

얼굴형 분석 기반 헤어스타일링 추천 서비스

김대웅¹, 김동현¹, 김명현¹, 소민준¹, 한태빈¹, 김영중²

¹ 숭실대학교 소프트웨어학부 학부생

² 숭실대학교 소프트웨어학부 교수

8330kdw@soongsil.ac.kr, znfnfns0365c@gmail.com, devmhyun@soongsil.ac.kr, somongi@naver.com, taebin325@gmail.com, youngjong@ssu.ac.kr

Hair Styling Recommendation Service Based on Face Shape Analysis

Daewoong Kim¹, Dongheon Kim¹, Myeonghyun Kim¹, Minjun So¹, Taebin Han¹,
Youngjong Kim²

^{1,2}School of Software, Soongsil University (Undergraduate Student, Professor)

요 약

Personal color 진단, MBTI 등 ‘나’라는 개인에 대한 관심이 커져 개인화 되어가는 사회에서 헤어스타일의 관심도 높다. 디자이너에게 컨설팅을 받고자 할 때 소모되는 경제적, 시간적, 물리적 자원 소모를 해소하기 위해 헤어스타일을 추천해주고, 실제 이미지로 보여줌으로써 큰 자원 소모를 해소하는 서비스를 AI 기반 기술과 프레임워크 순서로 제안 및 소개한다.

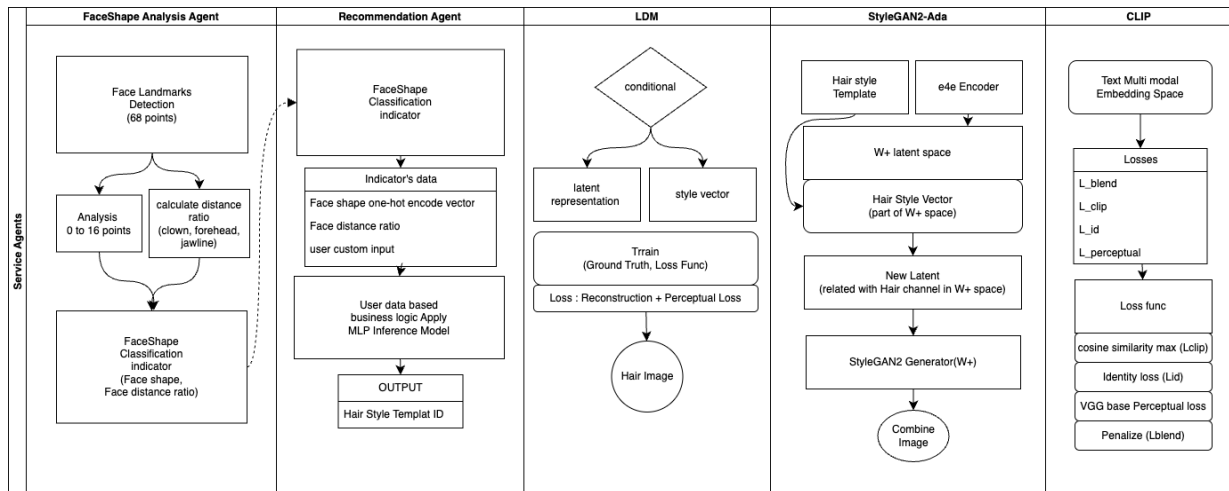


Fig. 1. Service's AI Agents

1. 서론

경제 수준과 문화 생활, 여가 생활의 다양성과 문화가 발달한 요즘 personal color 진단과 같은 개인에 맞춘 서비스를 제공하도록 사회상이 더욱이 발전하고 있다. 또, 집에서 간단한 휴대폰을 통해 성격을 진단하는 등 ‘나’에 대한 관심이 커지는 요즘, 디자이너를 통해 얼굴형을 정확히 진단하고 헤어스타일을 추천받는 것은 경제, 시간, 물리적 자원 소모가 큰 활동이다. 이를 해소하기 위하여, 집에서 휴대폰으로 간단하게 얼굴형을 진단 받고 직접 헤어스타일을 적용할 수 있는 서비스가 필요하다. 이에 본 논문에서는 얼굴형 분석 기반 헤어스타일링 추천 서비스를 제안한다.

2. AI Agents

Fig. 1. 은 본 논문에서 제안하는 시스템의 AI Agents 가 수행하는 얼굴형 분석과, 추천 및 생성 과정이다.

2-1. AI 기반 얼굴형 분석

본 시스템에서는 Face Landmark 를 이용하여 둥근형, 긴형, 계란형, 사각형, 하트형, 역삼각형 총 6 개로 얼굴형을 분류한다. 분류를 위해 300W 벤치마크에서 SOTA 성능을 보이는 D-ViT 모델을 사용한다. 본 연구에서는 사전 학습된 D-ViT 모델을 기반으로 사용자

의 얼굴 이미지를 입력 받아 얼굴 랜드마크 68 개의 좌표를 추출한다. 추출된 특징은 텍 선 라인(0~16 번 포인트)을 기반으로 얼굴의 윤곽선 곡률을 분석하고 광대, 이마, 턱선간의 거리 비를 통하여 얼굴형 분류 지표를 생성하도록 한다[1].

위 과정을 OpenCV 및 Mediapipe 를 기반으로 전처리 하여 PyTorch 프레임워크 위에서 실시간 추론이 가능하도록 구현해 경량화된 ONNX 모델을 모바일 환경에 배포한다.

2-2. Recommendation and Generator AI

본 Agent 는 얼굴형 분석을 통해 얻은 데이터 및 사용자 입력 정보를 기반으로 사용자 맞춤형 헤어스타일을 추천하고 헤어스타일이 적용된 이미지를 생성한다.

전문가 레이블링 기반의 얼굴형, 헤어스타일링 집합 Dataset 을 기반으로 학습된 Recommendation Agent 는 one-hot 인코딩 벡터 형식의 얼굴형과 얼굴 비율 파라미터, 사용자 입력 정보를 기반으로 비즈니스 로직이 적용된 MLP model 추론 파이프라인을 거쳐 헤어스타일 템플릿 ID 를 반환한다.

Generator Agent 는 헤어스타일 템플릿 ID 를 기반으로 Src Image 를 찾아 Dst Image 를 생성하는 Style Transfer 기반 이미지 합성을 수행한다. LDM 을 기반으로 가상의 헤어를 생성하거나, Dataset 에 저장된 헤어 이미지를 바탕으로 StyleGAN2-Ada 기반 Style Embedding Transfer 를 통해 기존 얼굴은 유지하며 헤어스타일만 바뀐 이미지를 생성한다. 이후 CLIP 과 Style Matching Loss 를 바탕으로 이미지를 자연스럽게 하는 과정을 거친다[2, 3].

본 논문에서 설계한 Loss 는 다음과 같다.

$$L_{total} = \lambda_1 * L_{clip} + \lambda_2 * L_{id} + \lambda_3 * L_{perceptual} + \lambda_4 * L_{blend}$$

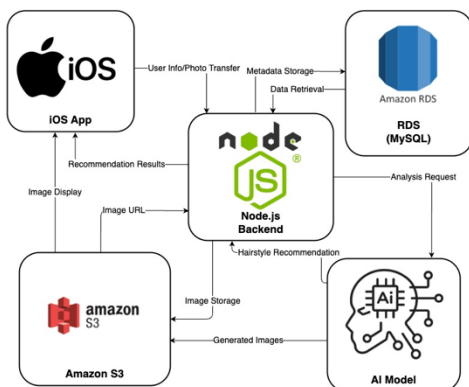


Fig. 2. Service Framework

3. 서비스 프레임워크

Fig. 2. 은 본 논문에서 제안하는 서비스의 전반적인 프레임워크 구조도를 나타낸 그림이다.

3-1. iOS App

iOS 앱은 사용자 인터페이스를 제공하며, 회원 가입, 정보 입력, 사진 촬영 및 결과 표시 기능을 담당

한다. 사용자가 입력한 헤어 관련 정보와 얼굴 사진은 User Info/Photo Transfer 과정을 통해 Node.js 백엔드로 전송된다.

3-2. Backend

Backend(Node.js)는 Express 기반 RESTful API 서버로, Amazon EC2 에서 운영되며, 데이터 처리 및 저장, 이미지 저장 관리를 담당한다. 또한 AI 모델과의 통신을 통해 얼굴 분석 및 헤어스타일 추천 요청(Analysis Request)을 처리하고 결과를 수신하여 사용자에게 전달한다.

3-3. Database

서버로부터 전송된 이미지는 Image Storage 과정을 통해 Amazon S3 에 저장되고, 사용자 관련 메타데이터는 MySQL 을 통해 Amazon RDS 에 저장된다. Amazon S3 는 사용자 사진 및 AI 가 생성한 이미지를 저장하며, 서버는 Image URL 을 통해 접근한다. 오브젝트 스토리지를 통해 저장 효율성과 유지보수 편의성을 확보하였다.

3-4. AI model

AI 모델은 얼굴 이미지를 분석해 얼굴형을 파악하고 적합한 헤어스타일을 추천한다. 추천 결과는 Hairstyle Recommendation 을 통해 백엔드로 전달되며, 생성된 스타일 이미지는 Generated Images 과정을 통해 S3 에 저장된다. 또한 Generator AI 를 통해 사용자가 원하는 디테일을 반영한 헤어스타일 적용 서비스도 제공한다.

4. 결론

본 논문은 AI Agent 를 기반으로 사용자의 얼굴형을 분석하고, 그에 맞는 헤어스타일을 추천하여 생성하는 서비스를 제안한다. 본 논문에서 제안한 서비스는 현재 개인화 되어가는 사회상에 맞춘 수요성 있는 서비스로, 헤어 컨설팅으로 쓰는 자원 소모를 줄일 수 있음을 알 수 있다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업의 지원을 받아 수행되었음 (2024-0-00071)

참고문헌

- [1] Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., ... & Feng, J. (2021). Deepvit: Towards deeper vision transformer. arXiv preprint arXiv:2103.11886.
- [2] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. Advances in neural information processing systems, 33, 12104-12114.
- [3] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).