로그램을 종료하거나 레시피를 추천할 때, 테이블이 길어질걸 막기 위해 테이블 길이가 30정도 가 될 때만 데이터베이스에 최종 반영  **<텍스트인식 시스템>**   * **기능 정의**:   영수증 이미지 업로드 및 처리.  텍스트 추출(OCR, Optical Character Recognition).  추출된 데이터에서 핵심 정보(상점명, 날짜, 품목명, 가격 등) 파싱.  결과를 저장 및 출력.  **1. 기술 스택 선정**   * **프로그래밍 언어**: Python . * **OCR 라이브러리**: Tesseract-OCR (오픈소스). * **데이터 저장소**: SQLite, PostgreSQL 등. * **배포 환경**: Flask, FastAPI 또는 라즈베리파이의 로컬 서버.   **2. 시스템 설계 및 구현**  **(1) 데이터 수집**   * **입력 데이터**: * JPEG, PNG 등의 영수증 이미지 파일. * 모바일 카메라 또는 파일 업로드 기능 구현.   **(2) 이미지 전처리**   * **전처리 단계**:   + OpenCV로 이미지 개선:   + 크기 조정 및 OCR 라이브러리에 적합한 포맷으로 변환.   **(3) OCR(Optical Character Recognition) 적용**   * **OCR 실행**: Tesseract-OCR을 사용해 이미지에서 텍스트 추출. * **결과 검증 및 후처리**:   + 추출된 텍스트의 불필요한 공백 및 특수 문자 제거.   + 텍스트에서 의미 있는 데이터만 필터링(정규표현식 사용).   **(4) 핵심 정보 파싱**   * **구조화된 데이터 추출**:   + 정규표현식으로 날짜, 품목 등을 필터링.   **(5) 데이터 저장**   * **데이터베이스 설계**:   + 테이블 이름: Receipts.     - id: 기본 키.     - store\_name: 상점명.     - date: 날짜.     - item: 품목명.     - price: 금액.     - timestamp: 처리 시간.   **<재료인식 시스템>**  **1. 라즈베리파이 환경 설정**   * **운영체제 설치**:   + Raspberry Pi OS(Lite 또는 Full) 설치. * **필수 라이브러리 설치**:   + Python3, OpenCV, TensorFlow/PyTorch, Flask/FastAPI 설치.   + 카메라 인터페이스를 위한 picamera2 또는 opencv-python.   **2. 시스템 설계 및 구현**  **백엔드 및 추론 코드**   * **이미지 캡처 및 처리**:   + 카메라로 이미지 캡처 후 OpenCV로 전처리. * **모델 추론**:   + 이미지를 입력으로 받아 재료 인식 수행.   **3. 데이터 저장**   * **로컬 데이터베이스**:   + SQLite를 사용하여 인식된 재료와 시간 정보를 저장. * **데이터 저장 구조**:   + 테이블 이름: Ingredients.     - id: 기본 키.     - name: 재료 이름.     - confidence: 인식 확률.     - timestamp: 시간 기록.   **4. 테스트 및 디버깅**   * **하드웨어 테스트**:   + 카메라 연결 및 이미지 캡처 확인. * **성능 테스트**:   + 모델 추론 시간 확인.   + 인식률 테스트(Precision, Recall). |

# 마. 선행 기술 및 관련연구(동일, 유사 기술에 대한 차별성, 독창성, 혁신성 등 명시적 포함)

|  |
| --- |
| **1. 선행 기술 및 관련 연구**   * **냉장고 재료 인식 관련 기술**   + **LG 스마트 냉장고 (ThinQ AI)**:   LG의 AI 기반 스마트 냉장고는 내부 카메라와 이미지 인식 기술을 활용해 재료를 자동으로 인식하며, 사용자에게 남은 유통기한을 알려준다.   * + - **기술적 특징**: 딥러닝 기반 이미지 분류 모델(CNN)을 사용하여 재료의 종류를 판별.     - **한계**: 사전 정의된 재료 세트에 한정되며, 사용자가 추가적으로 데이터 학습을 시킬 수 없음.   + **삼성 패밀리 허브**:   내부 카메라와 모바일 연동을 통해 냉장고 내부 재료를 사용자에게 실시간으로 제공. 재료 인식은 주로 수동 방식으로 입력하거나 사용자가 직접 확인해야 함.   * + - **기술적 한계**: 자동 재료 인식 기능이 부족하며, 사물 인식 AI의 도입이 미흡. * **유사 연구 사례**:   + **Vision-based Food Recognition**:Vision AI를 활용해 음식의 종류를 인식하는 연구로, 음식을 식별하는 딥러닝 모델 (ResNet, EfficientNet)을 활용.     - **결론**: 데이터의 다양성과 라벨링 품질이 시스템 성능에 직접적인 영향을 미친다는 것을 보여줌.   + **Food Recognition in IoT Devices**:IoT 기반 기기에서 음식 이미지를 처리하는 연구로, 경량화된 모델(TFLite, MobileNet)과 임베디드 디바이스에서의 추론 속도를 실험.     - **결론**: 하드웨어 리소스가 제한적인 경우에도 이미지 인식이 가능하다는 점을 확인.   **2. 관련 오픈소스 SW**   * **TensorFlow Lite**:   경량화된 딥러닝 모델을 라즈베리파이와 같은 임베디드 시스템에서 효율적으로 실행할 수 있도록 지원.   * + **활용 계획**:     - 사전 학습된 이미지 분류 모델을 경량화하여 라즈베리파이에서 실행.     - 데이터셋 추가 학습으로 재료 인식 정확도를 높임. * **OpenCV**:   이미지 처리 및 전처리를 위한 강력한 오픈소스 라이브러리.   * + **활용 계획**:     - 카메라에서 입력받은 이미지를 전처리(크기 조정, 노이즈 제거)하여 딥러닝 모델에 입력. * **Flask/FastAPI**:   Python 기반의 경량 웹 프레임워크로, 라즈베리파이에서 REST API를 구현하여 다른 디바이스와 통신 가능.   * + **활용 계획**:     - 인식 결과를 REST API로 제공하여 사용자 디바이스로 전송. |

# 바. 기대효과 및 활용 분야

|  |
| --- |
| **기대효과:**   * 사용자 맞춤형 레시피로 개인화 서비스 만족도 증대. * 정신 건강 관리 및 스트레스 완화 * 식재료 낭비 감소로 지속 가능성 기여   **활용 분야:**   * 다국어를 지원함으로써 글로벌 시장 진출 가능성 * 헬스케어와 연동함으로써 건강 상태 기반 요리 추천 가능 * 소셜 피드 기능으로 사용자 간 커뮤니티 구축 |

# **2. 구현 계획**

# 가. 세부 구현 계획

|  |
| --- |
| <프론트엔드 세부계획>앱 와이어프레임 제작Figma를 활용한 화면 레이아웃 설계로그인/회원가입, 홈 화면 개발React native를 사용하여 로그인/회원가입, 홈 화면 구현주요 기능 화면 개발사용자 초기 정보 조사 화면감정별 사용자 선호 음식, 보유 조리 기구, 냉장고 재료 인식 방법 선택 화면 구현감정 인식/선택, 추천 레시피 및 영상 제공 화면감정 인식/선택, 추천 레시피 및 영상 표시 화면 구현사용자가 입력한 정보들을 바탕으로 추천 데이터를 받아오고 이를 화면에 반영마이페이지 화면사용자 정보, 감정 기록 및 감정 일기 기능과 사용자 피드백을 할 수 있는 UI 설계<감정 인식 시스템 세부 계획>오픈소스 분석 및 선택FER 오픈소스 및 관련 라이브러리 서치 및 선택표정 데이터셋을 준비환경 구축개발 환경 설정(Python 등)필요한 라이브러리 설치(opencv-python 등)3. FER 오픈소스 실행 테스트사전 학습된 FER 모델을 활용하여 샘플 데이터 분석 테스트기능 구현 및 통합FER 오픈소스를 활용한 감정 분석 시스템을 구현API 설계 및 백엔드 구축(Flask, FastAPI 등을 기반)감정 분석 결과를 시각적으로 출력하는 것을 구현<텍스트인식 시스템 세부 계획>오픈소스 선택 및 학습OCR - tesseract영수증 인식 시스템 구현<재료인식 시스템 세부 계획>하드웨어 설정라즈베리파이 설정:Raspberry Pi OS 설치 및 기본 설정.카메라 모듈 연결 및 활성화(raspi-config).카메라 테스트:카메라 이미지 캡처 확인(raspistill, opencv).조명 환경 조정(LED 추가 또는 밝기 설정).데이터 준비 및 학습 모델1) 데이터 수집:일반적인 냉장고 재료 데이터셋 확보(Fruit-360, Grocery Store Dataset 등).추가적으로 직접 수집한 냉장고 내부 이미지.2) 데이터 전처리:OpenCV로 이미지 리사이즈, 크롭 및 노이즈 제거.데이터 증강(회전, 밝기 변경, 확대/축소).3) 모델 선택 및 학습:경량 모델 사용:MobileNetV2, EfficientNet, 또는 ResNet.모델 학습:TensorFlow/PyTorch로 학습 후 TFLite로 변환.모델 최적화:양자화(Quantization) 적용으로 추론 속도 개선.3. 이미지 처리 및 재료 인식1) 이미지 캡처 및 저장:OpenCV를 사용하여 라즈베리파이 카메라에서 이미지 캡처.이미지를 로컬 디스크 또는 메모리에서 처리.2)이미지 전처리:이미지 리사이즈(224x224 등 모델 입력 크기에 맞춤).대비 조정(히스토그램 평활화 등).3)모델 추론:TensorFlow Lite 모델 로드.입력 이미지에서 재료 이름 및 확률값 추출.4. 데이터 저장 및 제공데이터베이스 설계:테이블: ingredients.id: 기본 키.name: 재료 이름.confidence: 모델 확률값.timestamp: 인식 시간.결과 저장:인식된 데이터를 SQLite 또는 PostgreSQL에 저장.중복 데이터는 최근 업데이트 시간만 기록.<추천 시스템>데이터 전처리필요한 columns만 추출, 이후 NULL값 제거, 중복되는 데이터 제거, 조회수가 10000이상인 데이터만 선택.이후 one-hot 인코딩 으로 카테고리들 구분.추천 시스템 - Content Based유저의 input값과 전처리된 데이터들과의 Cosine 유사도로 유저에게 가장 잘 맞는 데이터 100개를 우선 뽑음추천 시스템 - SVD뽑힌 100개의데이터들과 유저들의 rating값들로 SVD를 실행해서 유저가 좋아할만한 레시피 최종적으로 3개 추천 |

# 나. 추진 일정(월/주 단위의 세부적인 구현 일정 명시 요망)

|  |  |  |
| --- | --- | --- |
| **추진일정** | **추진계획** | **비고** |
| 11월 26일  ~ 12월 2일 | 앱 와이어프레임 제작 |  |
|  | 구현단계 세부화 및 ocr, python, openCV 학습 |  |
|  | 요리 레시피 데이터셋 전처리 , 준비 및 환경 구축 |  |
| 12월 3일  ~ 12월 9일 | 로그인/회원가입, 홈 화면 개발 |  |
|  | ocr활용하여 영수증 인식 기술 구현 |  |
|  | FER 오픈소스를 활용해 감정 인식 시스템 구현, 추천 시스템 구현 |  |
| 12월 3일  ~ 12월 9일 | 주요 기능 화면 개발 | 사용자 초기 정보 조사, 감정 인식/선택, 추천 레시피 및 영상 제공, 마이페이지 화면 |
|  | openCV활용하여 재료인식 시스템 구현 |  |
|  | 얼굴 인식 시스템 최적화, 추천 시스템 구현 |  |
|  |  |  |

# **3. 참고문헌**

|  |
| --- |
| <https://www.youtube.com/watch?v=rNxMwtottmM>  <https://tesseract-ocr.github.io/tessdoc/Installation.html>  <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07613578>  <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07613578>  <https://github.com/justinshenk/fer>  <https://github.com/justinshenk/fer>  만개의 레시피 데이터  <https://kadx.co.kr/opmk/frn/pmumkproductDetail/PMU_6d212747-8bd4-42c2-bc55-60bc636d6121/5>  페이지 크롤링  <https://otugi.tistory.com/393>  레시피 추천 예제 코드  <https://github.com/hye1ns/datanalysis_recipe> |