

AI-Powered Personal Beauty Advisor

2025 전기 졸업과제 착수보고서 <Underdog>



부산대학교 정보컴퓨터공학부

제출일	2025. 05. 16	분과명	소프트웨어/인공지능
팀장	202155659 Han Nwae Nyein	팀명	Underdog
팀원	202155658 Nyi Nyi Htun 202055594 전재원	지도교수	전상률

목차

1. 배경 및 동기
 2. 과제의 목표
 3. 요구사항 분석서
 4. 개발환경 및 사용기술
 5. 현실적 제약 사항 분석 및 대책
 6. 설계 예시
 7. 추진 체계 및 일정
 8. 구성원 역할
 9. 참고문헌
-

1. 배경 및 동기

현대 사회에서 패션과 뷰티는 단순한 외모 꾸미기를 넘어 자기 표현의 핵심 수단으로 진화하였으며, 헤어 컬러, 렌즈, 메이크업 등 변경 가능한 요소를 통해 개인의 정체성을 자유롭게 표현하는 추세입니다. 그러나 기존 퍼스널 컬러 진단 시스템은 머리색, 눈동자 색 등 고정된 신체적 특징에 의존하거나, 오프라인 전문가의 주관적 판단에 기반하여 다음과 같은 한계를 노출시켰습니다:

- **접근성 문제:** 진단을 위해 전문가와의 대면 상담이 필수적이며, 시간과 비용이 과도하게 소요됩니다.
- **유연성 부재:** 사용자가 렌즈, 헤어 컬러, 메이크업 등으로 스타일을 변화시킬 경우, 기존 진단 결과가 실효성을 잃습니다.
- **후속 불만족:** 오프라인 진단 후 추천 색상으로 스타일링했음에도 예상과 다른 결과가 빈번히 발생합니다.

이에 본 프로젝트는 피부톤에 집중한 AI 기반 퍼스널 컬러 시스템을 제안합니다.

피부톤은 개인의 타고난 특성이자 모든 스타일링의 기반이 되며, 피부의 밝기, 색조, 채도를 Lab 색 공간을 기준으로 분석하여 진정한 피부톤을 바탕으로 사용자에게 가장 잘 어울리는 렌즈, 헤어 및 액세서리 색상을 제안하는 시스템을 개발하고자 합니다.

컴퓨터 비전과 딥러닝 기술을 활용하여 다음과 같은 혁신을 목표로 합니다:

2. 과제의 목표

- **정확한 피부 톤 분석:** 피부의 밝기, 색조, 채도를 Lab 색 공간 기반으로 하여, 진정한 피부 톤을 바탕으로 퍼스널 컬러를 정확히 분석할 수 있는 모델 개발.

- **개인 맞춤형 스타일링 제안:** 추출된 피부톤 정보를 활용하여 사용자의 고유한 스타일에 맞는 컬러를 제안.
- **실제감 있는 시각화:** 진단된 퍼스널 컬러를 바탕으로 렌즈, 헤어, 립스틱 등의 색상을 적용한 AI 기반의 시각적 결과를 제공.
- **사용자 친화적 인터페이스:** 비전문가도 쉽게 접근할 수 있는 직관적이고 간단한 사용자 인터페이스를 제공.

이 프로젝트를 통해 단순히 색상을 추천하는 것을 넘어, 데이터 기반의 객관성과 사용자 중심의 유연성을 결합함으로써 기존 서비스의 한계를 극복할 것입니다. 이를 통해 누구나 시간과 장소의 제약 없이 자신에게 최적화된 스타일을 발견하고, 지속적인 변화를 통해 개성을 탐구할 수 있는 플랫폼을 구축하고자 합니다.

3. 요구사항 분석서

구분	요구사항	설명/고려사항
기능적 요구사항	사용자 이미지 업로드	-사용자는 얼굴 이미지(정면)을 업로드해야 함
	얼굴 랜드마크 추출	-MediaPipe 를 통해 눈, 입술, 피부 패치 추출
	이미지 전처리	-OpenCV 및 Albumentations 로 밝기/채도 보정
	피부톤 클러스터링	K-Means 로 주요 피부톤 추출
	퍼스널컬러 분류	-사전 학습된 PyTorch 모델로 분류

	진단 결과 시각화	-색상 팔레트, 톤 설명, 추천 스타일 시각화(Streamlit 차트, 이미지 등)
	대안 시뮬레이션	-Stable Diffusion or ControlNet 을 통한 메이크업/헤어 시뮬 이미지 생성
	결과 다운로드	-분석 리포트를 PDF/이미지 형태로 저장
비기능 요구사항	실시간 또는 빠른 응답 시간	-모델 추론 및 필터링 속도 최적화 (ONNX or PyTorch + GPU 활용 가능)
	경량화된 모델 로딩	-Streamlit 환경에서 로딩 가능한 수준으로 모델 저장(TorchScript, ONNX 등)
	모바일/웹 브라우저 호환	-Streamlit 웹 기반 UI 는 모바일에서도 기본 대응 가능(단 해상도 최적화 필요)
	개인정보 보호	-사용자 이미지는 처리 후 즉시 삭제 또는 로컬 저장만 허용
	직관적인 UI 구성	-사이드바를 통한 설정, 버튼 기반 인터랙션, 설명 툴 팁 제공
	오류 대응 및 안내 메시지	-이미지 업로드 실패, 얼굴 감지 실패 시 사용자 피드백 제공
환경 요건	필수 라이브러리	- Streamlit, OpenCV, MediaPipe, scikit-learn, PyTorch/TensorFlow, matplotlib 등

4. 개발환경 및 사용기술

4.1 개발환경

4.1.1 프론트엔드 (Frontend)

- **React.js:** 사용자 인터페이스(UI) 구현을 위한 빠르고 확장 가능한 자바스크립트 프레임워크.
- **Tailwind CSS:** 모던하고 반응형 디자인을 위한 CSS 프레임워크.
- **Fabric.js** 또는 **Konva.js:** 실시간 이미지 렌더링 및 색상 오버레이를 위한 캔버스 라이브러리.
- **JavaScript (ES6+):** 인터랙티브 기능과 API 통신을 위한 기본 스크립트 언어.

4.1.2 백엔드 (Backend)

- **Node.js (Express.js):** 고성능 API 서버 구현을 위한 자바스크립트 런타임 환경.
- **Flask** 또는 **FastAPI:** 빠르고 간단한 Python 기반 API 서버 옵션.
- **Streamlit:** 백엔드에서 실행되는 프론트엔드 역할 도구, 머신러닝 모델을 웹에서 시각화/제어하는 데 최적화된 Python 라이브러리
- **WebSockets:** 실시간 데이터 통신을 위한 양방향 통신 프로토콜 (선택 사항).

4.1.3 머신러닝 (Machine Learning)

- **TensorFlow.js:** 브라우저에서의 모델 로딩 및 실시간 추론.
- **scikit-learn:** 데이터 분석, 클러스터링, 모델 평가.
- **K-Means Clustering:** 피부톤 분류 모델.
- **ONNX.js:** PyTorch 모델의 웹 변환을 위한 경량 라이브러리 (선택 사항).

4.1.4 데이터베이스 (Database)

- **Firebase Firestore:** 실시간 데이터 동기화 및 사용자 데이터 저장.
- **MongoDB:** 구조적 데이터 저장을 위한 NoSQL 데이터베이스.
- **LocalStorage:** 간단한 사용자 세션 데이터 저장.

4.1.5 이미지 처리 (Image Processing)

- **OpenCV.js:** 실시간 이미지 필터링, 패치 추출, 얼굴 인식.
- **MediaPipe:** 얼굴 랜드마크 추출 및 이미지 전처리.
- **Dlib:** 추가적인 얼굴 랜드마크와 특징점 추출.

4.2 사용기술

4.2.1 피부톤 분류 모델 개발

- **K-Means 클러스터링 모델 개발:**
 - 사용자의 피부톤을 분석하여 **개인 맞춤형** 색상 추천이 가능하도록 **K-Means** 클러스터링 모델을 개발.
 - 다양한 **LAB 색상 데이터**를 활용하여 **피부 색조**를 정확하게 분류.
 - **머리색, 눈동자, 립스틱** 등 **변경 가능한 요소**를 제외하고 **피부 색상**을 기준으로 퍼스널 컬러를 진단.
- **모델 학습 데이터 준비:**
 - 실제 피부톤 샘플, 합성 데이터, 다양한 조명 조건에서의 데이터 포함.
 - 색상 범위를 자유롭게 설정하여 **고유한 색상 분류 체계** 설계.

Approximate LAB Ranges for 12 Types (Example)

Skin Type	L (Lightness)	A (Green-Red)	B (Blue-Yellow)
Light Spring	75-90	5-15	20-30
Warm Spring	60-80	10-20	20-40
Bright Spring	70-85	10-25	25-40
Soft Summer	70-85	-5 to 10	0-15
Cool Summer	75-90	-5 to 15	0-15
Light Summer	75-90	0-15	0-20
Warm Autumn	50-75	15-30	30-50
Deep Autumn	40-65	20-30	30-50
Soft Autumn	55-75	15-25	20-40
Deep Winter	20-65	5-25	0-20
Cool Winter	30-80	0-15	0-20
Bright Winter	50-80	10-25	5-25

4.2.2 이미지 처리 및 색상 추출 기능

- **피부 영역 추출:**
 - 이마, 양 볼, 턱 부위의 피부 패치를 선택하여 평균 색상을 계산.
 - **MediaPipe** 또는 **OpenCV** 를 활용하여 **정확한 얼굴 영역** 추출.
 - RGB → LAB 으로 변환하여 피부톤 추출.

4.2.3 색상 추천 및 시각화 기능

- **렌즈, 립스틱, 헤어 색상 추천:**
 - 분석된 피부톤에 맞는 **렌즈, 립스틱, 헤어** 색상을 제안.
 - 색상 추천 시 **명도, 채도, 색조**를 고려하여 다양한 옵션 제공.
- **실시간 색상 시각화:**
 - 선택한 색상을 사용자 이미지에 실시간으로 반영하여 **직관적인 피드백** 제공.
 - **Canvas API** 또는 **Fabric.js** 를 사용하여 색상 변경 효과 구현.
- **랜덤 색상 변환 기능:**
 - 다양한 스타일링 옵션을 위해 랜덤 색상 버튼 제공.

4.2.4 웹 애플리케이션 통합 및 배포

- **프론트엔드:** **React.js** 또는 **Next.js** 를 사용한 UI/UX 설계.
- **백엔드:** **Node.js, Flask** 또는 **FastAPI** 를 통한 데이터 처리 및 모델 배포.
- **데이터베이스:** **Firebase** 또는 **MongoDB** 를 활용한 사용자 데이터 관리.
- **실시간 통신:** 빠른 응답성을 위한 **WebSocket** 또는 **REST API** 사용.
- **보안:** 이미지 처리 시 개인정보 보호를 위한 **HTTPS** 암호화.

5. 현실적 제약 사항 분석 및 대책

구분	문제점	대책
다양한 조명 환경	조명 조건에 따라 피부 색상이 왜곡되어 정확한 진단이 어려움	-다중 이미지 기반 평균 색상 분석 -Albumentations 등을 활용한 조명 보정 데이터 증강
얼굴 인식 및 피부 패치 추출	얼굴 랜드마크 인식이 부정확할 경우 피부 영역 추출 오류 발생 가능	-MediaPipe + Dlib 을 함께 활용해서 랜드마크 정확도 향상 -주변 픽셀 평균을 통한 패치 색상 보정
웹 환경의 모델 성능 저하	브라우저에서 모델 로딩 속도 느림, 실시간 처리 어려움	- TensorFlow.js 기반 경량 모델 사용 -Lazy Loading 및 Web Worker 활용 -사전 훈련 모델 최적화
개인정보 및 데이터 보안	사용자 이미지 저장 시 개인정보 유출 우려	-HTTPS 기반 암호화 사용 -익명화 및 최소 데이터 수집 원칙 준수

6. 설계 예시

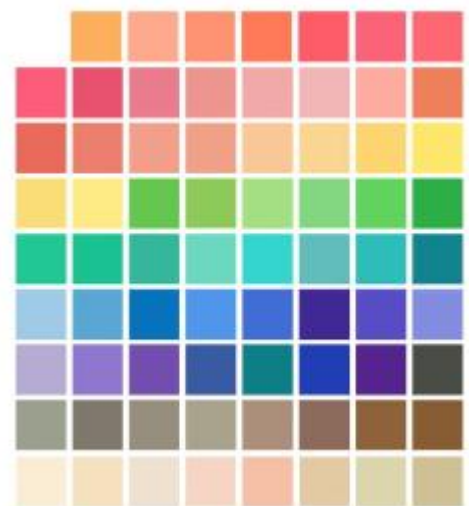
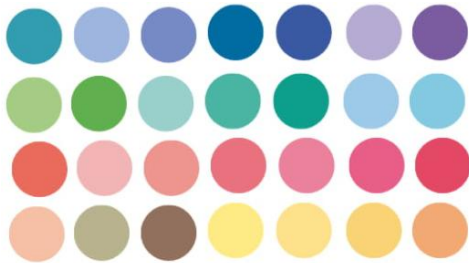
Upload Pic



Analysis Result – LIGHT SPRING



Recommendations for Light Spring



7. 추진 체계 및 일정

[illegible]

8. 역할 분담

이름	담당
Han Nwae Nyein	<ul style="list-style-type: none">- 색채 이론 연구 및 학습- 색상 분석 모델 개발- 디버깅 및 최적화
전재원	<ul style="list-style-type: none">- UI/UX 디자인- 색상 오버레이 로직- 인터랙티브 기능 구현
Nyi Nyi Htun	<ul style="list-style-type: none">- 백엔드 개발- 이미지 세그멘테이션- 데이터베이스 설정
공통	<ul style="list-style-type: none">- 프로젝트 문서화- 통합 테스트

9. 참고문헌

<https://gabriellearruda.com/seasonal-color-analysis-ultimate-guide/>

<https://www.perfectcorp.com/consumer/apps/ymk>