피부 개선 화장품 디스펜서

인공지능 기반의 개인화 피부 관리 솔루션

AI 기술을 활용한 피부 상태 분석
맞춤형 화장품 추천 및 디스펜싱
피부 상태 추적 및 효과 분석

목차

팀 구성

신승엽 팀장
팀장, PM (Project Manager)

김진형 팀원
GUI 제작, Server 제작

오민지 팀원
하드웨어 제작, 모터 기능 구현

윤치영 팀원
하드웨어 제작, UART 통신

황경태 팀원
Server 제작, AI 모델 제작

배경

피부 관리의 중요성 증가

현대 사회에서 피부 관리의 중요성이 점차 증가하고 있습니다. 건강한 피부 상태는 외모뿐만 아니라, 개인의 자존심과 사회적 이미지에 큰 영향을 미칩니다.

화장품 선택의 어려움

다양한 화장품 선택의 어려움이 발생하고 있습니다. 수많은 제품 중에서 자신의 피 부 상태와 맞는 제품을 찾는 것은 쉽지 않습니다. 이로 인해 많은 소비자가 제품 선택 에 많은 시간과 노력을 기울입니다.

초개인화 트렌드 부상

현대 뷰티 산업에서는 '초개인화(hyper-personalization)'가 중요한 트렌드로 부 상하고 있습니다. 이는 사용자의 개별적인 피부 상태와 필요에 맞춰 화장품을 추천 하고 제공하는 서비스의 필요성을 강조합니다.

피부 상태에 따른 제품 추천

목표

사용자 얼굴 사진 촬영 후 피부 상태 분석

AI 기술을 활용하여 사용자의 피부 상태를 정확하게 진단하고, 특성에 따른 분석 결과 제공

피부 상태에 따른 화장품 추천

분석된 피부 상태를 기반으로 사용자에게 최적화된 화장품을 추천하는 맞춤형 서비스 제공

디스펜서를 통한 화장품 체험

추천된 화장품을 디스펜서를 통해 적정량만큼 직접 체험할 수 있도록 하는 실질적인 경험 제공

피부 상태 기록 및 변화 추적

사용자의 피부 상태 변화를 지속적으로 기록하고 비교하여 개선 효과를 시각적으로 확인할 수 있는 기능 제공

이러한 목표들은 '초개인화(hyper-personalization)' 트렌드에 부합하며,
사용자의 개별적인 피부 상태와 필요에 맞춰 서비스를 제공합니다.

기능 설명 - 전체 흐름도

사용자 등록/로그인

디스펜서 사용을 위한
사용자 식별 과정

얼굴 사진 촬영

내장된 카메라를 통해
사용자 얼굴 사진 촬영

피부 상태 분석

AI 모델을 활용하여
피부 타입, 문제점 분석

피부 분석 결과 제공

분석된 피부 상 태
정보를 시각적으로 제공

데이터 관리

사용자 피부 데이터
및 사용 기록 저장

피부 상태 기록 및 추적

피부 상태 변화 추적적
기록 및 비교

화장품 추천

분석 결과에 따른
최적화된 화장품 추천

화장품 선택 및 디스펜싱

사용자가 추천된 화장품 선택
및 적정량 분사

기능 설명 - 피부 분석 프로세스

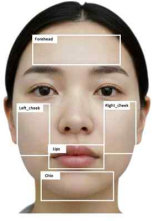
얼굴부위 디텍팅

사용자 얼굴 촬영

카메라 모듈을 통해 사용자 얼굴 사진을 촬영합니다.

이미지 처리-얼굴부위 디텍팅

촬영된 이미지를 전달 받아 얼굴 부위를 파악합니다.



피부 분석

이미지 처리-피부 분석

탐지된 피부를 부위별로 탄력도,수분,색소침착,모공 개수 등을 분석합니다.

분석 결과 제공

분석된 피부 상태 정보를 시각적으로 제공하고 저장합니다.

피부 상태 분석 세부 항목

이마, 왼쪽/오른쪽 볼, 턱, 입술 5개 부위

이마 : 수분, 탄력, 색소침착

볼 : 수분, 탄력, 색소침착, 모공 개수

입술 : 건조도

턱 : 수분, 탄력

하드웨어 구성

카메라 모듈

사용자 얼굴 사진 촬영 및 피부 상태 분석을 위한 핵심 장치

카메라로 얼굴 이미지를 촬영

디스펜서

화장품을 적정량만큼 분사하는 장치로, 모터 및 펌프 시스템

사용자에게 추천된 화장품을 펌프로 눌러 제공

제어 보드 (STM32)

라즈베리 파이 서버와 통신

모터 및 LED의 작동을 제어

디스플레이

사용자에게 정보(피부 분석 결과, 추천 화장품 등)를 시각적으로 제공.

터치스크린으로 사용자 인터페이스를 제공

라즈베리 파이

QT GUI 제공

PC와 HTTP 통신하여 이미지를 전송, 추론 결과 수신

STM32 보드와 UART통신하여 하드웨어 제어 명령 전달

컴퓨터 (서버)

라즈베리 파이로부터 이미지를 받아서 얼굴 부위 디텍팅 및 피부 상태 추론 후, 라즈베리 파이로 해당 결과 전송

데이터셋 소개

#헬스케어 건강서비스

한국인 피부상태 측정 데이터

분야 영상이미지 유형 텍스트, 이미지

구축년도 : 2023 갱신년월 : 2024-10 조회수 : 20,128 다운로드 : 924 용량 : 21.12 GB

다운로드 샘플 데이터

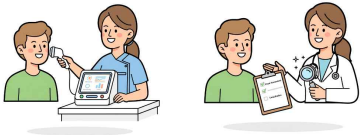
데이터셋 이미지 촬영 도구

스마트폰, 스마트패드, 디지털 카메라

데이터셋 메타데이터 구성

전문가 평가 : 입술 건조도, 주름, 색소침착(볼)

정밀기계 측정 : 탄력도, 수분, 모공, 색소침착 (이마)



얼굴 부위 Detecting 학습 과정

1. Yolov8 사용

Why?	Procedure	**Result**
➢ Framwork 많음 ➢ 개발 쉬움 ➢ 빠르고 가벼움	➢ Annotation Data Parsing ➢ 학습 진행	➢ 원하는 **정확도 도달 실패**

얼굴 부위 Detecting 학습 과정

2. PyTroch 사용

Why?

정확도가 높은 모델 사용

- PyTorch Framwork 필요

Procedure

모델 테스트

- SSD Llite
- FasterRCNN
- Retinanet

Result

- 학습 시간이 **오래 걸림**
- 볼 좌, 우 구분 어려움
- 원하는 **정확도 도달 실패**

얼굴 부위 Detecting 학습 과정

3. Intel Geti 이용

Why?

- Annotaion 과정 편리
- 정확도 높음 모델 제공

Procedure

- 라벨링 작업 수행
- MobilenetV2-ATSS 모델 학습
- YOLOX-TINY 모델 학습

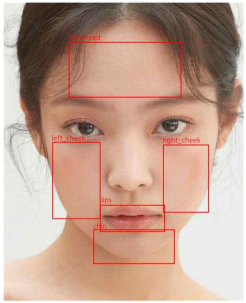
Result

- Geti를 활용한 MobilenetV2-ATSS모델 결정
- 학습 **시간이 단축**
- 학습 모델 **정확도 높음**

But, ONNX 표준 파일 변환 어려움

피부 상태 학습 과정

4. 커스텀 모델 사용



Why?

- 부위별 회귀와 분류 추론 필요

Procedure

- Input Data 증강
- Early Stop 구현
- 회귀 손실 및 MAE 계산 (Head별로)
- Backbone 모델 선택 학습
  - ResNet
  - Convnext_tiny
  - Convnext_large

Issue

- Backbon Model 무거움
- 무거운 모델 과적합 발생
- 학습 시간이 오래 걸림

Result

- Convnext_tiny 모델 사용
- 학습 모델 **정확도 높음**
- 학습 **속도 빠름**

사용자 인터페이스

1. Qt 프레임워크 선택 이유

VS

적은 리소스 사용

컴파일 언어의 네이티브 성능

최적화

카메라, V4L2 접근 용이

OpenCV 접근 용이

Hot Reload로 빠른 개발 속도

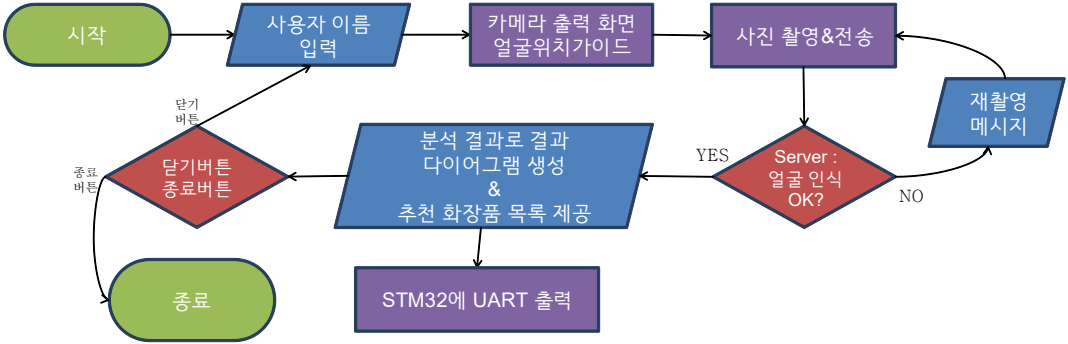
쉬운 개발 언어 :Dart

Material Design

비동기 처리, 네트워크 처리 용이

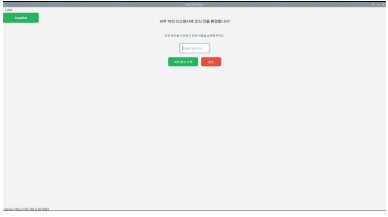
사용자 인터페이스

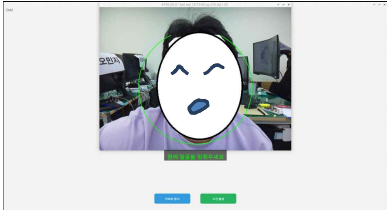
2. UI 설계 : 사용자 플로우 설계

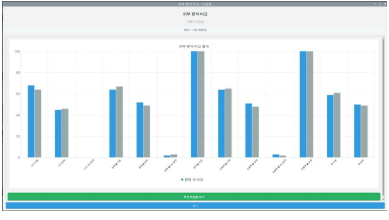


사용자 인터페이스

2. UI 설계 : 사용자 인터페이스







사용자 인터페이스

3. 카메라 창 가져오기. V4L2 실패 사례

Http 통신

1. 서버 통신 아키텍처

처리 흐름:

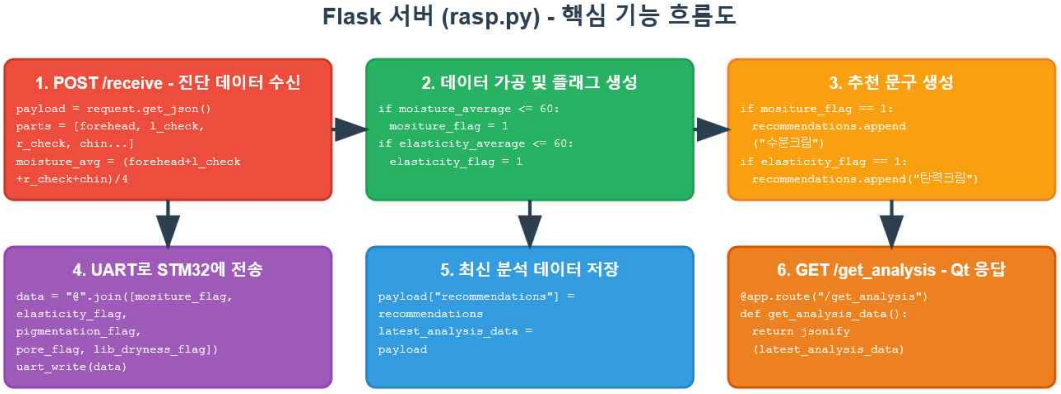
1. Qt에서 이미지 업로드 및 GUI/Database 관리
2. Server가 AI 진단 결과를 rasp.py로 전송
3. rasp.py가 결과를 가공하여 화장품 지정
4. rasp.py에서 진단 결과를 Qt로 전송
5. Qt는 진단 결과를 그래프로 시각화
6. UART로 STM32에 제공해야할 화장품 정보 전달

Http 통신


Flask 선택 이유
유연한 프레임워크
요구사항 변경에 쉽게 적응
빠른 프로토타이핑
간단한 구조로 API 서버 구축
파이썬 AI 생태계
파이썬의 AI 도구 및 라이브러리
API

Http 통신

1. 서버 통신 아키텍처



Http 통신

1. 서버 통신 아키텍처 : 수신 Json

raspberry pi 수신 json

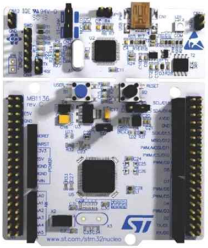
```
{
        "forehead": {
          "moisture": {0 ~ 100},
          "elasticity": {0 ~ 100},
          "pigmentation": {0 ~ 5},
        },
        "l_check": {
          "moisture": {0 ~ 100},
          "elasticity": {0 ~ 100},
          "pigmentation": {0 ~ 5}
          "pore": {0 ~ 5},
        },
        "r_check": {
          "moisture": {0 ~ 100},
          "elasticity": {0 ~ 100},
          "pigmentation":{0 ~ 5},
          "pore": {0 ~ 5},
        },
        "chin": {
          "moisture": {0 ~ 100},
          "elasticity": {0 ~ 100},
        },
        "lib": {0 ~ 4},
}
```

디스펜서 HW 제작 과정

1. 서버와 UART 통신

```
HAL_UARTEx_ReceiveToIdle_DMA(&huart1, rx_buf, RX_BUF_LEN);
```

```
void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
{
  if(huart->Instance==USART1){
      HAL_UART_Transmit(&huart2, rx_buf, Size, 100);
      osSemaphoreRelease(uartRxSemHandle);
      HAL_UARTEx_ReceiveToIdle_DMA(&huart1, rx_buf, RX_BUF_LEN);
  }
}
```



서버로부터 모델의 추론된 값을 수신하기 위해 UART Receive DMA 통신 적용

[DMA 방식 구현 이유]

1. 짧게 반복적인 문자를 전송 받는 상태는 아니지만 **CPU연산을 최소화**하기 위해 DMA를 사용
2. 단순히 DMA를 활용해 버퍼에 데이터를 채울 경우 **문자열의 끝**을 구분하기 **어려움** 그래서 **IDLE인터럽트**를 활용해 **문자열 끝을 판단**

디스펜서 HW 제작 과정

2. RTOS 구현

[freeRTOS 선택 이유]

1. 각 기능을 독립적으로 태스크 분리

→ 코드의 모듈화로 가독성, 유지보수성, 디버깅 용이성 향상

2. HAL_Delay() 대신 osDelay()를 사용해 CPU의 점유 방지

→ 시스템의 동시 반응성 확보 및 자원의 효율적 사용

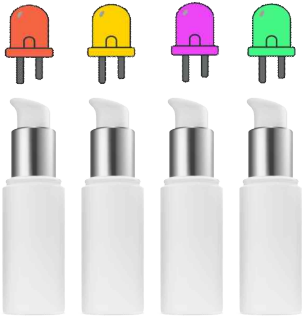
3. 세마포어를 사용해 태스크 간 실행 흐름을 제어

→ 복잡한 동작 순서를 플래그 변수 없이 명확하고 안정적으로 구현

디스펜서 HW 제작 과정

3. 디스펜서 구현

추천 화장품 LED표시



시스템 결과


SAMSUNG
탄력 개선 크림
수분 개선 크림
모공 개선 크림
미백 개선 크림
탄력 개선 크림
수분 개선 크림
모공 개선 크림
미백 개선 크림

기대 효과

값비싼 의료기기 부담 감소

전문 피부과 방문이나 고가의 의료기기 구매 없이도 개인 맞춤형 피부 진단 및 관리가 가능해집니다. 디스펜서를 통해 소비자들은 효과적인 피부 관리를 위한 전문적인 도구에 접근할 수 있습니다.

피부 상태 진단 맞춤 화장품으로 효과적인 피부 관리

개인의 피부 특성에 정확히 맞는 화장품을 사용하여 피부 관리의 효율성을 극대화할 수 있습니다. AI 기술을 통해 지속적인 피부 상태 모니터링 및 맞춤형 화장품 제공으로 피부 문제 개선에 기여합니다.

온라인/오프라인 매장에서 응용 가능

본 디스펜서는 가정용뿐만 아니라, 화장품 매장, 뷰티 살롱 등 다양한 온라인 및 오프라인 환경에서 고객 맞춤형 서비스를 제공하는 데 활용될 수 있습니다. 이는 새로운 비즈니스 모델 창출 및 고객 경험 향상에 기여할 것입니다.

Q&A

감사합니다!

궁금한 점이 더 있으시면 언제든지 물어보세요.