


캡스톤 디자인

종합설계 프로젝트

프로젝트 명	HAI!
팀 명	무투మ్
문서 제목	결과보고서

Version	1.5
Date	2021-05-26

팀원	오규석 (조장)
	양성민
	유선종
	차윤성
	최나라

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


1.1 CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "HAI"를 수행하는 팀 "무투멍"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "무투멍"의 팀원들의 서면 허락 없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	결과보고서-HAI!.doc
원안작성자	유선종
수정작업자	양성민, 차윤성, 최나라, 오규석

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2021-05-25	유선종	1.0	최초 작성	연구/개발 내용 및 결과물
2021-05-26	차윤성	1.1	내용 추가	시스템 구조 및 설계도
2021-05-26	양성민	1.2	내용 수정	사용자 매뉴얼, 활용/개발된 기술
2021-05-26	최나라	1.3	내용 수정	프로젝트 개요 및 필요성, 목표 수정
2021-05-26	오규석	1.4	내용 수정	활용/개발된 기술 추가, 자기 평가 작성
2021-05-26	전원	1.5	최종 검수	

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	5
2	개발 내용 및 결과물	6
2.1	목표	6
2.2	연구/개발 내용 및 결과물	7
2.2.1	연구/개발 내용	7
2.2.2	시스템 기능 요구 사항	30
2.2.3	시스템 비기능(품질) 요구 사항	37
2.2.4	시스템 기능 및 구조 설계도	38
2.2.5	활용/개발된 기술	39
2.2.6	현실적 제한 요소 및 그 해결 방안	43
2.2.7	결과물 목록	44
2.3	기대효과 및 활용방안	44
3	자기평가	45
4	참고 문헌	46
5	부록	46
5.1	사용자 매뉴얼	49
5.2	배포 가이드	56

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

1 개요

1.1 프로젝트 개요




[그림1] 로고

본 프로젝트 'HAI!'는 사용자의 얼굴을 인식하고, 직접 개발한 AI 모델을 통해 사용자의 얼굴형을 분석한다. 최종적으로 AI 모델이 판단한 사용자의 얼굴형에 따라 어울리는 헤어스타일을 추천해 주는 웹 서비스(Web service) 개발을 목표로 한다.

프로젝트 이름인 'HAI!'는 'Hair'와 'AI'를 합성하여 만든 단어로서 본 프로젝트의 목표인 '헤어스타일 추천하는 AI'의 의미를 담고 있다. 또한 프로젝트 이름인 'HAI'는 'Hi'와 비슷한 발음을 가진다. 산학 기업인 '로보링크'에서 'AI를 체험할 수 있는 서비스'를 개발하는 것을 요구했고, AI를 체험하기 위해 AI에게 인사를 하는 의미도 내포하고 있다.

본 프로젝트는 성별에 따른 대중적인 헤어스타일을 추천하기 위해 사용자가 자신의 성별을 선택을 하며 서비스가 시작된다. 사용자는 '사진 업로드 기능'과 '실시간 Web-Cam을 이용한 자동 캡처 기능' 중에 하나를 이용하여 자신의 얼굴 사진을 제공한다. Web-Cam을 이용하는 경우, 사용자의 얼굴을 인식하여 사용자가 정면을 바라보도록 안내한다. 사용자가 정면을 바라봤을 때 자동으로 캡처를 한다. AI 모델을 통해 사용자의 얼굴형을 판단하고, 판단한 얼굴형을 사용자에게 제시한다. 사용자의 성별과 얼굴형을 기준으로 사용자에게 어울리는 헤어스타일을 추천한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26

1.2 추진 배경 및 필요성

본 프로젝트는 산학 협력 프로젝트로서, 산학 협력 기업인 '로보링크'의 요구 사항에 맞춰서 개발을 진행했다. 산학 기업의 요구 사항은 'AI를 흥미롭게 체험할 수 있는 서비스'를 개발하는 것과 'Object Detection을 이용한 서비스'를 개발하는 것이었다. 구체적인 주제는 팀 내부에서 자체적인 회의를 거친 뒤에 결정해달라고 했다. 우리는 평소에 겪는 불편함을 해소해 주는 서비스였으면 좋겠다고 생각해서 본 프로젝트를 기획했다. 프로젝트의 주제를 기획한 배경은 다음과 같다.


현 시대에는 인간의 외적인 모습을 아름답게 디자인하는 분야인 패션(Fashion), 메이크업(Make-up), 이미지 메이킹(Image making) 등 모든 분야에서 얼굴 유형의 특성을 다루고 있다. 특히 헤어 분야를 연구하여 발표된 여러 학문서적에서는 각각의 얼굴유형을 분석하고, 분석된 얼굴유형의 특성을 보완하는 방법까지 다방면으로 제시하고 있다.

(출처 : Tak, W.-W., & Kim, P.-J. (2015). 얼굴형에 따른 헤어스타일연구. *The Korean Journal of Food & Health Convergence*, 1(1), 1-20. <https://doi.org/10.13106/KJFHC.2015.VOL1.NO1.1>)

개성과 취향의 시대인 만큼 우리는 다양한 헤어스타일을 접하고 이를 시도하려고 한다. 하지만 자신에게 어울리는 헤어스타일을 찾기란 쉽지 않다. 그렇다면 헤어 디자이너와 헤어 컨설턴트는 고객에게 어떤 기준으로 헤어스타일을 추천할까? 여러 논문과 전문가들의 의견을 조사한 결과, 헤어 디자이너와 헤어 컨설턴트는 고객의 얼굴형을 분석하고, 분석된 얼굴형의 특성을 보완하는 방법으로 헤어스타일을 추천한다.

그러나 대부분의 사람들은 자신의 얼굴형에 대해 정확히 인지하지 못하고, 얼굴형의 특성을 보완하는 헤어스타일이 무엇인지 모르는 경우도 있다. 심지어 헤어를 시술하지만 직접적으로 디자인에 관여를 하지 않는 스태프들은 얼굴형에 따라 헤어스타일의 결과가 달라질 수 있다는 것을 인지하는 경우가 드물었다. 또한 전문가들도 얼굴형을 육안으로 정확하게 판단하는데 어려움이 있고, 분류하는 방법이 서로 다르다.

이와 같은 사람들의 고민과 어려움을 해결하고자 본 프로젝트를 기획했다. 본 프로젝트를 통해 사람들이 자신의 얼굴형을 정확하게 인지하고, 얼굴형을 보완하는 헤어스타일을 시도할 수 있을 것이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

2 개발 내용 및 결과물


2.1 목표

본 프로젝트는 산학 협력 기업의 요구 사항과 가이드라인을 준수하며 주요 기능들을 개발하는 것을 목표로 하고 있다.

본 프로젝트에서 제공하는 서비스의 목표는 사용자의 얼굴형을 판단하여 얼굴형에 따라 어울리는 헤어스타일을 추천해 주는 AI 기반 웹 서비스(Web-Service)를 개발하는 것이다. 이때, 사용자의 얼굴을 인식하고 개발한 AI 모델을 토대로 사용자의 얼굴형을 정확하게 판단하는 것을 중점으로 하고 있다.

본 서비스는 사용자의 개인 정보와 초상권을 침해하지 않는 것을 목표로 한다. 사용자의 정보나 사진을 따로 저장하지 않는 방식으로 개발을 진행함으로써 사용자의 개인 정보와 초상권을 보호할 것이다.

서비스의 접근성에 대한 목표는 '누구나 사용할 수 있는 서비스'이다. 즉 사용자가 접근하기 어렵거나 불편한 서비스는 최대한 지양한다. 그러므로 사용자가 프로그램을 설치하거나, 추가적인 설정을 하지 않아도 이용할 수 있는 서비스를 제작한다. 깔끔하고 간단한 UI를 통해 어떤 사용자든 편하게 접근할 수 있게 하며 직관적으로 서비스를 이해할 수 있도록 한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

1. 웹

1.1 Front-End

1. Create-react-app을 통한 프로젝트 생성

페이스북에서 만든 react 웹 개발용 boilerplate인 Create-react-app을 통해 프로젝트를 생성하였다. 먼저 npx create-react-app 명령어를 사용해서 프로젝트 환경을 새로 설정하였다.

2. MobX – React 설치

React 상에 존재하는 여러 상태들을 효율적으로 관리해 주는 상태 관리 라이브러리인 mobX를 설치하였다. 상태가 많아지면 하나하나 관리하는 것이 힘들 것이라고 판단해 mobX를 채택하였다.

3. React- Reveal 설치

여러가지 애니메이션을 지원해주는 라이브러리인 React-Reveal을 설치하였다.


4. Styled-Components React 설치

Css 스타일링을 위한 프레임워크인 styled-components를 설치하였다.

5. 조건부 렌더링을 통한 페이지 분기

6. Store 폴더 생성

mobx에서 지원하는 observable 변수와 action 함수들을 선언하고 정의하는 공간인 store 폴더를 디렉토리에 생성하였다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

7. Provider적용

store폴더에 있는 js파일들에 선언된 각종 observable 변수와 action 함수들을 제공해주기 위해 index.js에 mobX의 provider를 적용하였다.

8. 조건부 렌더링을 통한 페이지 분기

먼저 Home.js에 mobX에서 지원하는 iinject, observable을 활용해 provider가 적용된 stores 폴더의 ManageFile.js를 참조하였다. 그 후 Home.js에서 ManageFile.js에서 선언된 observable 변수인 pageIndex를 활용해 조건부 렌더링으로 페이지를 분기하였다. PageIndex의 값에 따라 렌더링 되는 컨테이너를 분리함으로써 버튼이 누를 때 페이지가 넘어가는 효과를 구현할 수 있었다.

9. Container, Content 컴포넌트 적용

생성한 Container.js, Content.js 컴포넌트를 활용해 모든 페이지의 상위 요소에 적용시켜서 코드를 재사용 할 수 있도록 하였다. 코드의 재사용성을 높임으로써 작업과 resource의 효율을 증가시킬 수 있었다.


10. React-webcam을 이용한 사용자의 안면 인식 input 제공

React-webcam 라이브러리를 이용해 사용자가 사용하는 기기의 webcam을 통해 얼굴이 인식되게 하였다.

11. getScreenShot 함수를 통한 웹캠 캡처

getScreenShot함수를 통해 웹캠 화면을 캡처하고 이미지의 형태로 모델에 넘겨주었다.

12. 조건부 렌더링을 통한 얼굴 인식 상태 감지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

조건부 렌더링을 통해 얼굴이 인식되었는지 확인한 후 얼굴 인식 상태에 따라 화면에 렌더링되는 문구를 다르게 출력하였다.

13. 성별 선택 기능 구현

페이지의 첫 시작 페이지에 성별을 선택할 수 있는 버튼을 2개 만들어서 성별을 선택하고 mobX변수로 성별을 저장하였다.

14. 얼굴형 출력 페이지 구현

얼굴형이 인식되면 해당 얼굴형을 출력해주고 얼굴형에 따른 페이지를 출력해주는 페이지를 구현하였다.


1.2 Back-End

[서버]

본 서비스는 산학협력 기업 “로보링크”의 요청으로 서버리스 아키텍처의 형태를 띈다. 때문에 외부에 다른 서버를 두고 있지 않고, gh-pages라는 외부 라이브러리를 통해 React Application을 Github page를 통해 배포한다.

Github page는 정적 페이지를 배포해 주는 github의 service로 React의 SPA(Single Page Application)을 배포할 수 있다. SPA는 단일 페이지 어플리케이션으로, 겉으로 보기에는 여러 페이지들로 이루어진 웹 앱이지만 실제 동작하는 index.html 하나이기 때문에 Github page를 통해 배포 가능하다.

https://{users_github_name.github.io}/{Repository_name}의 형태로 배포된 React SPA에 도메인이 부여되며 사용자 설정을 통해 custom 도메인을 사용할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

또한 본 서비스에서 얼굴형 분석에 사용되는 모델 또한 Github page를 통해 배포되어 있으며, 배포한 모델은 어플리케이션 내부의 Tensorflow.js 프레임워크를 통해 load 하여 사용할 수 있다.

[데이터베이스]


본 서비스는 유저의 정보나 사진, 관리자의 정보 등 외부에서 따로 관리해야 하는 데이터가 없기 때문에 구성되어 있지 않다.

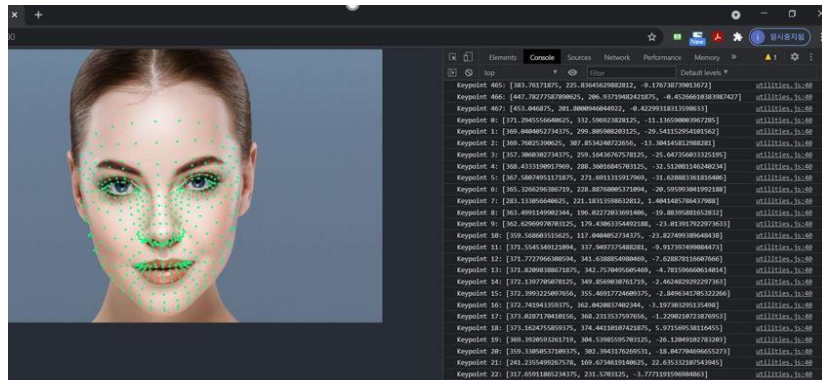
2. 얼굴 인식

2.1 Tensorflow.js의 오픈소스 모델 이용

Tensorflow.js에서 다양한 오픈소스 모델을 제공한다. 본 프로젝트에서는 개발 목표에 맞게 사용자의 얼굴형을 판단하기 위해 제공되는 오픈소스 모델 중 Face-Landmarks-Detection 모델(이하 FLD 모델)에 대해 조사해 보았다.

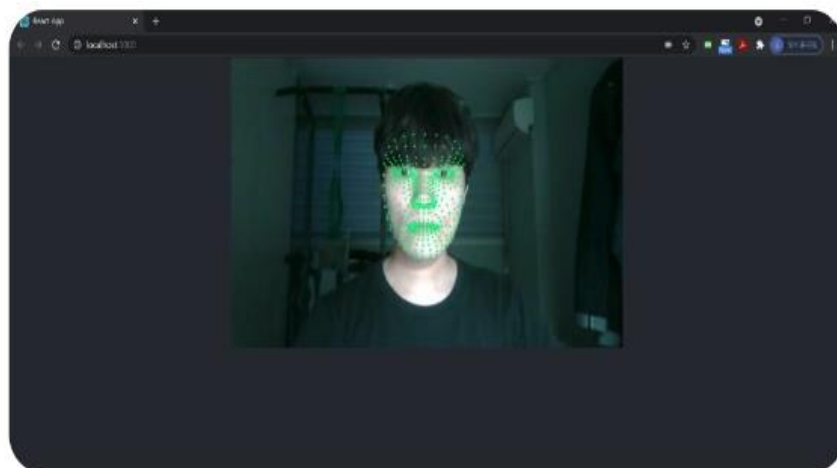
FLD 모델은 Google에서 개발한 MediaPipe 프레임워크를 이용하여 개발된 모델이다. MediaPipe란 모바일이나 클라우드에서 머신러닝을 실행할 수 있는 프레임워크로 작은 크기의 모델(lightweight model)로 구성되어 가볍고 좋은 성능의 모델을 제공한다. FLD 모델은 웹캠이나 사진을 이용하여 실시간으로 사용자의 얼굴을 인식하고 사용자의 얼굴에 대한 특징점들을 468개 추출하여 제시한다. (그림 2와 그림 3 참고)

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26




Keypoint 0: [371.2945556640625, 332.596923828125, -11.136590003967285]

[그림2] 특징점 추출 1



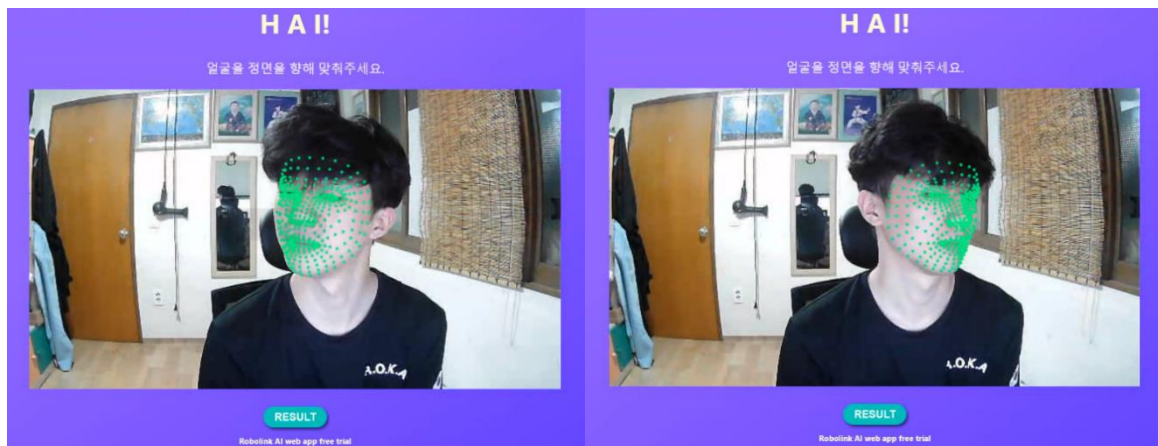
[그림3] 특징점 추출 2

위와 같이 FLD 모델의 성능을 파악하고 출력되는 468개의 특징점들을 이용하여 얼굴형을 판단하는 모델 개발을 계획하게 되었다.

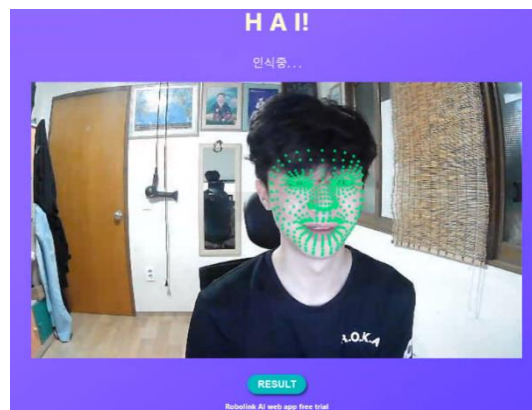
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

2.2 정면 인식 기능


Face-Landmarks-Detection을 사용해 얼굴의 윤곽점들을 추출하고, 그 점들의 좌표값을 사용해 정면 인식 기능을 구현했다. 사람마다 얼굴의 크기가 다르고, 얼굴을 인식시킬 때 웹캠과 얼굴의 거리가 모두 다르기 때문에 수치화를 진행하고 가장 올바르게 인식시킨 케이스를 기준으로 비율화를 진행해 좌표값이 사람마다 큰 차이가 있더라도 정면을 보고 있는 것을 인식할 수 있도록 했다. 정면을 바라보고 있으면 "인식 중..."이라는 텍스트가 출력되고 정면을 바라보지 않고 있다면 "얼굴을 정면을 향해 맞춰주세요."라는 텍스트가 출력된다. 5초간 정면을 올바르게 바라보면 얼굴형 분석을 시작한다.



[그림4,5] 정면을 바라보지 않은 상태



[그림6] 정면을 바라본 상태

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

3. 얼굴형 판단 AI 모델 개발

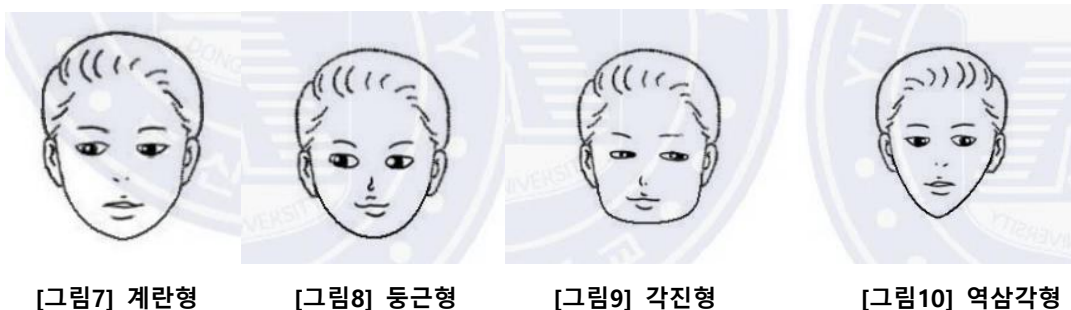
3.1 Data 수집

모델 학습을 위한 Dataset을 구축하기 위해 얼굴 사진 데이터들이 필요했다. 머신러닝 모델 학습을 위한 Dataset이 다량 존재하는 Kaggle과 GitHub를 중심으로 데이터를 수집하였고 Kaggle에서 수집한 데이터의 경우 Public Domain으로 저작권이 없음을 확인하였고 GitHub에서 수집한 데이터는 "출처 명시"와 "비영리적인 목적"으로 사용한다는 저작권을 확인하였다.

(주)로보링크(산학 협력 기업)와도 위와 관련된 사항에 대해 회의를 진행하였고 회의 결과 승인을 받고 개발을 진행하였다.

3.2 Data Labeling


얼굴형을 분류하는 논문을 토대로 계란형, 둥근형, 각진형, 역삼각형 4가지로 얼굴형을 구분했다. 구분된 기준으로 수집한 데이터에 대해 직접 Data Labeling을 진행했다.



3.3 초기 개발 모델

3.3.1 Image Preprocessing

Data Labeling을 통해 얼굴형 별로 구분한 사진들로 Dataset을 구축하기에 앞서 개발할 모델의 성능 향상을 위해 정확하고 신뢰할 수 있는 데이터들을 추려냈고 그 결과 얼굴형 별로 각각 100장씩 총 400장의 사진으로 추려졌다. 하지만, 400장의 사진으로 Dataset을 구축하여 학습을 진행하기엔 많이 부족하다고 판단하였다. 이와 관련하여 자료 조사를 했고 그 결과 Data Augmentation 기술에 대한 내용을 접할 수 있었다. Data

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


Augmentation 이란 원본 데이터에 증강 기법을 적용하여 다양한 효과가 적용된 데이터들을 확보할 수 있게 해주는 기술이다. Data Augmentation에서 제공하는 효과로는 Horizontal Flip(좌우 반전)과 Vertical Flip(상하 반전), Zoom(확대), Bright(밝기 조절), Rotation(회전), Rescale(정규화) 등이 있다. 본 프로젝트에서는 일반적으로 많이 사용되는 Rescale과 Horizontal Flip, Rotation, Bright 효과를 적용하여 약 3000장의 Data를 확보할 수 있었다.

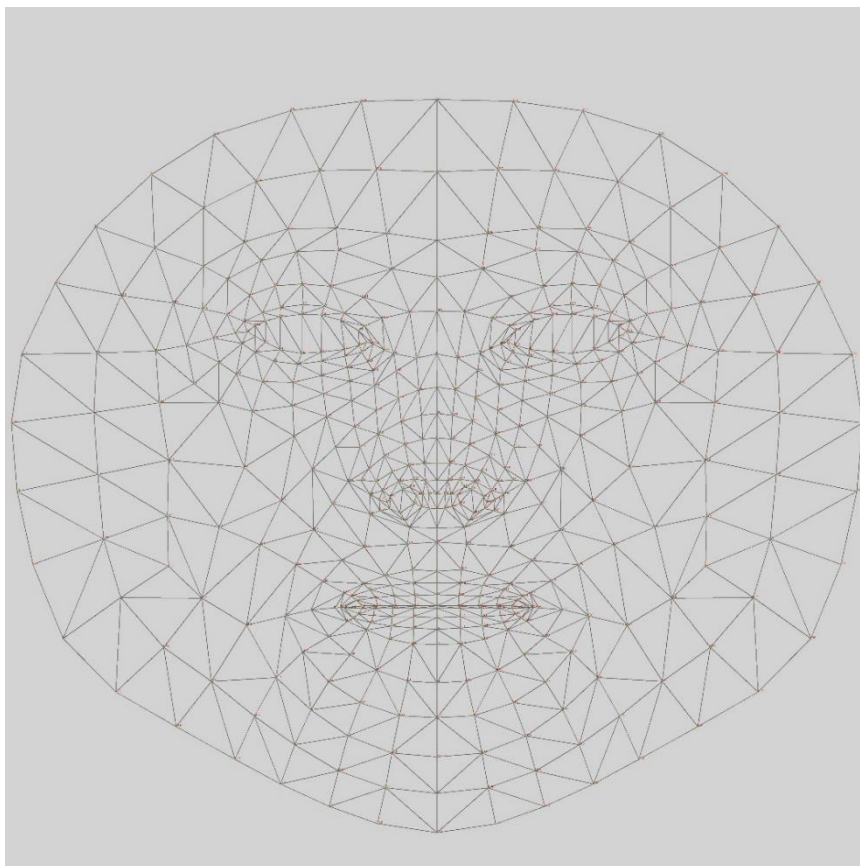
3.3.2 Dataset 구축

본 프로젝트의 개발 목표를 세울 때 얼굴형 판단 기준을 설립하기 위한 방법을 고안하던 중 FLD 모델의 특징점들을 Dataset으로 하여 딥러닝 모델을 개발하는 방법에 대해 논의하였고 이와 같은 방법으로 모델을 개발한 사례가 있는지 조사해 보았다. “재실자 활동량 산출을 위한 Pose 분류 모델 개발”이라는 제목의 논문과 웹 검색 자료에서 해당 사례를 찾을 수 있었다. 최종적으로 위와 같은 사례들과 함께 간단한 신경망 구조로 가벼운 모델을 개발할 수 있을 것이라는 결론을 내리고 이 방법을 선택하게 되었다.

본 프로젝트는 입력 데이터에 대해 4가지의 얼굴형 중에서 알맞은 결과를 출력하는 것이 개발 목표이기 때문에 분류 모델을 개발하기로 결정하였고 이에 맞게 Dataset을 구축하는 작업을 진행하였다.


Dataset을 구축하기 위해 3.3.1번 항목에서 설명한 바와 같이 정확하고 신뢰성 있는 사진들을 다량 확보하였고 이 사진 데이터들을 FLD 모델에 입력으로 넣어 Prediction 결과로 추출되는 특징점들 중에서 얼굴형 판단의 기준이 될 수 있는 점들만 추출하였다. 추출한 특징점들은 468개의 점들 중에서 얼굴 외곽의 점들에 해당한다. 특징점들을 추출할 때 그림 11을 참고하여 작업을 진행하였다.

	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26



[그림 11]

얼굴 외곽에 해당하는 점들을 추출한 결과 총 72개의 점들이 나왔고 (x, y, z) 좌표에서 x좌표와 y좌표를 이용하였다. 해당 좌표값들은 saveAs 모듈을 통해 txt 파일로 저장하도록 하였다. txt 파일에는 144개의 실수형 숫자들이 저장된다. 총 이미지가 3000개이고 이미지 별로 144개의 데이터들을 가지기 때문에 이를 효율적으로 처리하기 위해 Python의 Pandas 프레임워크를 사용하였다. Pandas 프레임워크는 다량의 데이터를 간편하게 처리할 수 있는 Dataframe 모듈을 제공한다. 앞서 txt 파일에 저장된 데이터들을 Dataframe으로 변환하여 나타내었다. Dataframe으로 변환할 때 이전에 얼굴형 별로 라벨링 작업을 한 사진들을 이용하였기 때문에 데이터에 해당하는 라벨을 붙이는 작업은 수월하게 할 수 있었다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26

5	6	7	8	9	...	135	136	137	138	139	140	141	142	143	144
313615	-0.987851	-0.944359	-0.839149	-0.224755	...	-0.206127	-1.037350	-0.759505	-1.003949	-0.659576	-1.084403	-0.976509	-1.095312	-0.982432	Egg
739230	-1.009730	-0.845483	-1.095952	-0.548914	...	-0.852859	-1.189803	-0.475674	-1.184777	-0.512888	-1.171534	-0.447948	-1.219151	-0.266942	Egg
786031	-0.960051	-0.891157	-1.012012	-0.513473	...	-0.074382	-0.868747	-0.400037	-0.871746	-0.335952	-0.894109	-0.510410	-0.866587	-0.476268	Egg
307531	-0.639081	-1.536051	-0.892747	-1.556849	...	-0.546049	-1.158975	-0.257851	-1.202208	-0.271115	-1.058248	-0.260661	-1.008286	-0.406285	Egg
346272	1.011885	-0.768678	1.068164	-0.547652	...	-0.519206	0.998630	-0.751113	0.994185	-0.718287	0.972725	-0.774800	0.941750	-0.708678	Egg
...
233080	-1.069888	1.045799	-0.984188	1.599575	...	1.613458	-0.740416	1.045020	-0.707992	1.140136	-0.814137	0.860620	-0.772646	0.923779	Rtriangle
316058	0.938100	-0.783227	1.134481	-0.904415	...	-0.679050	1.142584	-1.048932	1.142231	-1.009285	1.146077	-1.114334	1.181624	-1.217877	Rtriangle
219366	0.822222	-0.310893	1.141171	-0.213859	...	-0.712405	1.195552	-1.331935	1.232346	-1.259992	1.102580	-1.429094	1.107638	-1.422068	Rtriangle
285436	1.273969	0.375174	0.862095	0.404033	...	1.303287	0.886368	1.740813	0.829087	1.713777	0.964748	1.770480	0.987104	1.791763	Rtriangle
133460	-0.716905	0.218385	-0.985860	0.116950	...	0.987575	-1.078797	1.493314	-1.127386	1.450859	-0.961419	1.525896	-0.907449	1.367130	Rtriangle

[그림 12]

각 행의 값들은 전처리 이전에 100이 넘어가는 숫자들로 구성되어 있었고 이를 통해 임시로 학습을 진행했을 때 학습이 전혀 진행되지 않는 문제가 발생해 그림12와 같이 정규화를 적용하여 Dataframe으로 표현하였다.

작업이 끝난 Dataframe을 javascript에서 이용하기 위해 csv 파일로 저장하였고 javascript에서 Danfo.js를 사용하였다. Danfo.js는 Pandas와 같은 기능을 하는 프레임워크로 javascript에서도 데이터를 처리할 수 있는 기능들을 제공한다. 그림 13은 Danfo.js를 이용하여 로컬에 저장된 csv 파일을 브라우저 상에서 Dataframe으로 나타낸 것이다.

Shape: (3000,145)															
	0	1	2	3	...	141	142	143	144						
0	-1.1853601491...	1.22010264010...	-0.9878323539...	1.74866674351...	...	0.49695846663...	-1.0869551251...	0.60889770519...	Circle						
1	1.18835979273...	1.18763956744...	1.73080684624615	-0.3236656534...	...	-1.0751989944...	0.68741430729...	-1.1361617202...	Egg						
2	-0.8509349410...	-0.7952777315...	-0.9203839511...	-1.1511154070...	...	-0.651196579117	-1.1055913128...	-0.7137254874...	Egg						
3	-0.7738189442...	-0.5727256704...	-0.9376026877...	-1.1179213860...	...	-0.1255407919...	-1.1783295283...	-0.0453956162...	Egg						
4	0.78829821606...	0.66024403011...	0.78337973070...	1.46130885760...	...	0.49589109404...	0.82937700033...	0.61125433782...	Rect						
5	1.05572780139...	-0.6181252481...	0.88518472879...	-0.6893433419...	...	-0.1249449123...	1.14647363504...	0.06351260468...	Rect						
6	-1.0423938579...	-0.8315737900...	-1.0383060692...	-0.5203215115...	...	-0.9495036854...	-1.1110092636...	-0.8744443916...	Rect						
7	0.79763658551...	2.24520809939...	0.87067770838...	2.33718395918...	...	1.05079709238...	0.83344038061...	0.88801691391...	Circle						
8	-0.7403324710...	0.58050967683...	-0.8136574893...	0.24531650873...	...	1.65672097918...	-0.9400486306...	1.87142102900...	Rect						
9	1.26035957832...	0.60447780627...	1.01393635108...	-0.2077239696...	...	1.50056857150...	0.93124030187...	1.38537538183...	Circle						
10	-0.7711567936...	-1.2057207506...	-0.7709687206...	-1.5329543585...	...	0.16340117040...	-0.5545930396...	0.21876192509...	Circle						
11	1.15914783374...	-1.3062227509...	1.11158108156...	-1.4578590447...	...	-0.4574113033...	0.85860936600...	-0.5147711201...	Egg						
12	-1.0328959194...	-1.1435528669...	-1.1215211167...	-0.9280578994...	...	-1.1793762401...	-0.9703875655...	-1.3375636750...	Rect						
13	-0.8813217411...	0.34557423226...	-0.7608383026...	-0.1955622844...	...	0.33794348332...	-0.9948393669...	0.19860271922...	Circle						
14	1.07285120227...	0.79983922391...	1.00572362439...	0.25462900206...	...	1.13159971881...	1.03187783469...	0.95119204009...	Rect						
15	-1.0252751257...	-0.9509935529...	-1.0561765430...	-0.5787893146...	...	-0.8338013371...	-1.1707034058...	-0.7546568203...	Circle						
16	0.99905130533...	-1.0819663059...	0.87990738372...	-1.0600614539...	...	-1.0272706033...	1.00306134301...	-1.1993917471...	Rect						
17	-0.9315639562...	1.05612175833...	-0.8851018806...	0.87765802233...	...	1.01600334921...	-1.1186267771...	1.02242691543...	Rtriangle						
18	-0.7331695418...	-0.4391242863...	-0.55390100810...	-1.3187325222...	...	-0.8613639230...	-1.3189234131...	-0.8260875354...	Circle						
19	0.68973161474...	0.67069883442...	0.83593619654...	1.37167668277...	...	0.12668355330...	1.19048075818...	0.00601515927...	Circle						
20	1.20787589442...	1.27478205343...	0.96425181198...	0.54684909261...	...	1.31783438407...	0.88915690006...	1.27314362054...	Circle						

[그림13]


[그림13]과 같이 Dataframe을 이용하여 모델의 독립변수와 종속변수를 구성하였다. 독립변수는 모델의 특징들을 의미하는 것으로 본 프로젝트에서 독립변수는 144개로 구성된다. 종속변수는 각 데이터마다 해당하는 라벨을 의미한다. 본 프로젝트에서 라벨은 범주형 데이터로 모델을 학습하기 위해서 수치로 변경하는 것이 필요하다. 이를 위해 사용되는 방법으로 One Hot Encoding 방법이 있다. One Hot Encoding 이란 범주형 데이터를 수치형 데이터로 변환하는 것으로 주어진 범주형 데이터에 대해 1 또는 0의 인덱스를 부여하는 방법이다. 정답인 범주에 대해서는 1을 부여하고 정답이 아닌 범주에 대해서는 0을 부여한다.

Shape: (3000,4)

	Circle	Egg	Rect	Rtriangle
0	1	0	0	0
1	0	1	0	0
2	0	1	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	1	0
6	0	0	1	0
7	1	0	0	0
8	0	0	1	0
9	1	0	0	0
10	1	0	0	0
11	0	1	0	0
12	0	0	1	0
13	1	0	0	0
14	0	0	1	0
15	1	0	0	0
16	0	0	1	0
17	0	0	0	1
18	1	0	0	0
19	1	0	0	0
20	1	0	0	0

[그림14]

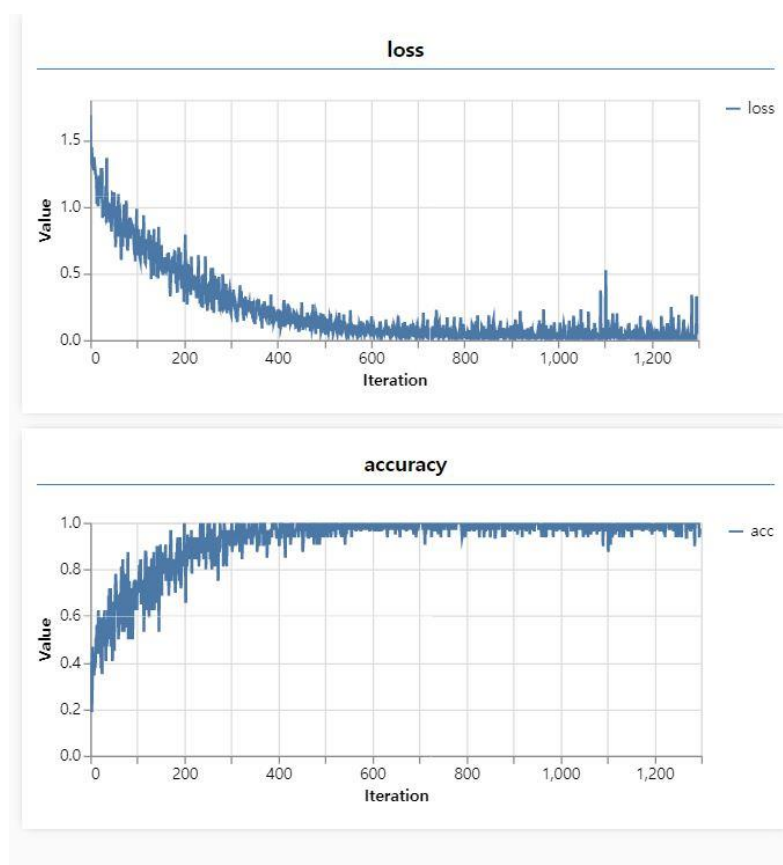
[그림14]와 같이 [그림13]에서 나타낸 Dataframe에서 범주형 데이터를 Danfo.js의 OneHotEncoder 모듈을 이용하여 수치형 데이터로 변환한 것이다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

3.3.3 학습 진행 및 결과


모델은 "입력층-은닉층-은닉층-출력층"으로 완전 연결층의 구조로 설계하였다. 은닉층은 각각 200개와 100개의 노드로 구성하였고 활성화함수는 relu를 적용하였다. 4개의 class에 대한 분류 모델이므로 출력층은 4개의 노드로 구성하고 활성화함수로 softmax를 적용하였다. 최적화 함수로는 Adam과 손실 함수로는 categorical crossentropy를 적용하였다.

모델의 학습이 정상적으로 이루어지는지 확인하기 위해 먼저 3000개 데이터를 모두 Train Set으로 하여 학습을 진행해 보았다.



[그림15]

Train Set에 대한 학습 결과는 그림 15와 같이 높은 정확도를 확인할 수 있었다. 실제로 모델의 동작을 확인하기 위해 학습에 사용했던 사진을 FLD 모델에 넣고 출력값을 곧바로 개발한 모델에 입력으로 넣어 보았다. 하지만, 학습한 데이터임에도 불구하고 올바

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투뎀	
	Confidential Restricted	Version 1.5	2021-05-26

른 값을 출력하지 못하는 것을 확인할 수 있었다. 원인을 찾아본 결과 같은 사진이라도 FLD 모델에 입력으로 들어간 사진의 특징점들은 미세하지만 값들의 차이가 있었고 이것은 개발한 모델의 성능에도 영향을 주었다. 또한, 모델 개발 계획에 대해 멘토님과 논의를 했을 때 얼굴형을 판단하는데 지금과 같은 방식이 일반적인 방식은 아니라는 피드백을 받았다. 결국, 본 프로젝트의 완성도와 기한을 고려해 모델 개발 방식을 변경하기로 결정하였다.


3.4 변경한 개발 모델

3.4.1 모델 개발 계획

3.3번에서 진행했던 방법에서 이미지를 Dataset으로 하여 모델 개발하는 방법에 대해 조사하였다. 이미지를 Dataset으로 하는 모델의 구조는 CNN(Convolutional Neural Network)으로 일반적으로 이미지나 영상 데이터를 학습할 때 사용되는 신경망이다. 또한, CNN으로 개발된 좋은 성능의 모델들이 많이 존재하였고 이와 관련된 자료들도 많았기 때문에 변경한 방법으로 개발을하기로 결정하였다. 모델 개발은 Transfer Learning을 하여 진행하였다. Transfer Learning이란 이미 만들어진 모델의 신경망과 가중치를 이용하여 자신의 개발 목적에 맞는 모델을 개발하는 것으로 검증된 모델을 이용한다면 더 빠르게 좋은 성능의 모델을 개발할 수 있다.

본 프로젝트는 여러 검증된 CNN 모델 중에서 ImageNet Dataset으로 학습된 VGGNet(VGG16)과 MobileNet V2, Inception V3를 이용하여 Transfer Learning을 통해 모델을 개발하는 것으로 계획하였다.

VGG16은 멘토님의 조언과 더불어 대부분의 사용자가 CNN 모델을 개발하는데 이용하기 쉽고 높은 성능을 보인다는 점에서 선택하였다. MobileNet V2는 FLD 모델이 개발될 때 사용된 모델로 이와 관련하여 자료 조사를 진행한 결과 모바일이나 사물 인터넷과 같은 환경에서도 딥러닝을 활용하고자 개발된 모델이다. 본 프로젝트에서는 낮은 성능의 환경에서도 높은 성능의 모델을 개발할 것을 기대하여 선택하게 되었다. Inception V3는

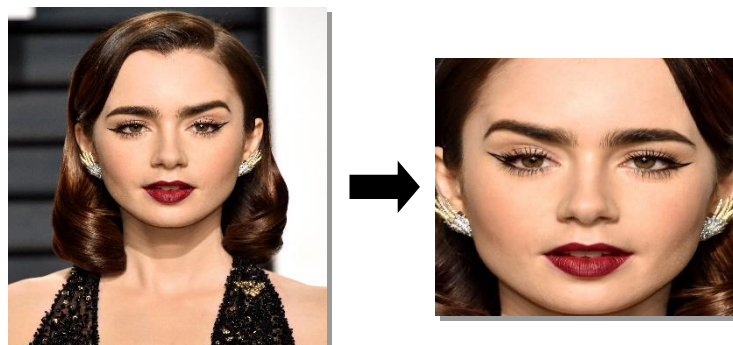
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

“Face shape classification using Inception V3”라는 제목의 논문에서 정보를 얻었다. 해당 논문에서는 Inception V3와 CNN을 이용하지 않은 분류 모델의 성능 비교를 제시하면서 Inception V3의 높은 성능에 대해 이야기한다. 이와 같은 근거를 통해 본 프로젝트에서 사용하기로 결정하였다.

3.4.2 Dataset 구축

Dataset 구축은 3.3.1번 항목을 진행하여 확보한 데이터들을 이용하였다. 3000개의 데이터들을 Train Set으로 하고 500개의 데이터를 추가로 확보하여 Test Set을 구성하였다. 또한, CNN을 이용하여 모델을 개발하기 때문에 이미지 데이터에 대한 전처리가 추가적으로 필요했다.

Python과 OpenCV를 사용해 학습시킬 데이터에 대해 이미지에서 사람의 얼굴 부분만 추출하여 전처리를 진행했다. Haarcascade_frontalface_alt.xml URL을 변수에 저장하여 CascadeClassifier() 메소드에 인자로 넣어 사전 학습된 얼굴 분류기를 생성한다. Vidimread() 메소드로 이미지를 받고 grayscale로 변환한다. detectMultiScale() 메소드에 grayscale로 변환한 이미지를 인자로 넣고 Scalefactor값과 minNeighbors도 인자로 넣는다. 이후 리턴된 좌표를 기준으로 ROI를 설정하여 얼굴 부분만 추출한 이미지로 저장하여 전처리를 진행했다.



[그림16]



	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

그림16과 같이 전처리를 진행한 사진들로 Dataset을 구성하여 CNN모델 학습 시 얼굴을 제외한 특징점들이 추출되는 것을 최소화하여 정확하고 신뢰할 수 있는 Dataset을 구축하였다. Keras의 ImageDatagenerator 모듈을 통해 Train Set과 Test Set을 이용할 수 있었고 이전에 얼굴형 별로 라벨링 작업을 한 사진들을 이용하였기 때문에 ImageDatagenerator의 flow_from_directory API를 통해 자동으로 Dataset에 대한 라벨링을 처리할 수 있었다.

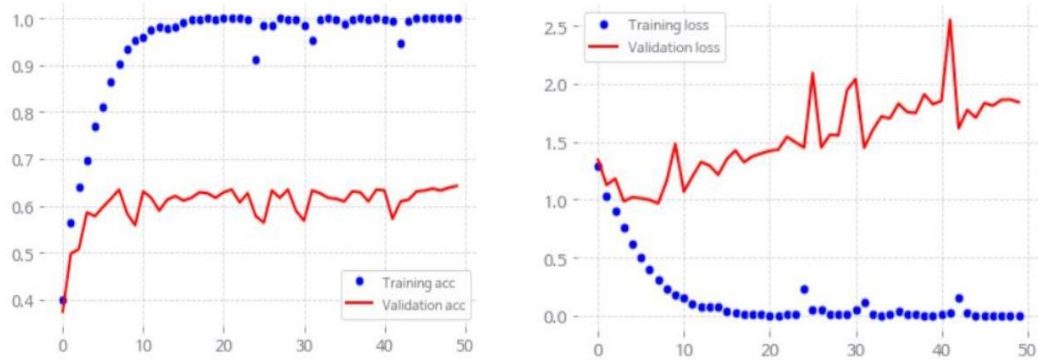
3.4.3 학습 진행 및 결과

모델 학습은 앞서 선택한 3개의 모델에 다음과 같은 조건을 적용하여 진행하였다. 또한, 학습 진행은 Keras의 Early Stopping을 통해 모델 성능의 개선이 없을 때 학습을 중지하도록 설정하였다. 먼저, VGG16을 이용해 학습을 진행해 보았다. Keras를 통해 모델을 불러오고 동시에 include_top=False로 설정하여 VGG16의 Input Layer에서 5번째 블록까지 base model로 만들고 flatten을 적용하여 이전 Layer의 모양을 일차원으로 변경한다. 이후, 2048개의 노드로 구성된 완전 연결층(FC Layer1)과 1024개의 노드로 구성된 완전 연결층(FC Layer2), 512개의 노드로 구성된 완전 연결층(FC Layer3)으로 구성하고 각각 활성화함수로 "relu"를 적용하였다. 출력층은 4개의 노드로 구성하고 활성화함수로 "softmax"를 적용하여 전체 모델을 설계하였다. 최적화 함수로 "Adam"을 사용하고 손실 함수로 "categorical_crossentropy"를 사용하였다.

첫 번째로 base model의 가중치를 고정하지 않고 모델의 전체 Layer에 대해 학습을 진행했다. 하지만, 전혀 학습이 진행되지 않거나 학습이 진행되어도 약 50%의 Accuracy 밖에 보이지 못했다. 두 번째로 base model의 가중치를 고정하고 dropout의 적용 여부에 나눠 학습을 진행해 보았다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

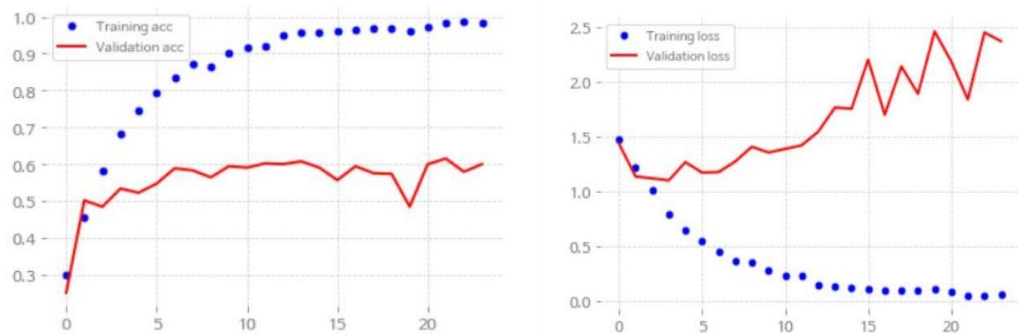
- learning rate = 0.0001



[그림17]


[그림 17]과 같이 Test Set에 대한 Accuracy는 약 60%로 유지되나 Loss는 계속해서 증가하는 것으로 보아 오버 피팅이 된 것을 알 수 있다. 오버 피팅을 방지하기 위해 다음 실험에서는 dropout을 적용해 보았다.

- dropout=0.5 적용(FC layer2와 FC Layer3 사이), learning rate= 0.0001

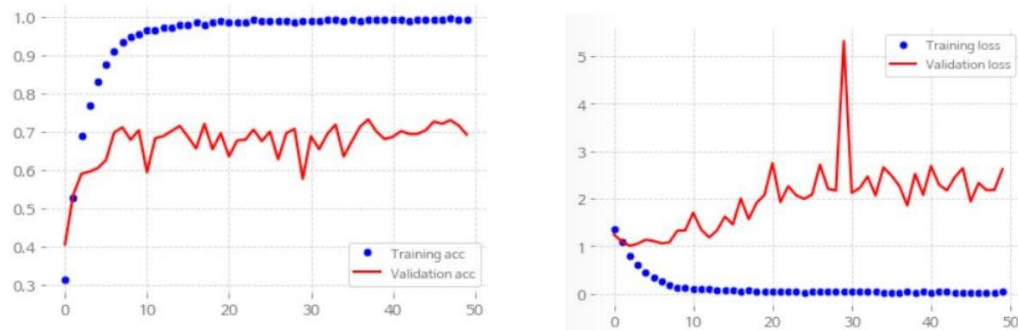


[그림 18]

[그림18]과 같이 Test Set에 대한 Accuracy는 약 60%로 유지되나 Loss는 계속해서 증가하는 것으로 보아 이전과 동일하게 오버 피팅이 된 것을 알 수 있다. 세 번째로 base model의 가중치를 고정하지 않고 최적화 함수를 RMSprop으로 변경하여 dropout의 여부로 나눠 학습을 진행해 보았다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26

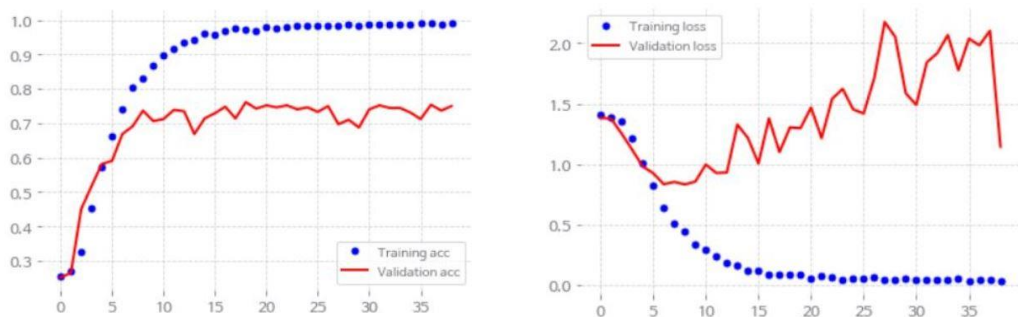
- learning rate = 0.00002



[그림19]


[그림19]와 같이 Accuracy는 약 70%로 Adam을 적용했을 때보다 10%정도 상승한 것을 확인할 수 있다. 하지만, Loss의 경우 Adam을 적용했을 때와 동일하게 상승하는 것으로 보아 여전히 오버 피팅이 되는 것을 확인할 수 있다. 오버 피팅을 방지하기 위해 다음 실험에서 dropout을 적용해 보았다.

- dropout 0.5 적용(FC Layer2와 FC Layer3 사이), learning rate = 0.00002



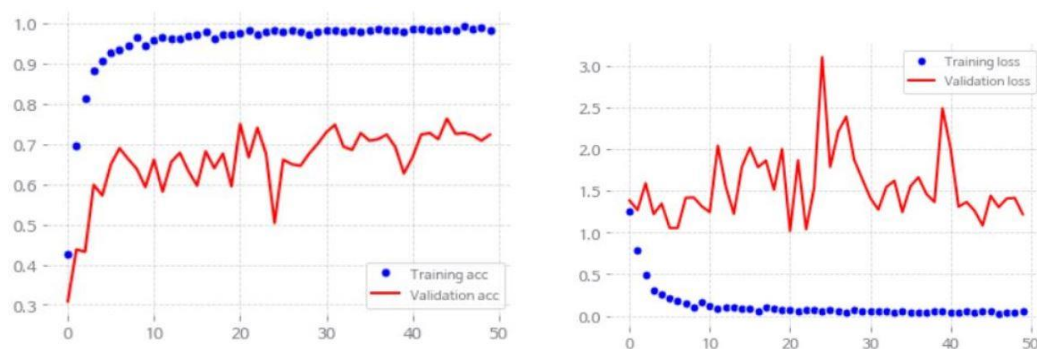
[그림20]

[그림20]과 같이 Accuracy는 약 73%로 이전 보다 약 3% 상승했으나 Loss의 경우 여전히 오버 피팅이 되는 것을 확인할 수 있다. 추가로, Learning rate를 변경하여 실험을 진행하려 했으나 각 모델의 성능을 비교하기 위해 다음 모델에 대해 학습을 진행하였다. 다음으로, MobileNet V2를 이용해 학습을 진행해 보았다. VGG16에서와 마찬가지로 MobileNet V2를 불러올 때 include_top=False로 설정하여 base model을 구성하였다. base model에 GlobalAveragePooling2D를 적용하여 이전 Layer에서 각 채널의 kernel size에 대한 평균값을 계산하여 전체 채널의 개수만큼 값이 출력되도록 한다. 이후, 1280

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26

개의 노드로 구성된 완전 연결층 2개(FC Layer1, FC Layer2)와 640개의 노드로 구성된 완전 연결층(FC Layer3)를 구성하였고 활성화함수로 "relu"를 적용하였다. 출력층은 4개의 노드로 구성하고 활성화함수로 "softmax"를 적용하였다. 최적화 함수로 "Adam"을 사용하고 손실 함수로 "categorical crossentropy"를 사용하였다. base model의 가중치를 고정하지 않고 다음과 같이 학습을 진행해 보았다.


- learning rate = 0.0001
- dropout 0.5 적용(FC Layer2와 FC Layer3 사이), learning rate = 0.0001



[그림21]

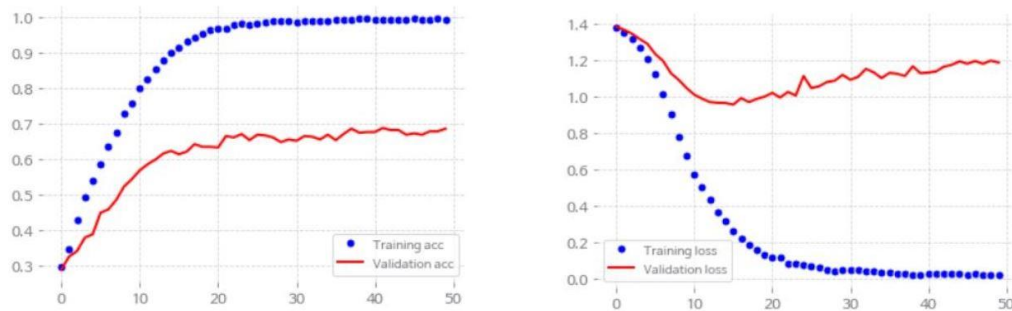
[그림21]과 같이 전체적으로 Accuracy와 Loss가 불안정하게 나타나는 것을 볼 수 있다.

마지막으로, Inception V3에 대해 학습을 진행하였다. base model 구성은 이전과 동일하게 하였고 Mobilenet과 마찬가지로 GlobalAveragePooling2D를 적용하였다. 이후, 1024개의 노드로 구성된 완전 연결층(FC Layer1)과 512개의 노드로 구성된 완전 연결층 2개(FC Layer2, FC Layer3)를 구성하고 활성화함수로 각각 "relu"를 적용하였다. 출력층은 4개의 노드로 구성하고 활성화함수로 "softmax"를 적용하였다. 최적화 함수로 "SGD"를 사용하고 손실함수는 "categorical crossentropy"를 사용하였다. base model의 가중치를 고정하고 임시로 실험한 결과 성능이 좋지 못해 가중치를 고정하지 않고 대부분의 실험을 진행하였다. 또한, 최적화 함수로 SGD를 사용하기 때문에 momentum을 적용하여 SGD를 이용할 때의 변화량을

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투뎀	
		Confidential Restricted	Version 1.5	2021-05-26

줄였다.

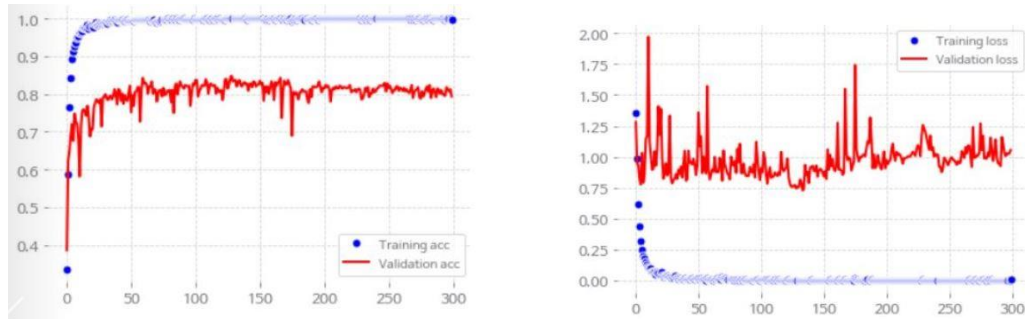
- learning rate=0.001, momentum=0.9



[그림22]

[그림22]와 같이 Accuracy는 약 68%로 유지되나 Loss는 상승하는 것으로 보아 이전 실험에서와 같이 오버 피팅이 발생한 것을 볼 수 있다.


- dropout=0.5(FC Layer2와 FC Layer3사이), learning rate=0.001, momentum=0.9



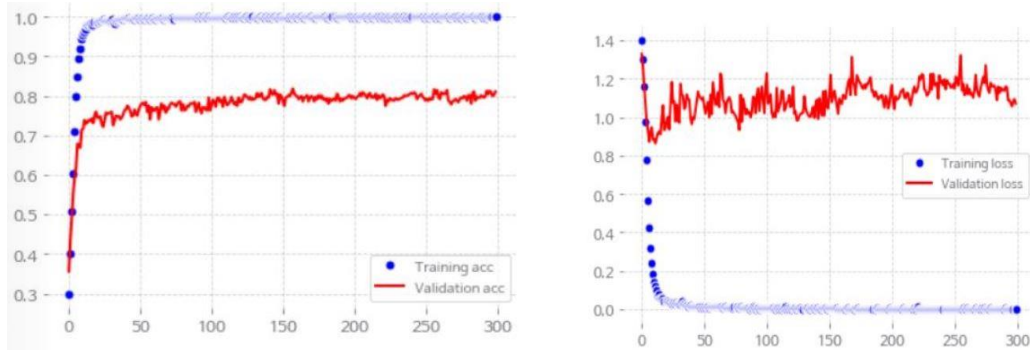
[그림23]

[그림23]을 보면 Accuracy는 약 80%로 유지되고 Loss의 경우 전체적으로 균일하게 유지되는 것으로 보아 이전 상황에 비해 많이 개선된 것을 볼 수 있다.

[그림23]의 결과는 이전에 진행했던 실험들 중에서 가장 높은 성능을 보였기 때문에 같은 조건에서 momentum과 FC Layer의 개수를 변화시켜 학습을 진행해 보았다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

- dropout=0.5(FC Layer2와 FC Layer3사이), learning rate=0.001, momentum=0.6



[그림24]

[그림24]의 학습 결과를 보면 Accuracy는 [그림23]에서와 같지만 Loss의 경우 약간의 차이가 있는 것을 확인할 수 있다. [그림24]에서는 이전 실험 결과와 비교해 보면 비교적 안정적이지만 학습이 진행되면서 Loss가 조금씩 상승하는 것을 볼 수 있다. 따라서, 오버 피팅이 발생한 것을 알 수 있다.


위와 같이 모델의 성능을 올리기 위해 여러 번의 실험을 진행했지만 완벽한 성능의 모델을 개발하는 것에는 어려움이 있었다. 최종적으로 본 프로젝트에서는 [그림23]과 같이 여러 실험들 중에서 가장 안정적인 결과를 보인 Inception V3 모델을 사용하기로 결정하였다.

3.5 모델 배포

모델 배포 방식에는 일반적으로 AWS나 Google Cloud를 이용하는 클라우드 환경에서의 배포 방법과 Django나 Flask와 같은 프레임워크를 이용하여 Rest API를 구현하고 이를 통해 웹 서버를 구축하여 배포하는 방법 등이 있다.

본 프로젝트에서 웹 서비스 개발 시 Javascript를 사용하였고 Javascript에서 머신러닝 모델을 이용하기(Load) 위해 Tensorflow.js 프레임워크를 사용하였다. Tensorflow.js 프레임워크는 Javascript 환경에서 머신러닝 모델을 개발 및 학습, 배포가 가능하도록 만들어진 프레임워크이다.

Tensorflow.js에서 모델을 이용하는(Load) 일반적인 방법은 Tensorflow.js API 문서에서 볼 수 있듯이 `tf.loadGraphModel()` 함수와 `tf.loadLayersMode()` 함수를 사용하면 된다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

tf.loadGraphModel() 함수는 Tensorflow의 SavedModel 포맷으로 저장된 모델을 이용할 때 사용하고 tf.loadLayerModel() 함수는 H5 포맷으로 저장된 모델을 이용할 때 사용한다. 본 프로젝트에서 모델은 Tensorflow와 Keras 프레임워크를 이용하여 개발을 진행하였고 Keras를 이용하는 경우 모델이 H5 포맷으로 저장되기 때문에 tf.loadLayersModel() 함수를 사용하였다.


tf.loadLayersModel() 함수를 사용하여 모델을 이용하는 방법으로는 localStorage://와 indexeddb://, HTTP(S), file://와 같은 프로토콜을 사용하여 CORS(Cross-Origin Resource Sharing) 정책을 위반하지 않도록 해야 한다. localStorage와 indexeddb, file 프로토콜의 경우 로컬에 저장된 모델만 이용할 수 있고 HTTP(S)의 경우 모델이 저장된 서버의 URL이 존재한다면 로컬뿐만 아니라 외부 사용자에게도 서비스를 제공할 수 있다.

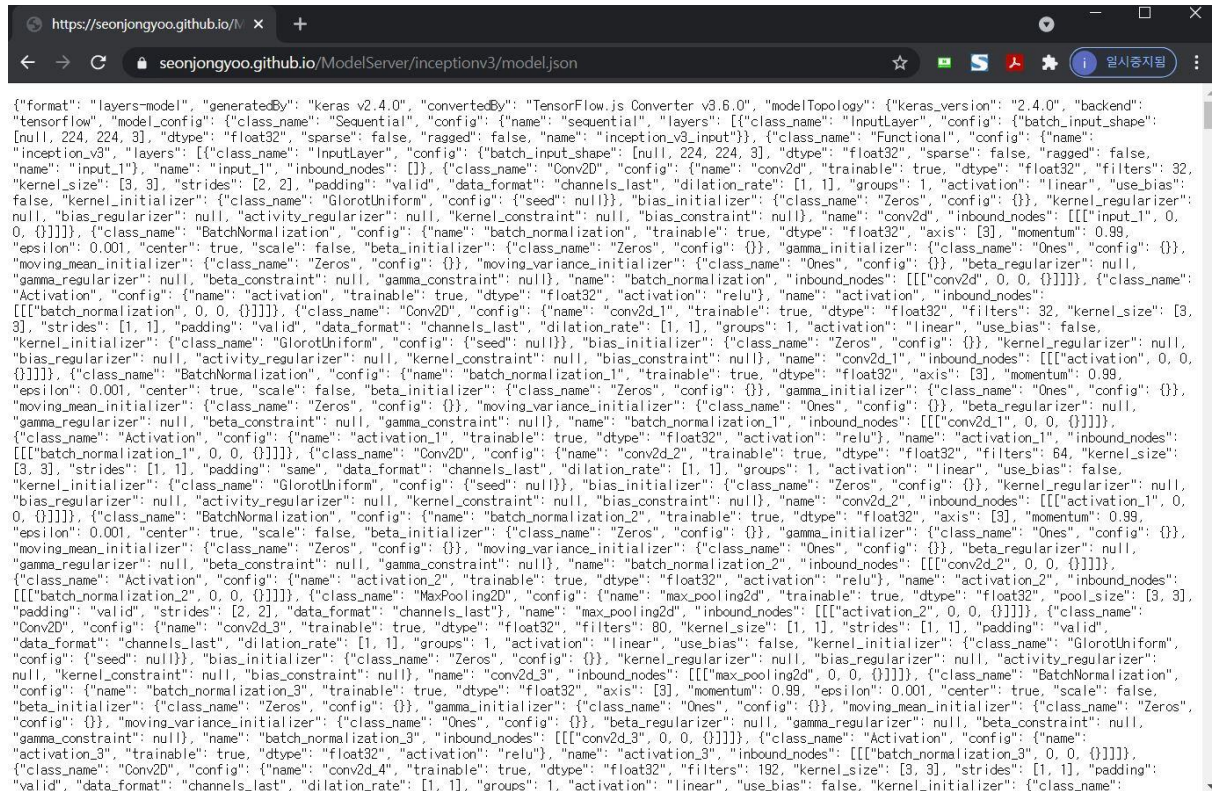
본 프로젝트에서는 외부 사용자에게 개발한 모델의 기능을 제공해야 하므로 HTTP(S) 프로토콜 정책에 맞게 모델을 배포할 수 있는 방법을 고안했다. HTTP(S) 프로토콜을 이용하는 경우

'<http://modelserver.domain/download/model.json>'(공식 홈페이지 예시)

와 같이 모델의 정보가 저장된 json 파일과 이 파일을 저장하고 있는 서버의 HTTP(S) URL이 필요하다.

본 프로젝트는 앞서 웹 서버 구축 시 글 서두에 설명했던 클라우드나 웹 프레임워크를 사용하여 서버를 구축하는 방법이 아닌 GitHub Page 호스팅을 통해 서버를 구축하였기 때문에 이와 같은 방법으로 모델을 저장하는 서버를 구축할 수 있을 것이라 판단하여 모델의 정보를 저장하는 json 파일과 weights 파일들로 GitHub Page를 호스팅 하여 개발을 진행해 보았다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투염	
		Confidential Restricted	Version 1.5	2021-05-26




[그림25] "model.json"의 이름으로 GitHub Page 구성

[그림25]와같이 json 파일로 GitHub Page를 구성하였고 해당 브라우저의 URL을 이용하여 HTTP(s) 프로토콜 정책에 맞게 모델을 배포할 수 있었다.

추가로, tf.loadLayersModel() 함수를 사용하여 모델을 이용하기 전에 H5 포맷을 Tensorflow.js의 포맷에 맞게 json 파일로 변경하는 작업이 필요하다. 이 작업은 Python 환경에서 tensorflowjs 모듈을 이용하면 간단하게 할 수 있다. tensorflowjs 모듈에서 converters API를 이용하면 Keras를 통해 개발 및 학습된 모델에 대한 json 파일과 weights 파일들이 생성된다.

4. 얼굴 합성 기능 개발

팀 내부적인 회의를 한 결과, 추천하고 싶은 헤어스타일을 사용자의 얼굴에 합성하여 헤어스타일을 직관적으로 보여주자는 내용이 나왔다. 하지만 사용자의 얼굴에 헤어스타일만을 합성하게 될 경우, 부자연스러운 결과가 나올 것이라는 의견이 많았다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

그래서 추천하고 싶은 헤어스타일을 가진 얼굴 사진에 사용자의 이목구비를 합성하자는 의견이 나왔다.

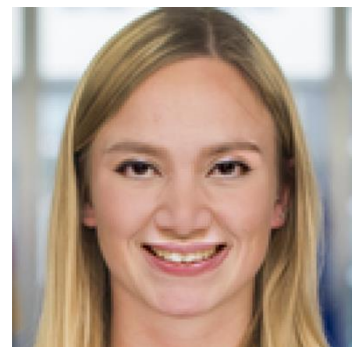
얼굴 합성을 하기 위해서 처음에는 openCV 를 이용하기 위해 Python 을 사용했다. 사용자의 사진과 합성 대상이 될 사진의 landmark 점들을 numpy matrix 에 저장시켰다. 그 후에 사용자의 이목구비를 합성할 사진에 맞게 회전, 크기 조정, 색상 변환을 하고, 합성할 사진에 사용자의 이목구비를 혼합했다. 매핑을 할 때는 openCV 의 wrapAffine 함수를 이용했다. 두 이미지의 간의 피부색과 조명 차이로 인해 중첩된 영역의 가장자리에서 불연속성이 발생하는 것이 문제였다. 이런 문제점을 해결하기 위해서 openCV 의 GaussianBlur() 함수를 이용하여 Gaussian 필터를 이용한 블러 처리를 진행했다. 그 결과 두 이미지의 색상 차이에 대한 문제점을 원만하게 해결했다. 또한 getFaceMask() 함수를 통해 두 이미지의 이목구비를 하얀색으로 나타내고 나머지 부분은 0 의 값으로 표현하여 두 이미지를 혼합한다. 위의 사진은 프로젝트 내에서 개발한 얼굴 합성 알고리즘을 바탕으로 1 번 사진의 얼굴형에 2 번 사진의 이목구비를 합성한 것이다.



[그림26-1]




[그림26-2]




[그림26-3]

26-1과 26-2의 합성 결과


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투뎀	
	Confidential Restricted	Version 1.5	2021-05-26

2.2.2 시스템 기능 요구사항

Functional Requirement	내용	중요도	완료/변경/미완료
FR1	사용자가 URL에 접속하면 "성별 선택" 버튼과 "NEXT" 버튼을 화면에 출력한다.	상	완료
FR2	사용자가 "남자" 버튼을 클릭하면 시스템 내부에 사용자가 남성임을 저장한다.	상	완료
FR3	사용자가 "여자" 버튼을 클릭하면 시스템 내부에 사용자가 여성임을 저장한다.	상	완료
FR4	사용자가 성별 선택을 하지 않으면 시스템은 화면에 "성별 선택" 안내 문구를 출력한다.	상	완료
FR5	사용자가 성별 선택 후 NEXT 버튼을 클릭하면 사진 첨부와 웹캠 사용을 선택하는 페이지를 렌더링한다.	상	완료
FR6	사용자가 사진 첨부와 웹캠 사용을 선택하는 페이지에서 PREV버튼을 클릭하면 성별을 선택하는 페이지를 렌더링한다.	중	완료
FR7	사용자가 "사진 선택"을 클릭하면 사진을 업로드하는 페이지를 렌더링한다.	상	완료
FR8	사용자가 사진을 업로드하는 페이지에서 "업로드"버튼을 클릭하면 OS의 파일 매니저를 실행한다.	상	완료
FR9	사용자가 OS 파일 매니저를 통해 업로드할 사진을 선택하면 사용자가 선택한 사진을 웹페이지 상에 렌더링한다.	중	완료
FR10	사용자가 "확인"버튼을 클릭하면 사용자가 업로드한 사진을 기반으로 얼굴형을 분석하는 페이지를 렌더링한다.	상	완료
FR11	사용자가 사진을 업로드 하지 않고 "확인"버튼을 클릭하면 사진을 업로드 해달라는 알림창을 표시한다.	상	완료
FR12	사용자가 사진이 아닌 다른 파일을 업로드 했을 때 "확인"버튼을 클릭하면 사진을 업로드해달라는 알림창을 표시한다.	상	미완료
FR13	사용자가 사진을 업로드하는 페이지에서 PREV버튼을 클릭하면 사진 선택과 웹캠 사용을 선택하는 페이지를 렌더링한다.	중	완료
FR14	사용자가 업로드한 사진을 기반으로 얼굴형을 분석하는 페이지에서 업로드된 사진을 렌더링한다.	중	완료

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

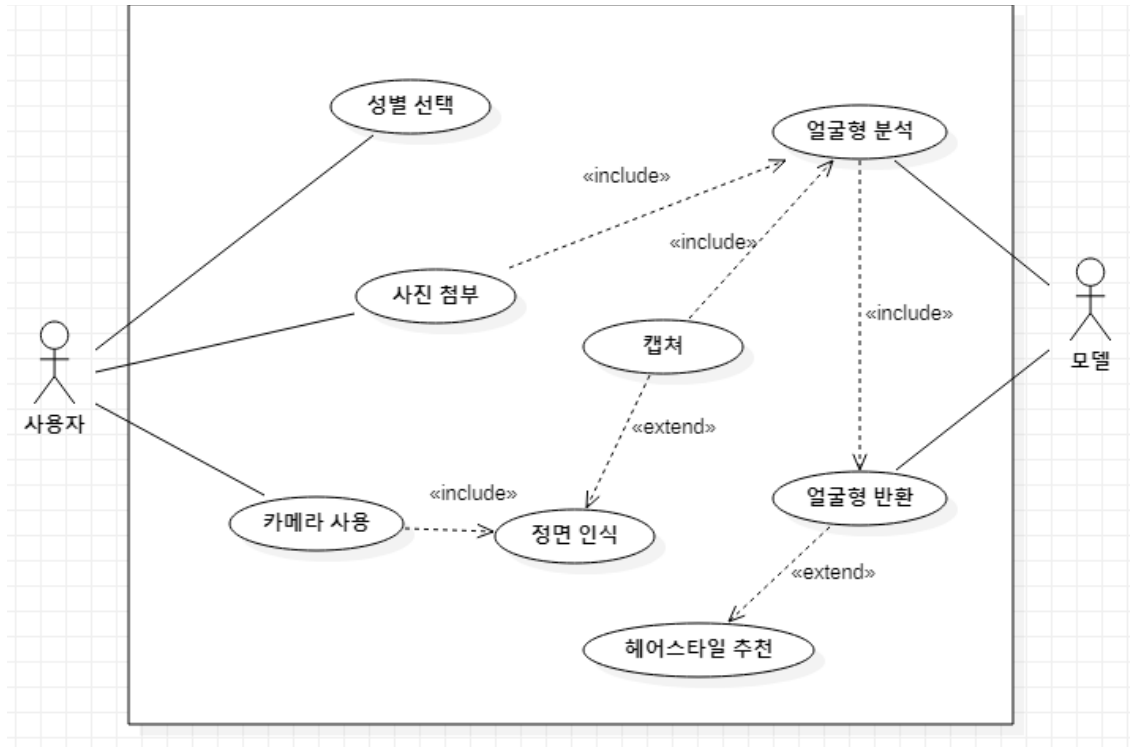
FR15	시스템은 얼굴형이 분석될 때 까지 약 1분 정도 소요되는 텍스트를 렌더링한다.	중	완
FR16	시스템은 사용자가 업로드한 사진 정보에 대해 전처리를 진행한다.	상	완료
FR17	시스템은 얼굴형을 분석하는 모델을 불러들인다.	상	완료
FR18	시스템은 전처리된 이미지를 모델로 넘겨준다.	상	완료
FR19	시스템은 입력된 사진에서 얼굴을 인식한다.	상	완료
FR20	시스템은 인식한 얼굴에서 특징점을 추출한다.	상	완료
FR21	시스템은 각 특징점들에 대해 Marker를 렌더링한다.	중	완료
FR22	모델을 넘겨받은 사진을 기반으로 얼굴형을 분석한다.	상	완료
FR23	모델은 분석된 얼굴형을 시스템으로 반환한다.	상	완료
FR24	시스템은 모델로부터 반환된 값을 기반으로 사용자의 얼굴형을 판단한다.	상	완료
FR25	시스템은 분석된 얼굴형을 사용자에게 TEXT로 표시한다.	상	완료
FR26	사용자가 RESULT 버튼을 클릭하면 분석된 얼굴형을 기반으로 헤어스타일을 추천하는 페이지로 이동한다.	상	완료
FR27	사용자가 얼굴형이 분석되기 전 RESULT 버튼을 누르면 잠시 기다려 달라는 알림창을 표시한다.	상	완료
FR28	사용자의 얼굴형이 계란형이면 추천하는 헤어스타일만 렌더링한다.	상	완료
FR29	사용자의 얼굴형이 계란형이 아니면 하는 헤어스타일과 비추천 헤어스타일을 렌더링한다.	상	완료
FR30	사용자의 얼굴형이 계란형일 때 >>> 버튼을 누르면 추천하는 다른 헤어스타일을 렌더링한다.	상	완료
FR31	사용자의 얼굴형이 계란형이 아닐 때 >>> 버튼을 누르면 추천하는 헤어스타일과 비추천 헤어스타일의 다른 쌍을 렌더링한다.	상	완료
FR32	사용자가 HOME버튼을 누르면 최초 성별을 선택하는 페이지를 렌더링하며 시스템 내부에 저장되어있는 모든 값을 초기값으로 설정한다.	상	완료
FR33	사용자가 "웹캠/카메라"버튼을 클릭하면 웹캠을 사용하는 페이지를 렌더링한다	상	완료
FR34	시스템은 웹캠(카메라)로 읽어 들인 영상을 웹페이지 상에 렌더링한다.	상	완료
FR35	시스템은 읽어 들인 영상을 기반으로 얼굴을 인식한다.	상	완료

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

FR36	시스템은 특징점들을 기반으로 사용자의 얼굴을 정면을 향해있는지 판단한다	상	완료
FR37	시스템은 사용자의 얼굴이 정면을 향해 있지 않으면 얼굴을 정면을 향해 맞춰달라는 텍스트를 렌더링한다	상	완료
FR38	시스템은 사용자의 얼굴이 정면을 향해 있을 경우 인식중이라는 텍스트를 렌더링한다	상	완료
FR39	사용자가 얼굴을 5초간 정면을 향해 있을 경우 시스템은 해당 시점의 프레임을 캡처한다.	상	완료
FR40	시스템은 캡처된 프레임을 기반으로 전처리를 진행한다.	상	완료
FR41	시스템은 모델로부터 얼굴형이 반환될 때 까지 1분 정도 소요된다는 텍스트를 렌더링한다.	중	완료
FR42	시스템은 모델로부터 얼굴형을 반환받으면 1분 정도 소요된다는 텍스트를 지우고 사용자의 얼굴형을 알리는 텍스트를 렌더링한다.	상	완료
FR43	시스템은 헤어스타일 추천 시 추천하려는 헤어스타일과 사용자의 얼굴을 기반으로 합성하여 렌더링한다.	상	변경


FR12 미완료: 기존 업로드 방식에서는 OS의 파일 매니저를 png, Jpg 등의 이미지 파일 형식들만 선택할 수 있도록 설정했었지만, local file path를 읽을 수 없는 이슈 사항이 발생했었다. 해당 사항을 해결하기 위해 js의 reader 객체를 사용하는 방식으로 코드를 수정했는데, 이 때 수정 과정에서 이미지 파일 형식만 선택하는 옵션이 빠진걸 놓치고 배포하여 미완료 사항으로 남게 되었다.

FR43 변경: Python과 OpenCV를 이용해 얼굴 합성 기능을 완성하는데에는 성공하였지만, 자바스크립트 내에서 opencv를 활용하는 것은 서버 사이드에서만 가능한 것과 Python 프로그램을 자바스크립트로 불러오기 위해 사용되는 Flask와 Github Page의 동시 사용이 안되는 문제점으로 합성 기능 대신 추천하는 헤어스타일과 비추천 헤어스타일을 조사하여 해당 이미지를 렌더링하는 방식으로 변경하였다.




[그림 27] 유스케이스 다이어그램

유스케이스명		성별 선택	
설명		서비스를 이용하는 사용자가 첫 페이지에서 성별 선택 버튼을 클릭한다. 선택된 성별의 정보는 시스템 내부적으로 관리되며, 차후 헤어스타일 추천 시의 기준이 된다.	
액터		사용자	
선행조건		사용자가 웹페이지에 접속해야 한다.	
후행조건		사용자의 성별 정보가 정상적으로 저장되어 다음 단계로 진행할 수 있다.	
이벤트 흐름	정상 흐름	액터 1. 사용자가 남자 또는 여자 아이콘을 선택한다. 2. 사용자가 NEXT 버튼을 누른다.	시스템 1.1 시스템은 사용자가 선택한 아이콘 별로 성별 정보를 저장한다. 1.2 선택된 성별에 맞는 아이콘에 선택되었다는 표시를 해준다. 2.1 시스템은 사용자의 성별 정보가 저장되어 있는 것을 확인하고 페이지를 변경한다.
	예외 흐름	2.a NEXT 버튼을 누른 시점에 사용자가 성별을 선택하지 않았을 경우 성별 선택을 요구하는 알림 창을 표시한다.	

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

유스케이스명		사진 첨부	
설명		사용자가 얼굴형을 분석하려는 대상 사진을 첨부하여 등록한다. 사용자의 사진은 Local Path로 URL이 부여된다.	
액터		사용자	
선행조건		사용자가 "사진 선택"을 선택해야 한다.	
후행조건		사용자의 사진 정보가 시스템 내부에서 저장되고, 이미지가 모델에 전달된다.	
이벤트 흐름	정상 흐름	액터	시스템
		1. 사용자가 업로드 버튼을 클릭한다. 2. 사용자가 얼굴형을 분석하고 싶은 사진을 첨부한다. 3. 사용자가 확인 버튼을 클릭한다.	1.1 시스템은 OS의 파일 매니저를 실행한다. 2.1 시스템은 첨부된 사진을 시스템 내부로 가져와 이미지 정보를 저장한다. 2.2 저장된 이미지의 경로로 URL을 부여한다. 2.3 첨부된 이미지를 사용자에게 다시 한번 보여준다. 3.1 시스템은 사용자의 이미지가 첨부된 것을 확인하고 페이지를 변경한다.
	예외 흐름	3.a 확인 버튼을 클릭한 시점에 사용자가 어떠한 사진도 업로드하지 않았을 경우, 사진 첨부를 요구하는 알림 창을 표시한다.	


유스케이스명		카메라 사용	
설명		사용자가 웹캠 또는 카메라를 사용하여 서비스를 이용한다.	
액터		사용자	
선행조건		사용자가 "웹캠/카메라 사용"을 선택하고, 카메라 및 마이크 접근 권한이 허용되어야 한다.	
후행조건		디바이스에 연결된 카메라(웹캠)을 가져와 웹페이지상에 표시한다.	
이벤트 흐름	정상 흐름	액터	시스템
			1.1 시스템이 디바이스(PC, 모바일)에 카메라(웹캠)이 연결되어 있음을 확인한다. 1.2 웹페이지 상에 카메라(웹캠)으로 읽어들이는 영상을 보여준다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

	예외 흐름	X
--	----------	---

유스케이스명		정면 인식	
설명		카메라(웹캠)로부터 읽어 들인 영상을 토대로 사용자의 얼굴이 정면을 향해 있는지를 판단한다.	
액터		사용자	
선행조건		디바이스에 연결된 카메라(웹캠)으로부터 읽어 들인 영상이 웹페이지 상에 표시되어야 한다.	
후행조건		정면 인식이 완료된 시점에 영상의 1프레임을 캡처한다.	
이벤트 흐름	정상 흐름	액터	시스템
		1. 사용자가 정면을 바라본다.	1.1 시스템은 사용자의 얼굴에서 Face Landmark 정보를 얻는다. 1.2 얻어진 Landmark 정보를 토대로 사용자의 얼굴이 정면인지 판단한다. 1.3 사용자의 얼굴이 정면임이 판단되면 "인식중 ..."이라는 텍스트를 렌더링 한다.
		2. 사용자가 정면을 바라보고 5초가 흐른다.	2.1 시스템은 사용자가 정면을 5초 동안 바라봤는지 판단한다.
이벤트 흐름	예외 흐름	1.a 사용자가 정면을 바라보지 않는다면 시스템은 "정면을 향해 주세요"라는 텍스트를 렌더링 한다. 2.a 사용자가 정면을 바라보다 다시 다른 방향을 바라본다면 사용자가 다시 정면을 바라보는 시점에서부터 5초를 카운트한다.	


유스케이스명		캡처	
설명		영상에서 정면이 인식된 사용자의 한 프레임을 캡처한다.	
액터		사용자	
선행조건		사용자가 카메라(웹캠)을 향해 5초 동안 정면을 응시해야 한다.	
후행조건		캡처된 이미지를 모델에서 로드한다.	
이벤트 흐름	정상 흐름	액터	시스템
			1.1 시스템은 영상에서 1 프레임을 캡처 한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

			1.2 캡처된 이미지는 비율은 유지된 채로 시스템 내부에서 크기가 높기와 너비가 결정된다. 1.3 시스템은 캡처된 이미지에 가상의 URL을 부여한다.
	예외 흐름	X	

유스케이스명		얼굴형 분석	
설명		사용자가 업로드한 사진이나 캡처된 이미지에서 얼굴형을 분석한다.	
액터		모델	
선행조건		사용자가 사진을 업로드하거나 영상을 통해 이미지가 캡처되어야 하고, 각 이미지에 URL이 부여된 상태여야 한다.	
후행조건		시스템 내부에 분석된 얼굴형 인덱스를 반환한다.	
이벤트 흐름	정상 흐름	액터	시스템
		1. 사용자로부터 전달받은 이미지를 기반으로 얼굴형을 분석한다.	1.1 시스템은 프로그램 내부에 배포한 모델을 불러온다. 1.2 시스템은 불러온 모델에 사용자가 전달한 이미지를 입력한다. 1.3 시스템은 얼굴형이 분석되는 시간 동안 웹페이지 상에 "인식 중 ..."과 "약 1분 정도 소요됩니다"라는 텍스트를 렌더링 한다.
	예외 흐름	X	

유스케이스명		얼굴형 반환	
설명		모델이 입력받은 이미지를 바탕으로 얼굴형을 분석한 뒤 시스템 내부에 어떤 얼굴형인지에 대한 정보를 반환한다.	
액터		모델	
선행조건		모델에서 얼굴형 분석 작업이 끝나야 한다.	
후행조건		시스템이 얼굴형 인덱스를 기반으로 얼굴형이 결정한다.	
이벤트 흐름	정상 흐름	액터	시스템
		1. 모델이 얼굴형을 분석한 후 시스템에 0~3 사이의 값을 반환한다.	1.1 시스템은 모델로부터 얼굴형 인덱스를 받아온다.


	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

			1.2 시스템은 얼굴형 인덱스를 기반으로 0: 둥근형, 1: 계란형, 2: 역삼각형, 3: 각진형으로 판단한다. 1.3 기존 "인식중 ..." 코멘트 대신 "당신의 얼굴형은 {건네받은 얼굴형}입니다" 텍스트를 웹페이지상에 출력한다.
	예외 흐름	X	

유스케이스명		헤어스타일 추천	
설명		사용자가 분석된 얼굴형을 바탕으로 헤어스타일을 추천받는다.	
액터		사용자	
선행조건		모델로부터 얼굴형이 분석된 이후 사용자가 하단의 RESULT 버튼을 클릭한다.	
후행조건		X	
이벤트 흐름	정상 흐름	액터	시스템
		1. 사용자가 헤어스타일을 추천받는다. 2. 사용자가 하단의 >>> 버튼을 클릭한다.	1.1 시스템은 분석된 얼굴형과 초기 사용자가 선택한 성별에 따라 1쌍의 추천과 비추천 헤어스타일을 렌더링 한다. 2.1 시스템은 기존에 렌더링 된 추천/비추천 헤어스타일과 다른 헤어스타일 이미지를 렌더링 한다.
	예외 흐름	1.a 사용자가 얼굴형이 분석되기 전 RESULT 버튼을 클릭할 경우 "잠시 기다려주세요"라는 알림 창을 표시한다. 이후 얼굴형이 분석된 후 RESULT 버튼을 클릭하면 정상 흐름으로 넘어간다.	

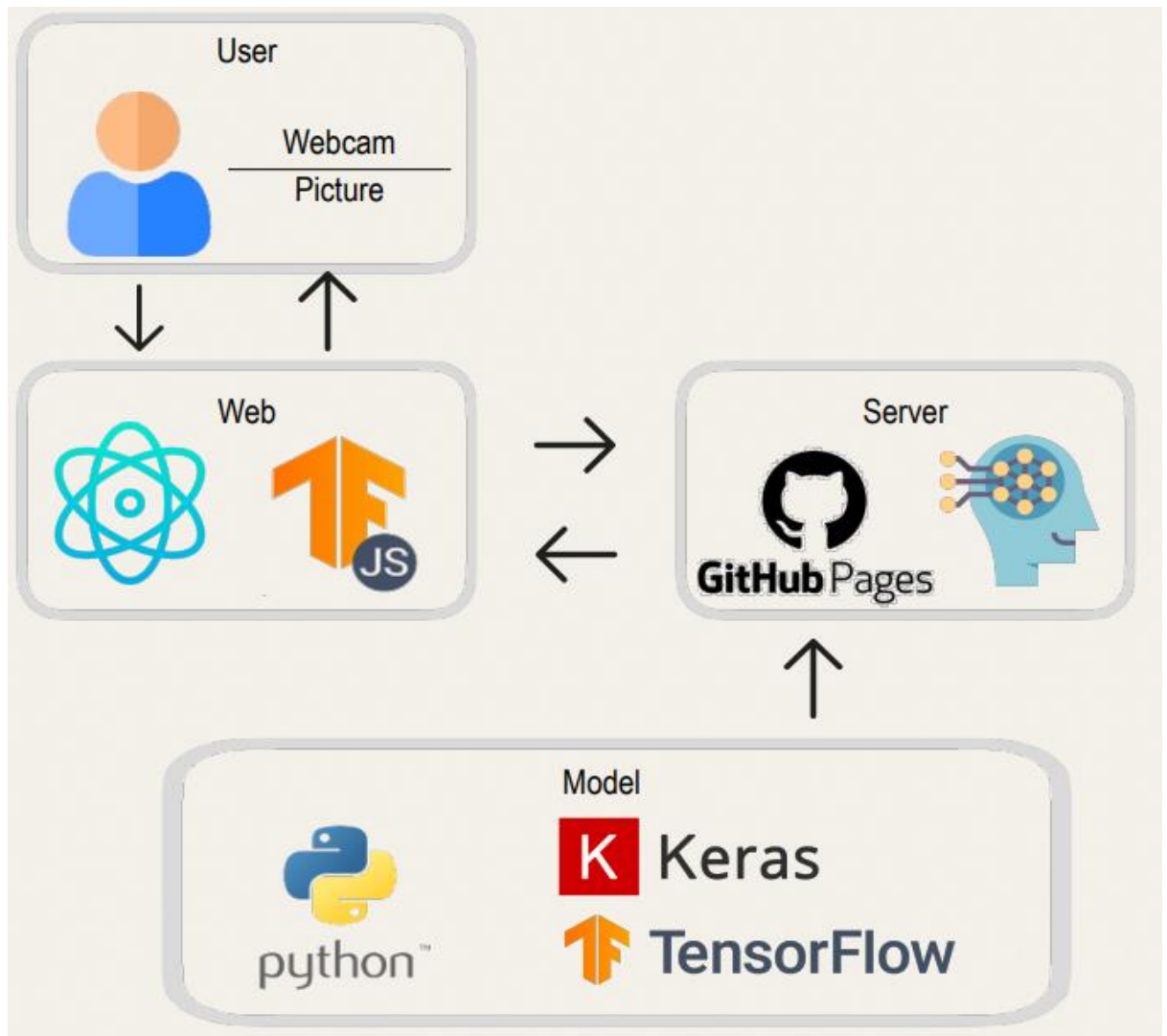
2.2.3 시스템 비기능(품질) 요구사항

Nonfunctional Requirement	내용	중요도 (상/중/하)	완료/미완료
NFR 1	URL요청이 오면 서버는 1분 이내로 요청 받은 페이지를 로드한다.	상	완료
NFR 2	시스템은 예외 발생 시 안내 문구를 출력해야 한다.	상	완료


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

NFR 3	시스템은 입력 받은 사진에 대해 정확하게 얼굴을 인식해야 한다.	상	완료
NFR 4	시스템은 인식한 얼굴에 대해 정확하게 UI를 제공해야 한다.	중	완료
NFR 5	시스템은 저장된 URL에서 모델을 빠르게 로드해야 한다.	상	완료
NFR 6	시스템은 인식한 얼굴에 대해 얼굴형을 정확하게 판단해야 한다.	상	완료

2.2.4 시스템 구조 및 설계도



[그림 28] 시스템 구조도

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


2.2.5 활용/개발된 기술

OS	Window 10
개발 언어	Javascript, Python
개발 도구	Visual Studio Code, Jupyter Notebook, Google Colab
프레임워크	React, Mobx, Keras, Tensorflow.js, OpenCV, Pandas, Danfo.js
서버	GitHub Page



[그림 29] TensorFlow 로고

Tensorflow 는 머신러닝 개발을 위한 프레임워크로 이와 관련된 다양한 API 를 제공한다. Javascript 환경에서 머신러닝 모델을 이용할 수 있는 Tensorflow.js 를 통해 초기 얼굴형 판단 모델을 개발하였고 웹캠에서 사용자의 정면 인식 기능을 개발하였다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26




[그림 30] TensorFlow & Keras 로고

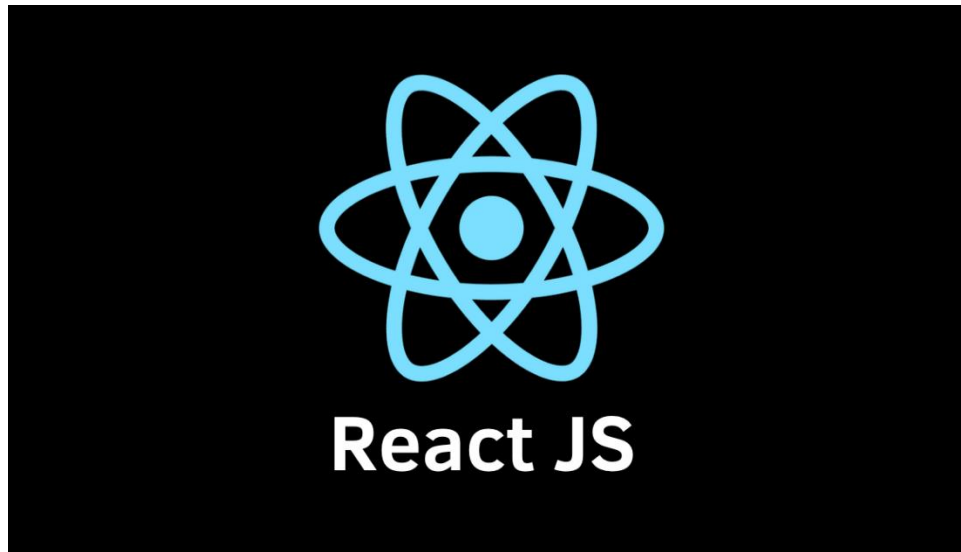
개발 중에 모델 개발과 관련하여 변경된 사항에 대해 Tensorflow Keras 를 통해 검증된 CNN 모델을 이용하였고 Keras API 를 통해 얼굴형 판단 모델을 개발하였다.



[그림 31] OpenCV 로고

OpenCV 는 오픈소스 컴퓨터 비전 라이브러리로 크로스 플랫폼과 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 즉, 객체, 얼굴, 행동, 인식, 모션 추적 등의 응용 프로그램에서 사용하는 영상처리에 관한 라이브러리이다. 이 프로젝트에서는 주로 데이터에 대한 전처리를 하는데 사용했다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26



[그림32] React.js

[React.js]


Front-end에서 주로 사용하는 툴은 'ReactJS'이다. 이는 기존의 JS 언어들과 다르게, 렌더링을 할 때 Virtual DOM이라는 가상 영역에 먼저 수행되며, UI에 렌더링된 페이지와 VirtualDOM에 렌더링된 페이지를 비교하여 수정이 된 부분만 빠르게 렌더링이 들어가 페이지 로딩 같은 UX 부분에서 강점을 가진다.



[그림33] MobX-React

[mobx를 사용한 상태관리]

페이지가 많아질수록 상태를 관리하기 힘들어지는데 mobX-react와 같은 상태 관리 라이브러리를 사용함으로써 상태를 훨씬 효율적으로 관리할 수 있다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

[조건부 렌더링을 통한 페이지 분기]

mobx의 observable 변수를 활용하여 변수의 index에 따라서 조건부로 컨테이너가 렌더링 되게 구현하였다. link tag를 사용하는 전통적인 웹 방식은 새로운 페이지 요청 시마다 정적 리소스가 다운로드되고 전체 페이지를 다시 렌더링 하는 방식을 사용하므로 새로 고침이 발생되어 사용성이 좋지 않다. 그리고 변경이 필요한 부분을 포함하여 전체 페이지를 갱신하므로 비효율적이다.

페이지가 바뀌면 URL이 변경되는 방식이 아니고 한 페이지에서 컨테이너만 바뀌는 SPA(Single Page Application) 방식을 채택함으로써 resource가 load 되는 시간을 단축시키고 user의 접근성을 개선하였다.


[styled-components]

Css 스타일링을 위한 프레임워크이다. SASS 코드를 설치 없이 사용하고 CSS 파일 없이 CSS 코드를 작성할 수 있다. 기존의 className을 사용하지 않는 것과 컴포넌트에 스타일을 적용하여 스타일 코드를 몰아넣을 수 있으며 공통 코드를 줄이고 React의 컴포넌트 형태로 사용하였다. 또한 컴포넌트에 들어가는 props의 상태에 따라 css를 동적으로 관리할 수 있어서 조건에 따른 css의 렌더링도 유동적으로 구현하였다.

[react-reveal]

Fade, Zoom 등 여러 가지 animation을 지원해 주는 라이브러리인 Reveal을 사용하였다.

Reveal의 장점은, react-reveal이 적용된 컴포넌트가 보일 때부터 애니메이션이 시작된다는 것이다. 기존에는 animate.css나 js를 이용한 css 애니메이션을 작성하면 페이지가 onload 됐을 때 한 번에 실행이 돼서 밑에 있는 컨텐츠들은 효과를 못

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

보는 경우가 있었다. 이럴 때 scroll 이벤트를 사용해서 값을 구하고, 특정 스크롤 이상으로 내려갔을 때 class를 추가해서 효과를 내야 했었는데 reveal은 그걸 알아서 동작하게 해준다.

사용자가 스크롤을 내리면서 해당 태그를 보게 되는 순간 적용된 애니메이션이 실행되게 구현하였다.


2.2.6 현실적 제한 요소 및 그 해결 방안

[얼굴 합성 알고리즘]

얼굴 합성 알고리즘은 Python 과 OpenCV 를 기반으로 작성되었다. Python 스크립트를 React 에서 사용하기 위해서는 Flask 웹 프레임워크를 사용해 하나의 서버를 구축해야 했다. 하지만 로보링크 측에서 Github Page 를 통한 React 어플리케이션 배포를 원했는데, Github Page 는 정적 페이지만을 호스팅 할 수 있었기 때문에 Flask 로 연결된 React App 은 동적 페이지로 분류되기 때문에 웹페이지 배포를 위해선 다른 서버가 필요했다. 또 다른 방법으로 JavaScript 에서 OpenCV 를 사용하기 위한 외부 라이브러리 opencv4 nodejs 를 찾아봤는데, 이 또한 Node.js 에서 이뤄지는 방식이었기 때문에 사용하려면 외부 서버가 필요했다. 결과적으로 얼굴 합성 기능은 합성이 아닌 추천하는 헤어스타일과 추천하지 않는 헤어스타일을 출력하는 형태로 변경하였다.

[코로나 19]

코로나 19 로 인해 팀의 모든 인원이 만나서 프로젝트를 진행하지 못했다. 어느 팀이나 비슷한 고충을 겪었겠지만, 우리 팀은 팀원이 5명이어서 한 번도 모든 인원이 다 같이 만나지 못했다. 다 같이 만나지 못하다 보니 개발에 대한 사기가 떨어졌던 순간도 있었다. 그래서 google meeting 을 통해 더 많은 소통을 하도록 노력했다. 또한 Notion 이나 Trello 와 같은 협업 Tool 을 자주 이용하여, 개발에 대한 사기를 높이기 위해 노력했다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

2.2.7 결과물 목록

2.2 연구/개발 내용 및 결과물과 중복되므로 2.2 로 결과물 목록을 대체한다.

2.3 기대효과 및 활용방안

이 프로젝트는 웹캠이 있는 컴퓨터, 스마트폰을 통해서 실시간으로 자신의 얼굴형을 알 수 있고, 카메라가 없다고 하더라도 정면이 나온 사진을 통해 자신의 얼굴형을 파악할 수 있다. 즉 사용자들이 다른 프로그램을 다운로드하지 않더라도, 즉각적으로 자신의 얼굴형을 파악할 수 있다는 장점이 있다. 이렇게 낮은 진입장벽을 통해 다양한 사용자에게 서비스를 제공하는 것을 기대할 수 있다.

헤어 컨설팅



나의 얼굴형과 이목구비, 체형을 72시간 분석하여 단점을 보완함으로써 결과적으로 장점은 더 드러날 수 있는, 나에게 가장 잘 어울리는 헤어스타일을 추천해드립니다.

Best/Worst 스타일을 기본적으로 알려드리고, 나에게 잘 어울리는 기장감이나 앞머리 스타일링 가이드까지 드리기 때문에 Best 라고 해서 한가지만 고수할 필요 없이 어떤 스타일에도 적용해보실 수 있도록 자세히 설명해 드립니다.


헤어컨설팅(남자)는 준비중입니다.

110,000원

[그림 34] 헤어 컨설팅

(출처: 개인별 맞춤 스타일 컨설팅 서비스 – RareLee)

이 서비스를 통해 사용자는 자신의 얼굴형에 대해 정확하게 인지할 수 있다. 헤어 컨설팅은 전문가들이 사용자의 얼굴과 더불어 여러 지표를 활용하여 헤어스타일을 추천해 주지만 많은 비용이 들어간다. 헤어 컨설팅만큼의 효과를 낼 수는 없지만 이 서비스를 이용하면 사용자는 무료로 자신의 얼굴형을 인지할 수 있다. 또한 인지한


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

얼굴형을 바탕으로 어울리는 헤어스타일을 시도하게 되면 헤어스타일을 시도하는 기회비용을 절약할 수 있다.

3 자기평가


이 프로젝트에 대해서는 아쉬움이 남는다. 최종적으로 사용자에게 추천하는 헤어스타일을 사용자의 이미지와 합성하는 것까지 진행할 예정이었다. Python으로 OpenCV를 사용하여 합성하는 것은 성공했으나 웹서비스 내부에서 합성을 진행하려면 Javascript에서 OpenCV를 사용해야 한다. 하지만 결국 Javascript에서 OpenCV를 Python처럼 사용하는 것은 할 수 없어서 이미지 합성에는 실패했다.

조금 더 구체적인 계획을 세워서 개발을 진행했다면 보다 더 완성도 높은 서비스를 개발할 수 있었을 것 같다. GitHub Page를 통한 React 어플리케이션의 배포가 아닌 AWS나 Heroku 같은 클라우드 플랫폼을 통해 배포가 진행되었다면 얼굴 합성 기능도 Flask를 통해 Python을 사용하여 OpenCV를 수월하게 진행해 프로젝트에 추가할 수 있었을 것이라는 아쉬움이 남는다.


	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

4 참고 문헌


번호	종류	제목	출처	발행년도	저자	기타
1	서적	리엑트를 다루는 기술 (개정판)	(주)도서출판 길벗	2019.09	김민준	기술 서적
2	기술 문서	Tensorflow.js	https://www.tensorflow.org/js?hl=ko			Tensorflow.js 공식 API 문서
3	웹페이지	Difference between tfjs_layers_model and tfjs_graph_model	https://stackoverflow.com/questions/55829043/difference-between-tfjs-layers-model-and-tfjs-graph-model			tensorflow.js 에서 model 이용하기
4	웹페이지	Face Shape Dataset	https://www.kaggle.com/niten19/face-shape-dataset			얼굴형 Dataset
5	웹페이지	Flickr-Faces-HQ Dataset (FFHQ)	https://github.com/NVLabs/ffhq-dataset			얼굴 사진 Dataset
6	웹페이지	MediaPipe	https://google.github.io/mediapipe/solutions/models			Face-Landmarks-Detection 관련 자료
7	웹페이지	[차홍뷰티] 얼굴형에 맞는 헤어스타일 hair styles for face type	https://www.youtube.com/watch?v=n-dnQ1-J_Wo&t=127s			얼굴형별 추천, 비추천 헤어스타일 구분
8	웹페이지	넙대대한 얼굴인 여성분들께 맞는 헤어스타일 추천 (+ 피해야할 스타일)	https://www.youtube.com/watch?v=KVJ1K4zu1HM			얼굴형별 추천, 비추천 헤어스타일 구분

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

9	웹페이지	자세 추정 MLP 모델 개발	https://hongchan.tistory.com/3			Tensorflow.js 분류 모델 개발
10	웹페이지	Switching Eds: Face swapping with Python, dlib, and OpenCV	http://matthewearl.github.io/2015/07/28/switching-eds-with-python/	2015.07		얼굴 합성 알고리즘
11	기사	헤어스타일만 바꿔도 '훈남'..."얼굴형에 맞게 바꿔봐"	https://news.mt.co.kr/mtview.php?no=2015072909231160019	2015.07	유연수	얼굴형별 추천, 비추천 헤어스타일 구분
12	논문	얼굴형의 특성을 보완한 헤어스타일 연구	한국교육학술정보원 http://www.riss.kr/link?id=T13373759&outLink=K	2014.02	주연빈	얼굴형 구분 기준 설정 자료
13	논문	얼굴형에 따른 헤어스타일연구	https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=JAKO201534736521217&oCn=JAKO201534736521217&dbt=JAKO&journal=NJOU00579511	2015.05	탁원우, 김판진	얼굴형에 따른 헤어스타일 조사
14	논문	재실자_활동량_산출 을_위한_Pose_분류_ 모델_개발	https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07549379&language=ko_KR	2018.10	최은지, 박보람, 최영재, 문진우	분류 모델 개발
15	논문	Face shape classification using Inception V3	https://github.com/adonistio/inception-face-shape-classifier/blob/master/Paper%20v3.pdf	2019.11	Adonis Emman uel Tio	얼굴형 분류 모델
16	웹페이지	pre-trained VGG 를 사용한 blood cell classification	https://eremo2002.tistory.com/61?category=779320			VGG16 transfer learning
17	웹페이지	Overfitting 과 Underfitting	https://brunch.co.kr/@gimmesilver/44#comment			오버피팅 개념

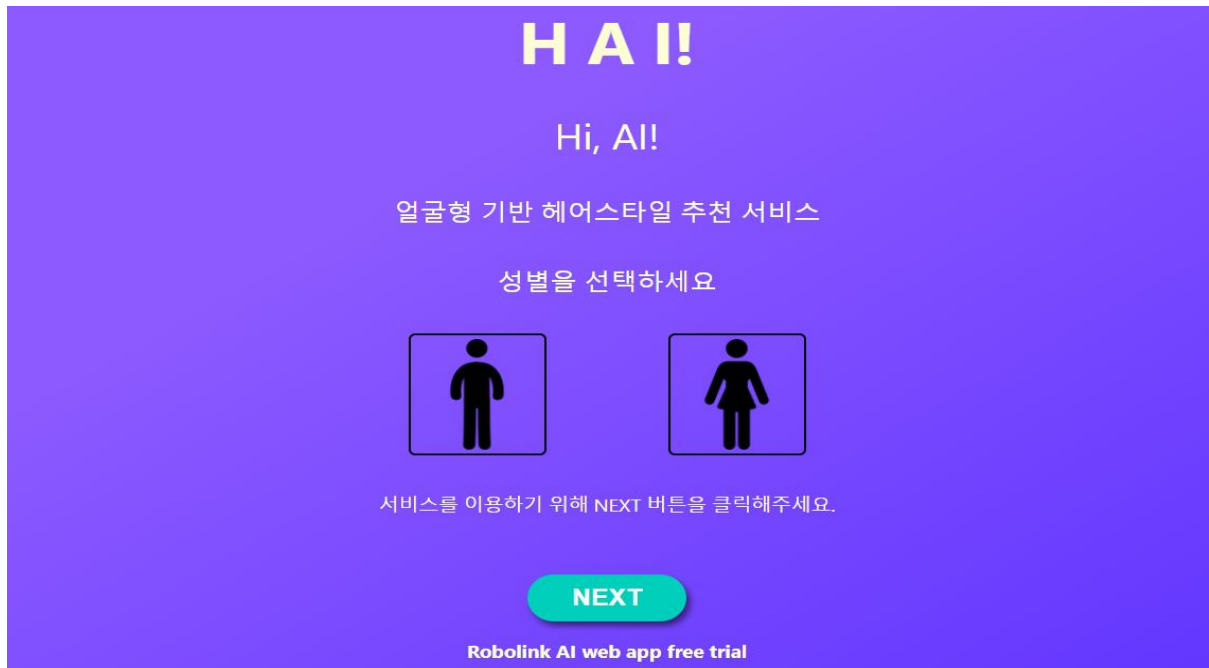
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

18	웹페이지	Data Preprocessing & Data Augmentation	https://nittaku.tistory.com/272			Data Augmentation 방법
19	기술 문서	Keras ImageDataGenerator	https://keras.io/api/preprocessing/image/			Data Augmentation 을 위한 ImageDataGenerator 사용
20	기술 문서	Tensorflow.js 가이드	https://www.tensorflow.org/js/guide?hl=ko			Tensorflow.js 사용 방법
21	기술 문서	MediaPipe Face Mesh	https://drive.google.com/file/d/1VFC_wlpw4O7xBOiTgUldI79d9LA-LsnA/view	2019.01	Artsiom Ablavatski, Ivan Grishchenko, Yury Kartynnik	Face Landmarks Detection 개발에 사용된 모델

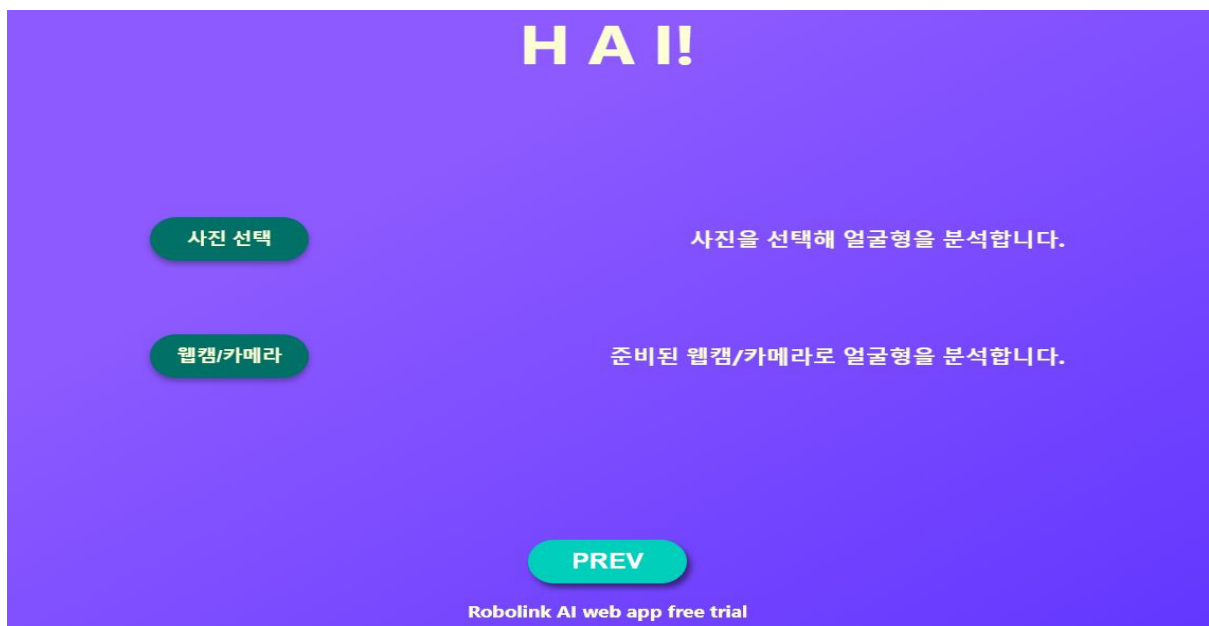
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

5 부록


5.1 사용자 매뉴얼



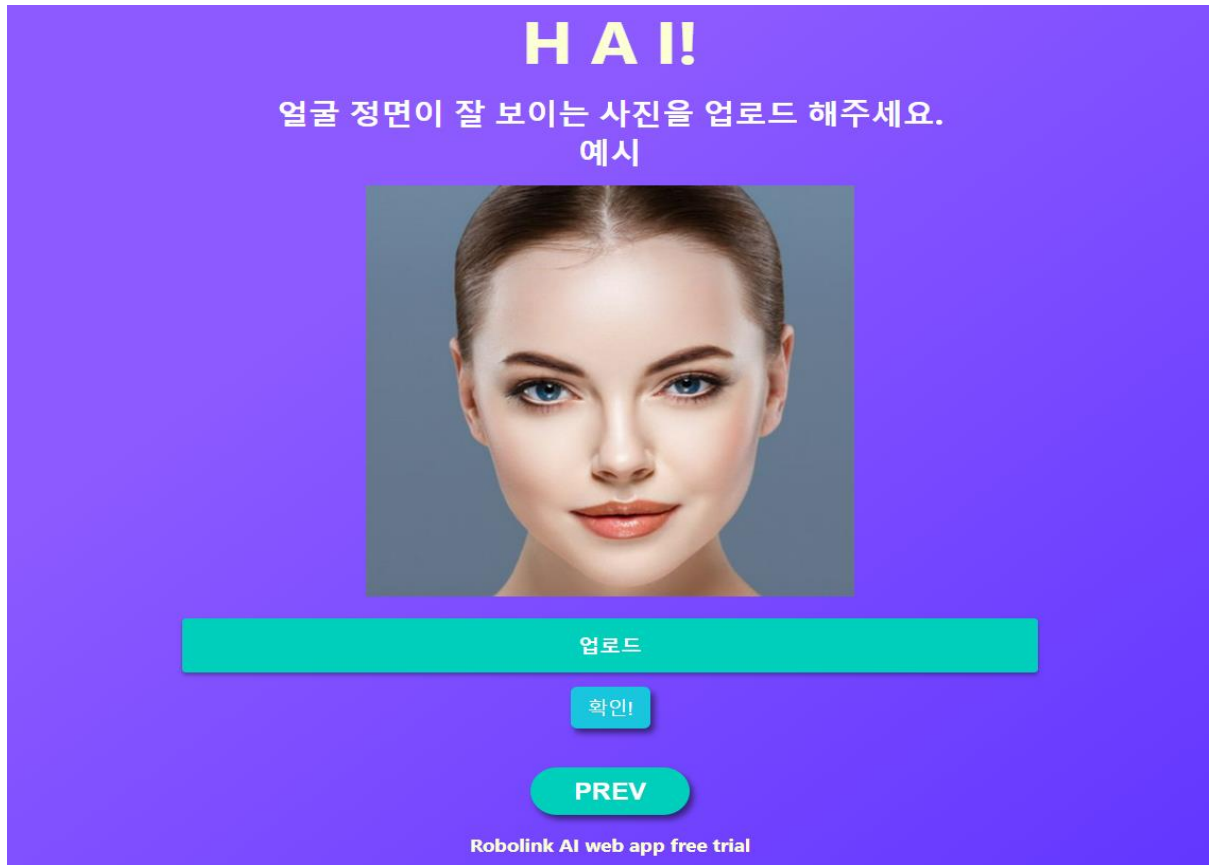
Hosting된 Github.io URL에 접속하면 위와 같이 시작하게 된다. 각자 성별에 맞게 선택을 하고 NEXT를 누르면 다음 페이지로 넘어간다.



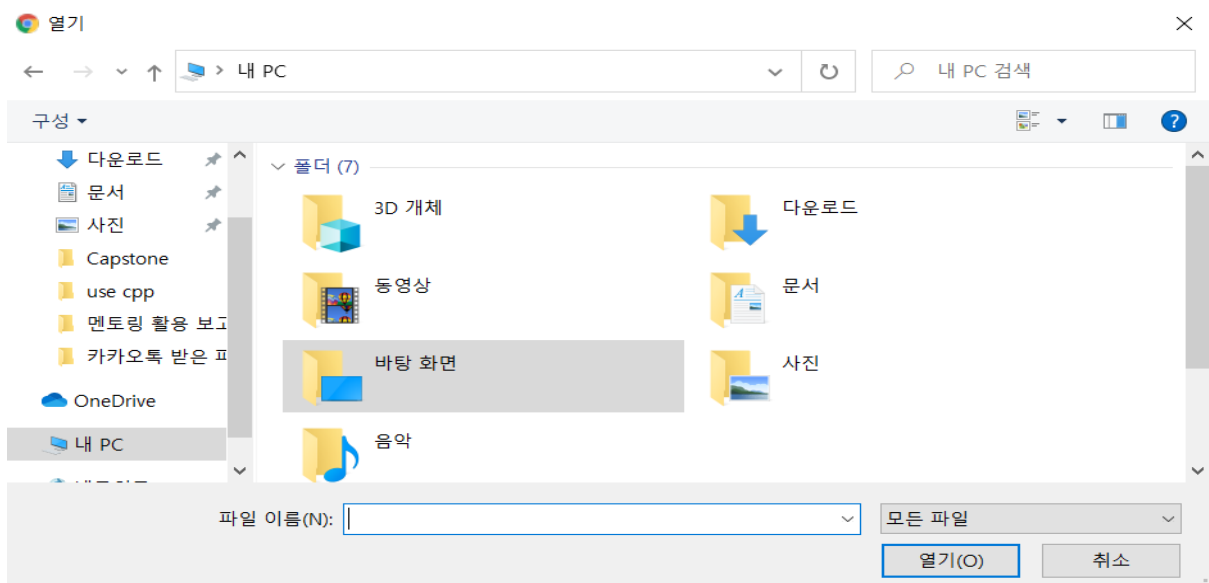
웹캠이 없는 사용자는 사진 선택으로 얼굴형을 확인할 수 있고, 웹캠이 있는 사용자는 웹캠/카메라 버튼을 눌러 실시간 웹캠을 사용해 얼굴형을 확인할 수 있다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


[사진 선택 버튼 클릭 시]



예시와 비슷하게 정면을 보고 있는 사진을 업로드 해야 한다. 사진을 업로드 하기 위해 업로드 버튼을 누른다.




업로드 버튼을 누르면 OS File Manager가 실행된다. 원하는 이미지를 선택하고 열기 버튼을 누른다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26



파일을 선택하면 이미지가 등록되고 이후 얼굴형 분석을 시작한다.

	결과보고서		
	프로젝트 명	HAI!	
	팀 명	무투멍	
	Confidential Restricted	Version 1.5	2021-05-26

HAI!


당신의 얼굴형은 **역삼각형**입니다.



RESULT


Robolink AI web app free trial

얼굴형 분석이 완료되면 자신의 얼굴형을 확인하고 "RESULT" 버튼을 누르면 결과창으로 이동한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


당신의 얼굴형은 **역삼각형**입니다.

추천 헤어스타일



얼굴 면적을 작게 해주고 균형감을 주는 덴디컷

비추천 헤어스타일




윗 볼륨과 뒷 기장이 길어져 세로폭이 부각되는 테일컷 비추천

>>>

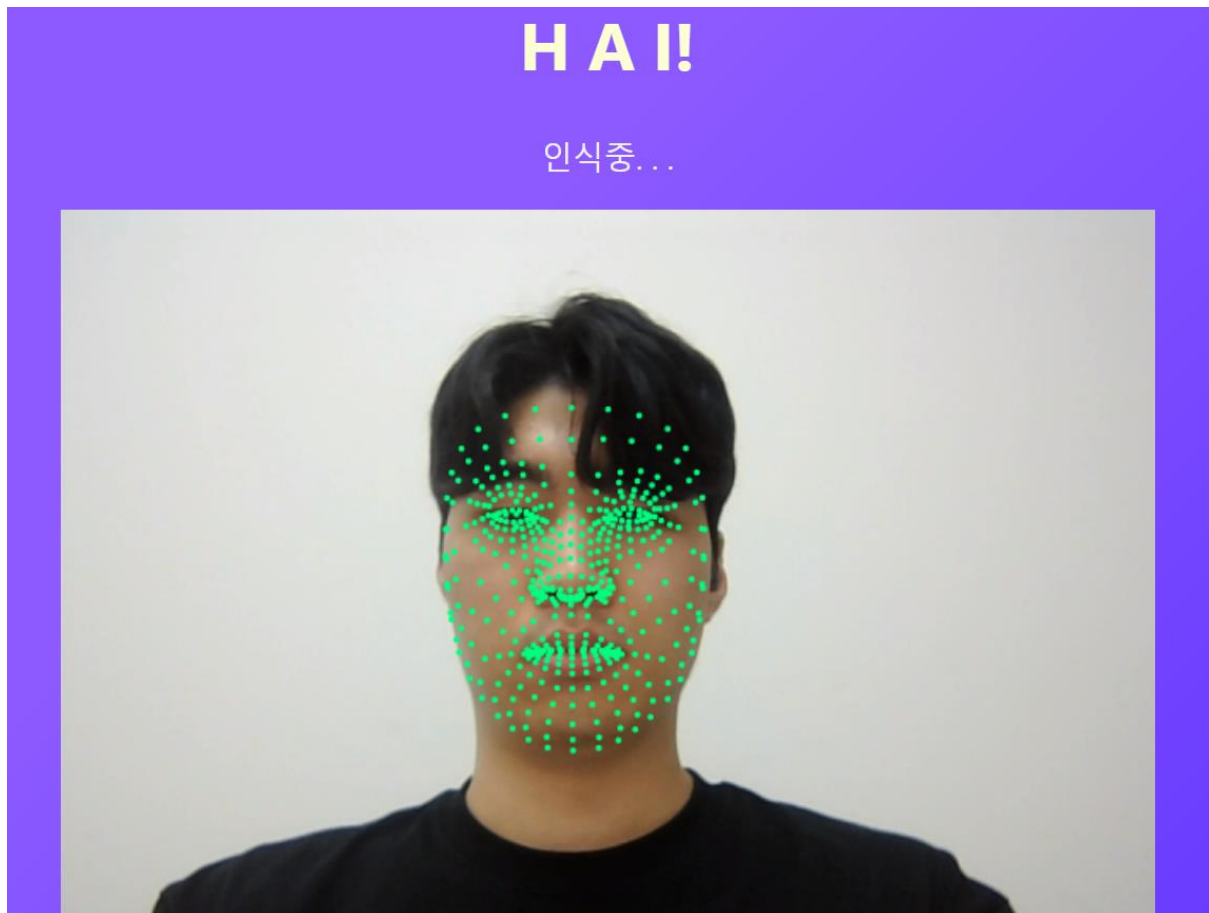
HOME

Robolink AI web app free trial


자신의 얼굴형과 얼굴형에 추천하는 헤어스타일과 추천하지 않는 헤어스타일을 확인한다. ">>>" 버튼을 누르면 얼굴형에 맞는 다른 추천, 비추천 헤어스타일로 바뀐다.

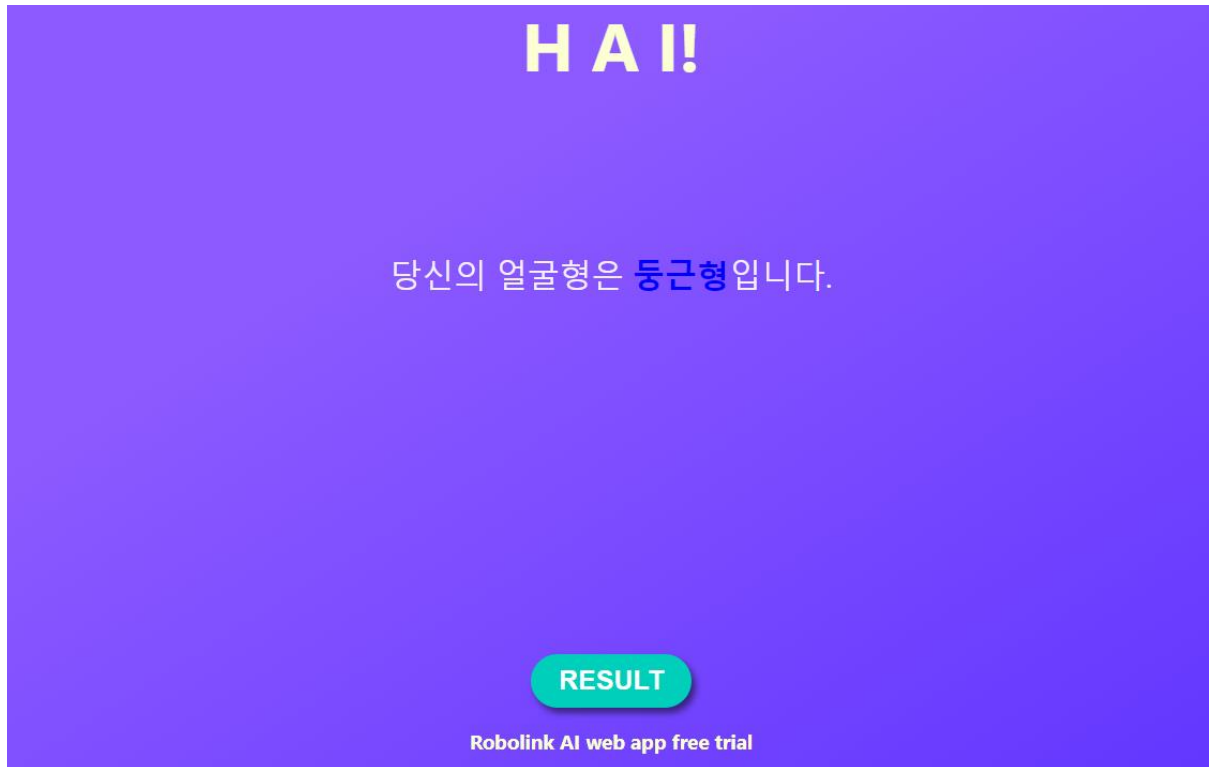
	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

[웹캠/카메라 버튼 클릭 시]




자신의 웹캠에 얼굴을 인식한다. 이 때 정면을 보고 있지 않다면 “얼굴을 정면을 향해 맞춰주세요.” 라는 멘트가 출력된다. 정면을 잘 바라본다면 “인식중...” 이라는 멘트와 함께 5초 후 얼굴형 분석을 시작한다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26

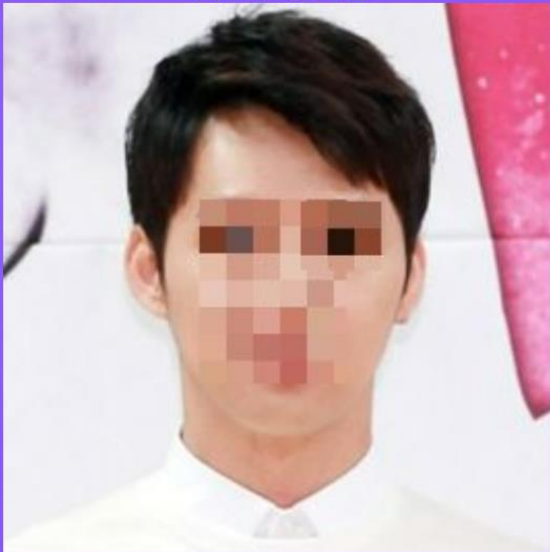


실시간 웹캠으로 사용자의 얼굴을 인식해 분석한 얼굴형을 알려준다. "RESULT" 버튼을 클릭하면 얼굴형에 맞는 추천, 비추천 헤어스타일을 확인할 수 있다.

	국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
		프로젝트 명	HAI!	
		팀 명	무투멍	
		Confidential Restricted	Version 1.5	2021-05-26


당신의 얼굴형은 **둥근형**입니다.

추천 헤어스타일



가르마를 타서 자연스러운 웨이브 연출

비추천 헤어스타일



얼굴이 짧아 보이고 가로를 부각시키는 눈썹 가리는 헤어 비추천

>>>

HOME

Robolink AI web app free trial

자신의 얼굴형에 추천하는 헤어스타일과 추천하지 않는 헤어스타일을 확인할 수 있다.

5.2 배포 가이드

<http://cha-y-s.github.io/HAI>

- PC / 모바일 환경에서 Chrome 으로 접속하길 권장한다.
- "사진 선택" 시 파일 접근 권한을 허용해야 한다.
- "웹캠/카메라"시 카메라 및 마이크 접근 권한을 허용해야 한다.