



프로젝트 명세서: 너의 색깔은?

퍼스널 컬러 진단 및 화장품 추천 키오스크

1. 프로젝트 개요

목표:

사용자의 퍼스널 컬러를 진단하여 맞춤형 화장품을 추천하고, 실제 얼굴 이미지에 화장품 색조를 합성해보는 체험형 키오스크 시스템 구축

주요 기능:

- 🌈 퍼스널 컬러 진단
- 🧴 퍼스널 컬러 기반 화장품 추천
- 🤖 화장품 색조 합성 및 테스트

2. 핵심 기능 상세

2.1 🌈 퍼스널 컬러 진단

진단 분류 (총 12가지)

- 봄: 라이트, 브라이트, 비비드
- 여름: 뮤트, 라이트, 브라이트
- 가을: 뮤트, 스트롱, 다크
- 겨울: 비비드, 스트롱, 다크

2.1.1 1차 검증: MobileNetV2 활용 이미지 분류

- 데이터 준비: 각 컬러별 100장의 얼굴 이미지 수집
- 데이터 증강: 이미지 증강을 통해 각 컬러별 이미지를 10,000장으로 확장
- 모델 학습:
 - TensorFlow의 MobileNetV2 모델 사용

- 총 120,000장의 증강 데이터로 학습
- **예측 단계:** 사용자 얼굴 이미지를 입력받아 12가지 컬러 중 하나로 분류

2.1.2 2차 검증: OpenAI API 활용

- 1차 검증 결과를 OpenAI API로 전송하여 보조 검증 수행
 - 모델 예측 결과의 신뢰도 향상 및 추가적인 해석 제공
 - 최종 퍼스널 컬러 결과 도출
-

2.2 🚪 퍼스널 컬러 기반 화장품 추천

- **추천 대상:** 립, 아이, 치크 제품
 - **추천 프로세스:**
 1. 사용자 퍼스널 컬러 진단 결과 수신
 2. 데이터베이스에서 해당 컬러에 최적화된 제품 검색
 3. 사용자에게 맞춤형 화장품 추천 리스트 제공
-

2.3 🤖👤 화장품 색조 합성 및 테스트

- **기능:** 사용자의 얼굴 이미지에 화장품 색조를 합성하여 시뮬레이션 제공
 - **기술 선택:** face-api.js 또는 MediaPipe 활용
 - **합성 프로세스:**
 1. 얼굴 인식 및 랜드마크 검출
 2. 선택한 제품 색상을 해당 부위(립, 아이, 치크)에 자연스럽게 적용
 3. 실시간 합성 결과 사용자에게 시각화
-

3. 시스템 아키텍처 🌐

- **프론트엔드:** React.js를 이용한 직관적 UI 및 사용자 인터랙션
 - **백엔드:** Spring Boot로 REST API 제공, 비즈니스 로직 처리
 - **머신러닝 서비스:** Python/TensorFlow로 퍼스널 컬러 예측 수행
 - **데이터베이스:** MySQL 사용
 - **클라우드:** AWS 환경에서 모든 서비스 호스팅
-

4. 기술 스택 및 구현 세부사항

기술 스택

- 프론트엔드: React.js, HTML5, JavaScript
- 백엔드: Spring Boot, RESTful API, Java
- 데이터베이스: MySQL
- 머신러닝: Python, TensorFlow (MobileNetV2), 데이터 증강 라이브러리
- 얼굴 합성: face-api.js 또는 MediaPipe
- 클라우드: AWS (EC2) , Docker

4.1 퍼스널 컬러 진단 모듈 구현

1. 데이터 전처리 & 증강

- 이미지 정규화, 크기 조정, 회전, 이동, 밝기 조절 등

2. 모델 학습

- MobileNetV2로 12개 클래스 분류 모델 학습
- Transfer Learning 및 fine-tuning 적용

3. 1차 예측 & 2차 검증

- 사용자 이미지 업로드 → MobileNetV2 예측 → OpenAI API 검증
- 최종 퍼스널 컬러 도출 및 사용자에게 결과 제공

4.2 화장품 추천 모듈 구현

- 데이터베이스에서 컬러 매칭 제품 조회
- 추천 알고리즘으로 사용자 맞춤형 화장품 리스트 생성
- 추천 결과를 프론트엔드에 전달하여 사용자에게 보여줌

4.3 화장품 색조 합성 모듈 구현

1. 얼굴 인식: face-api.js/MediaPipe로 얼굴 특징점 검출
2. 색조 합성: 검출된 랜드마크에 선택한 색상 자연스럽게 적용
3. 실시간 시뮬레이션: 얼굴에 실시간으로 색조 화장품을 합성

5. 데이터베이스 설계 (MySQL)

```
-- Database initialization script for personal color and prod

-- Create database
CREATE DATABASE IF NOT EXISTS personal_color_db;
USE personal_color_db;

-- Create 'personal_color' table
CREATE TABLE personal_color (
    personal_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);

-- Create 'products' table
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(255) NOT NULL,
    category VARCHAR(50) NOT NULL,
    brand VARCHAR(100) NOT NULL,
    product_detail_name VARCHAR(255) NOT NULL,
    price INT NOT NULL
);

-- Create 'product_color' table
CREATE TABLE product_color (
    product_id INT,
    color VARCHAR(7) NOT NULL,
    r INT NOT NULL,
    g INT NOT NULL,
    b INT NOT NULL,
    PRIMARY KEY (product_id, color),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

-- Create 'product_personal' table
CREATE TABLE product_personal (
    product_id INT,
```

```
personal_id INT,  
PRIMARY KEY (product_id, personal_id),  
FOREIGN KEY (product_id) REFERENCES products(product_id) (  
FOREIGN KEY (personal_id) REFERENCES personal_color(perso  
);
```

6. API 설계 📡

주요 엔드포인트 (가짜)

- **POST /api/diagnose**
설명: 얼굴 이미지 업로드 및 퍼스널 컬러 진단
입력: 이미지 파일, 사용자 메타데이터
출력: 퍼스널 컬러 결과, 신뢰도, 설명
- **GET /api/recommendations/{userId}**
설명: 사용자 맞춤 화장품 추천 조회
출력: 추천 제품 리스트
- **POST /api/simulate**
설명: 화장품 색조 합성 요청
입력: 얼굴 이미지, 제품 선택 정보
출력: 합성된 이미지 URL 또는 데이터

7. 결론 ✨

"너의 색깔은?" 프로젝트는 최신 머신러닝과 웹 기술을 활용하여 사용자에게 개인 맞춤형 퍼스널 컬러 진단, 화장품 추천, 그리고 실제 적용 시뮬레이션을 제공합니다. 아름답고 직관적인 UI와 정교한 알고리즘을 통해 사용자 경험을 극대화하며, 지속적인 피드백과 개선을 통해 서비스 품질을 높여나갈 예정입니다.