



국민대학교
소프트웨어융합대학
소프트웨어학부


캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	<i>My_Mood_Music #마무무</i>
팀 명	<i>포춘텔러</i>
문서 제목	중간보고서

Version	1.6
Date	2019-04-18

팀원	오윤정(조장)
	박정은
	원수빈
	홍자현
	강은채
지도교수	최 준수 교수

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

CONFIDENTIALITY/SECURITY WARNING

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

문서 정보 / 수정 내역


Filename	중간보고서-My_Mood_Music.doc
원안작성자	오윤정, 박정은, 원수빈, 홍자현, 강은채
수정작업자	오윤정, 박정은, 원수빈, 홍자현, 강은채

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-02	강은채	1.0	최초 작성	프로젝트 목표 및 계획서 상의 연구내용
2019-04-04	전원	1.1	내용 작성	각자 맡은 역할의 수행내용
2019-04-09	전원	1.2	내용 수정	각자 맡은 역할의 수행내용, 고충
2019-04-11	전원	1.3	내용 작성	각자 맡은 역할의 수정 사안
2019-04-13	전원	1.4	내용 작성	향후 계획
2019-04-16	전원	1.5	내용 수정	내용 수정 및 서식 수정
2019-04-17	전원	1.6	내용 수정	서식 수정 및 작업물 인쇄

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

목 차

- 1 프로젝트 목표
- 2 수행 내용 및 중간결과
 - 2.1 계획서 상의 연구내용
 - 2.1.1 분석 및 노래 추천을 위한 API 서버
 - 2.1.2 얼굴 이미지를 이용한 감정분석 단계
 - 2.1.3 음성 정보를 이용한 감정분석 단계
 - 2.1.4 분석 결과 및 노래 추천 단계
 - 2.1.5 어플리케이션 개발
 - 2.2 수행내용
 - 2.2.1 Back-End
 - 2.2.2 Front-End
- 3 수정된 연구내용 및 추진 방향
 - 3.1 수정사항
 - 3.1.1 Back-End
 - 3.1.2 Front-End
4. 향후 추진계획
 - 4.1 향후 계획의 세부 내용
 - 4.1.1 Back-End
 - 4.1.2 Front-End
- 5 고충 및 건의사항

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서	
	프로젝트 명	My_Mood_Music #마무무
	팀 명	포춘텔러
	Confidential Restricted	Version 1.6 2019-APR-19


1 프로젝트 목표

현대인의 우울증 문제가 날로 심각해지고 있다. 과도한 경쟁 사회 속에서 살고 있는 현대인들은 각박한 사회 속에서 살아남기 위해 자신의 감정을 돌보지 못하고 있는 실정이다. 이런 자기 인식의 부재로 인해 요즘 현대인들은 감정 표현에 서투를뿐만 아니라 또한 몹시 어색해한다. 이런 우울증, 소통 어려움 등의 문제가 더 심화되기 전에, 자신의 감정을 표출할 수 있는 기회가 있어야할 것이다. 하지만 갑자기 자기 자신을 돌아보고 감정에 대해 인식하게 되는 것은 쉽지 않은 일이다.

따라서 본 프로젝트는 사용자에게 감정 분석 결과를 알려주고 이를 바탕으로 노래를 추천해주는 어플리케이션 제작을 목표로 한다. 감정분석 결과는 얼굴인식을 통한 감정분석, 음성인식을 통한 억양분석 결과를 종합해 도출한다.

어플리케이션 구성도는 다음과 같다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

감정이란 어떤 현상이나 일에 대하여 일어나는 마음이나 느끼는 기분으로, 감정을 파악하기 위해서는 감정을 정교하게 표현하는 언어적 수단인 감정단어가 있다. 감정단어의 효과적인 의미 전달을 위해 표정, 억양, 강세, 빠르기 등의 준언어적 요소가 말소리에 반영되어야 표현의 강도를 극대화할 수 있음을 강조하였다(Owens, 2005; Van Rheenem & Russell, 2013). 우리 시스템은 여러 요소 중 표정과 음성을 통해 감정을 분석하고, 더 나아가 그 결과를 활용해 사용자에게 유용한 음악을 추천해준다.


핵심적인 개발 내용은 다음과 같다.

- (1) 분석 및 노래 추천을 위한 API 서버
- (2) 얼굴 이미지를 이용한 감정분석
- (3) 음성 데이터를 이용한 감정분석
- (4) 분석한 감정에 맞는 노래 추천 알고리즘
- (5) 애니메이션을 활용한 어플리케이션

우리 시스템에서 얼굴 이미지 관련 API 를 통해 얻어야 하는 정보는, 이미지 속 얼굴의 추정 나이와 얼굴에서 분석된 감정들이다. 이 조건에 부합하는 API 인 Microsoft Face API 을 선택하였다. Face API 는 입력된 비전 데이터를 통해 얼굴을 인식하거나 얼굴 감지를 이용한 애플리케이션을 만들 때 유용한 API 다. Google Cloud Vision 와 Clova Face Recognition 은 얼굴 이미지에 분석되는 감정이 한 가지 value 로 정의되지만 Microsoft Face API 는 다양한 감정에 대한 정도를 분석해주어 좀 더 정밀한 결과를 도출할 수 있을 것이다.

2.1.1 분석 및 노래 추천을 위한 API 서버

어플리케이션에서 직접 API 나 DB 와 데이터를 주고받을 때 생길 수 있는 부하를 줄이고자 하는 목적으로 API 서버를 만든다. 서버는 Face API 에 사용자의 얼굴 이미지를 넘겨주어 그 결과로 분석된

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

감정을 받는다. 그리고 서버는 DB 의 데이터를 가져와 감정분석의 최종결과를 도출하고 알고리즘을 통한 노래 추천을 수행한다. 이 목표에 부합하는 서버 개발 프레임워크로 Django REST framework 를 사용한다.

2.1.2 얼굴 이미지를 이용한 감정분석 단계

어플을 처음 실행하게 되면 캐릭터가 나타난다. 캐릭터는 더블클릭으로 활성화되고, 어플의 사용흐름을 안내한다. 첫 번째 단계인 얼굴 인식 단계에서, 우선 입력 이미지를 얻기 위해서 카메라 혹은 갤러리를 사용한다. 카메라를 사용할 경우 사용자는 자연스러운 상태로 얼굴을 찍게 되고, 갤러리를 사용할 경우 사용자가 원하는 이미지를 선택하게 된다. 이렇게 결정된 얼굴 이미지는 API 서버로 보내지고, 다시 API 서버에서 Face API 에 입력되어, 얼굴에서 분석된 감정 데이터를 API 로부터 얻는다.


2.1.3 음성 정보를 이용한 감정분석 단계

이 단계에서는 음성데이터를 이용한 감정분석을 수행함으로써, 얼굴 사진을 이용한 감정분석의 정확도를 좀 더 높이는 것을 목표로 한다. 캐릭터가 사용자에게 질문을 하면 사용자는 대답하고, 이 대답 음성이 인식되면 미리 학습해둔 모델에서 감정분석 결과를 얻게 된다. 이 결과는 얼굴 인식 결과와 종합되어 감정 분석 최종결과에 기여한다. 학습데이터는 IEMOCAP 으로부터 얻고, Microsoft API 에서 분류하는 감정과 같은 사이즈의 감정벡터로 분류하는 모델을 만드는 것을 목표로 한다. 구체적인 내용은 다음과 같다.

직접 학습 모델을 만들기 위해서 data 를 수집한다. IEMOCAP 에서 감정분석을 위한 음성 dataset 를 사용하여 학습시킨다. IEMOCAP 에서 제공받은 data 는 그 길이가 다양하다. 총 12 시간 분량에 해당하는 data 로 기존의 2 명의 배우가 연기한 음성 data 에서 24 명의 배우가 더 정교한 목소리연기와 텍스트를 제공한다. 활용하는 dataset 은 소음과 잡음이 거의 없다. 따라서 딥러닝으로 학습 모델을 만들어 나갈 때에는 데이터 전처리 하는 과정이 최소화될 수 있다. 실제로 어플리케이션 사용자의 음성을 인식하여 처리할 때에는 ‘푸리에변환’ 방식을 더한다. 음성을 주파수 성분으로 분해하여 음높이를 파악해 감정을 분석하게 된다.

사용자가 입력하는 음성 data 를 마이크로소프트 FACE API 가 설정한 ‘anger’, ‘contempt’, ‘disgust’, ‘fear’, ‘happiness’, ‘neutral’, ‘sadness’, ‘surprise’ 8 가지로 분류한다. 이를 위한 학습 모델은 tensorflow 로 구현이 된다.

이때에 어떤 모델 형식을 사용할 것인가 고민을 하게 된다. 캐릭터가 “오늘 기분 어때? 오늘 기분 어땠어?” 라고 질문을 던졌을 때, 사용자의 대답이 짧거나 처음부터 끝까지 일관된다면 CNN 방식을

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

채택하여 입력되는 음성을 일정한 길이로 쪼개어 처리할 수 있다. 하지만 사용자가 본인의 감정을 길게 서술할 때에 처음의 감정이 마지막까지 어떻게 이어지는지 분석을 해야한다면 RNN 방식을 채택할 수 있다. 사용자에게 강제로 음성 data의 길이를 제한하지 않는 한, 두 가지 모델형식을 모두 고려하여 학습이 이루어 진다.

이렇게 학습된 모델은 서버상에 올려지게 되고 여기서 나오는 결과는 이미지 분석을 통한 감정 분석 결과의 정확률에 힘을 실어준다.

2.1.4 분석 결과 및 노래 추천 단계

A. 감정 분석 결과 열람 단계

Face API에서 가져온 이미지 감정 분석 결과와 학습된 모델을 통해 도출된 감정결과를 통합한다. 두 가지의 결과를 어떻게 통합할 것인지를 고민하게 된다. 음성데이터 결과가 Face API 결과의 정확률에 힘을 실어주어야 한다. 만약 Face API의 정확률이 충분히 높게 나온다면 음성데이터의 분석 결과를 더 적은 비율로 반영하여 통합된 결과를 도출한다. 혹은 Face API의 결과와 같은 비율로 반영할 수 있는데 이는 상황에 따라 개발과정에서 정하기로 한다.


B. 노래 추천 및 선택 단계

감정분석 결과가 나오면 데이터베이스에서 그 감정에 해당하는 노래 리스트 중 하나를 가져오게 되고 그 방법은 추천 알고리즘을 따른다. 첫번째로는 감정별로 분류된 노래 리스트에서 랜덤함수를 사용한다. 두번째로 사용자의 정보를 저장하는 USER TABLE를 생성하여 분석된 감정과 연령을 저장하게 되고 사용자는 만족도 조사를 하게 된다. 그 정보로 기계 학습을 시켜서 다음 노래추천에 사용자의 만족도를 높이는 결과를 얻기위해 어플리케이션을 개발할 수 있다. 추천되는 노래는 우리가 임의로 리스트를 지정할 수 있고, 단어를 통해 노래를 분류해주는 TF-IDF 알고리즘 방식 등이 있을 것이다. 자세한 추천 알고리즘의 내용은 개발과정에서 정하기로 한다.

2.1.5 어플리케이션 개발

애니메이션이 잘 이루어지는지 확인하면서 개발할 수 있도록 unity로 개발한다.

시작화면은 스플래시 이미지로 한다. 이미지가 사라지고 나면 아직 깨어나지 않은 캐릭터가 화면

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

상에 나타나있다. 배경은 포켓몬고처럼 카메라 배경 혹은 집 안 같은 3d 모델로 한다. 잠들어있는 캐릭터는 더블클릭을 통해 깨울 수 있다. 캐릭터가 깨어나면 캐릭터는 인사하는 애니메이션과 함께 어플에 대한 소개와 흐름을 이끌어가는 역할을 한다.

어플 소개 후에는 얼굴 인식을 하기 위해서 카메라로 이미지를 찍을 것인지 혹은 내 갤러리에 있는 사진을 사용할 것인지 사용자가 선택할 수 있다. 갤러리 접근은 안드로이드로 해야하기 때문에 유니티 에셋스토어에 존재하는 native android asset 을 사용한다.

이렇게 사진을 찍거나 이미지를 정하고나면, 캐릭터가 사용자에게 몇가지 질문을 건넨다. 예를 들어, "오늘 기분은 어떨까요?", "오늘의 하루는 어땠나요?" 가 될 수 있다. 만약 들어온 음성데이터가 너무 작다면 더 자세하게 이야기를 해줄 수 있냐는 식의 대화를 통해 더 나은 음성을 얻어 음성 인식을 할 수 있게 할 것이다.

그렇게 일련의 정보 수집 단계가 끝나고, 만약 정보를 처리함에 있어서 시간이 어느정도 소요된다면 캐릭터가 지금 분석을 하는 중이라는 모션을 취하게 한다. 그게 아니라면 팝업으로 감정 분석에 대한 결과를 표시해주고, 해당 결과에 대한 작은 코멘트를 해준다. 예를 들어, 기쁨이 크게 나왔다면 "당신의 감정에 기쁨이 큰 비중을 차지하고 있습니다. 이 노래를 들으며 행복해 지길 바랄게요!!" 가 될 수 있다. 그리고 나서 감정 분석 정보에 맞춘 노래를 추천해줄 것이다. 그 노래에 해당하는 유튜브 영상을 띄우게 되는데 유튜브 영상을 띄우는 것 또한 유니티에 에셋으로 존재하기 때문에 에셋을 활용한다.


2.2 수행내용

2.2.1 Back-End

A. server

Django 애플리케이션을 MySQL 과 연동 후 프로젝트를 생성하였다. DB 테이블이 완성되기 전, Emotion 테이블과 Music 테이블을 임의로 생성하여 migrate 한 후 admin 사이트에서 확인하였다. 그리고 서버에서 Microsoft Face API 를 사용하여 로컬에 있는 사진을 분석할 수 있는 함수형 뷰를 생성하였고, 그에 대한 결과를 HttpResponse()로 보여줄 수 있게한 후 결과를 보여줄 url 을 설계하였다.

기존 Django 를 이용한 웹 개발에서 Django ORM 의 Queryset 은 Django template 으로 넘겨지며, HTML 로 렌더링되어 Response 로 보내지게 됨을 확인했다. (실험용으로 만든 Bookmark 앱의 코드를

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

보면 알 수 있다.) 하지만 JSON 으로 데이터를 보내야 하는 RESTful API 는 HTML 로 렌더링 되는 Django template 을 사용할 수 없다. 그래서 Queryset 을 JSON 으로 매핑하는 과정을 거쳐야 하는데, 이를 위해 Django REST framework 의 serializer 를 이용한다.

이후 어플(Unity)에서 WebRequest 로 POST 를 보냈을 때 그것을 처리하는 View 로직을 생성했다. 어플에서 byte array 나 png 파일을 보내면, 그 정보를 파일에 저장한다. (file open(), write(), close() 함수 적용). 그 후 파일을 Microsoft Face API 에 입력으로 넣어준 후, 나온 JSON 데이터를 dictionary 형태로 받아온다. 제일 수치가 크게 나온 감정을 하나 선택하여 그 감정에 해당하는 테이블에서 entry 하나를 받아와 유튜브 링크를 어플에 보내준다.

a. 이미지를 이용한 감정분석

이미지로 감정을 분석하는 API 로는 Microsoft, Naver, Google 등에서 오픈한 예들이 여러 가지가 있었다. 그 중 우리 프로젝트 목표에 잘 맞게끔 감정을 복합적으로 보여줄 수 있는 Microsoft Face API 를 선택하였다.

위의 API 를 쓰기로 하였지만, 이것이 믿을만 한 것인지 테스트 해 볼 필요가 있었다. 그래서 테스트 이미지를 100 개쯤 넣어서 정확률이 얼마만큼 되는지, 사람이 판별한 감정과 API 가 판별한 감정이 얼마나 일치하는지를 보았다.

얼굴 인식 면에서는 어두울 때, 얼굴의 측면만 나왔을 때, 초점이 안 맞고 흐릿한 사진일 때, 가까울 때나 멀때 잘 나오는 것을 확인하였다. 대신 얼굴의 반 이상이 잘렸을 경우나, 너무 아래쪽에서 사진을 찍은 경우는 얼굴 인식이 잘 되지 않는 것을 보았다. 이런 경우를 보완하기 위해 어플을 실행한 후 모델이 나와서 “나와 눈을 맞춰줄래” 라고 코멘트를 남기는 방법을 생각하였다.

가장 중요한 감정인식 면에서는 대체적으로 잘 나오는 것을 확인하였다. 그러나, 전체적으로 테스트를 해 보았을 때 주로 나오는 감정은 중립과 행복이어서 복합적인 감정을 고려한 추천 알고리즘 방안을 마련해야 할 것이다.

b. 음성을 이용한 감정분석

음성으로 감정을 분석하는 단계에서는 학습된 모델을 사용한다. 직접 모델을 설계하여 진행하는데에 정확률에 대한 어려움이있어 github 에 공개된 Speech_Emotion_Analyzer 를 사용하였다. (<https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer>)

이 모델(SEA)을 사용하여 실제로 음성을 입력으로 넣었을때 결과가 어떻게 나오는지 확률적으로

테스트를 해 보았다. SEA 는 외국인들이 진행한 프로젝트이며, 학습에 사용된 감정섞인 데이터셋 또한 영어로 녹음된 것이다. 따라서 우리가 직접 입력으로 넣는 영어 음성 감정 테스트데이터에 대해서 결과가 어떻게 나오는지 뿐 아니라, 이것을 우리 어플에 적용하기위해 한국음성 감정테스트 데이터에 대해서는 결과가 어떻게 나오는지도 확인해야했다. 결과는 다음과 같았다.

English Recording Dataset		
angry	10.00%	angry는 대개 fearful이나 sad로 인식하는경향
happy	90.00%	
neutral	0.09%	목소리가 낮아서 sad로 인식하는 경향
fearful	60.00%	
sad	10.00%	sad는 calm이나 fearful로 인식하는 경향
Korean Recording Dataset		
angry	12.50%	angry를 fearful로 인식하는 경향
happy	0.00%	fearful로 인식하는 경향이 있음
neutral	50.00%	sad로 인식하는 경향이 있음
fearful	0.00%	angry나 sad로 인식하는 경향
sad	75.00%	한국인의 격양된 슬픔이 가끔 angry로 인식,여성슬픔 잘 인식


그리고 SEA 는 duration 값 2.5 로설정되어 있다. 테스트를 진행할 때에 offset 값을 바꿔가면서 감정섞인 음성데이터의 타이밍을 찾아내곤 했다. 하지만 너무 짧게 데이터가 주어졌을 때 ValueError 가 났다. 따라서 어플에서 음성데이터를 받을 때, 사용자의 감정표출도 중요하지만 그 길이도 중요하다고 여겨진다. 따라서 어플에서 대답을 들을때에 어떤 질문을 유도할 것이냐와 더불어 어느 정도의 길이를 확보하는 전략을 세워야 한다는 결론을 얻었다.

이러한 결과를 토대로 앞으로 감정을 최종 도출하는 방법에서 음성-감정모델의 결과를 어떻게 반영할 것인가에 대해서 논의 중이다.

c. Database

본 프로젝트의 어플리케이션이 작동하기 위해 설계된 데이터베이스 테이블은 다음과 같다.
:EMOTION_INFORMATION, USER_INFORMATION, COMMENTS_LIST, HAPPINESS, SADNESS, ANGER, FEAR, DISGUST, SURPRISE, SUBCLASS_SAD, ANALYSIS_RESULT

EMOTION_INFORMATION 테이블에는 Microsoft Face API 의 emotion 분석 결과 값인 8 개의 감정을 담아둔다. 8 개의 감정은 happiness, surprise, sadness, fear, anger, contempt, disgust, neutral 이다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

USER_INFORMATION 테이블에는 회원가입 시 입력 받은 사용자의 닉네임과 패스워드가 저장된다. 더불어 어플리케이션의 마지막 단계인 사용자 만족도 조사에 대한 결과값도 저장된다. emotion 테이블에는 HAPPINESS, SADNESS, ANGER, FEAR, DISGUST, SURPRISE 와 같이 6 개의 테이블이 있는데, 이는 미국의 심리학자들이 정의한 인간의 6 가지 감정을 기준으로 하였다. SADNESS TABLE 에 대해서는 SUBCLASS_SAD TABLE 을 이용하여 슬픈감정에 관련된 노래들을 더 세부적으로 나눌 예정이다. 또한 본 어플리케이션에서는 선정된 노래와 함께 분석 결과 값과 관련된 간단한 코멘트가 같이 나가는데, 이 코멘트들은 COMMENTS_LIST TABLE 에 저장한다. ANALYSIS_RESULT 에는 사용자의 닉네임과 함께 분석 결과 값에 따라 선정된 노래와 코멘트들을 담아두는데, 이는 사용자들이 자신이 이전에 추천 받은 노래들을 다시 확인하고 싶을 때나, 만족도 조사를 통한 노래 분류, 추천 알고리즘의 피드백에 사용될 것이다.

database 는 MySQL 과 연동하여 Django 에서 작업하였고, 현재 설계한 TABLE 은 다 정의 되어있는 상태이며, 각 TABLE 에 알맞는 데이터를 업데이트하고 있다.

d. 노래 추천 알고리즘

Microsoft Face API 의 얼굴 이미지를 이용한 감정 분석 결과와 Speech Emotion Analyzer 의 음성(억양)을 이용한 감정 분석 결과를 통해 최종 결과를 도출하면, 그 결과를 추천 알고리즘의 입력으로 넣는다. 본 시스템의 추천 알고리즘에 따라 노래를 1 순위에서 3 순위까지 추천한다. 자세한 내용은 다음과 같다.

얼굴 이미지를 이용한 감정 분석 결과와 음성(억양)을 이용한 감정 분석 결과를 통해 도출되는 최종 결과는 다음과 같은 json 형태이다. 이는 추천 알고리즘의 입력이 된다.



```
{
  "userId" : "f70c4331-f391",
  "Attributes" : {
    "age" : 13.0,
    "emotion" : {
      "anger" : 0.0,
      "contempt" : 0.001,
      "disgust" : 0.0,
      "fear" : 0.0,
      "happiness" : 0.206,
      "neutral" : 0.793,
      "sadness" : 0.0,
      "surprise" : 0.0
    }
  }
}
```

(2) 추천 알고리즘 설계(정책)과 추천 알고리즘 순서도


본격적인 알고리즘 처리에 앞서, 입력되는 최종 감정 결과를 전처리 한다. 각 감정의 비율이 0.01 미만이면 0으로 스무딩한다. 정확한 전처리 기준은 테스트를 통해 확정하기로 하였다.

입력은 8개의 감정(화, 경멸, 역겨움, 공포, 행복, 중립, 슬픔, 놀람 / anger, contempt, disgust, fear, happy, neutral, sad, surprise)과 그에 따른 비율이다. Face API 테스트를 진행했을 때, '화'나 '경멸'이나 '역겨움' 중에서 두 가지 이상이 함께 나올 확률은 70%였다. 그래서 '화'와 '경멸'과 '역겨움'을 비슷한 성격의 감정으로 취급해 한 항목으로 만들었다. 따라서 6개의 감정 상황에 따라 알고리즘을 수행하면 된다.

본 시스템의 추천 알고리즘 정책은 다음과 같다.

ㄱ. 전처리를 마친 후에는 1순위 감정을 중심으로 알고리즘을 처리한다. 사용자가 특정 감정일 때 선호하는 노래는 각기 다르기 때문에, 다양성을 고려해 여러 분위기의 노래를 1~3순위까지 고루 추천하기로 하였다. 예를들어, '슬픔' 감정일 때, 추천할 노래 분위기의 종류는 3가지로 만든다. 이 3가지 종류는 각각 잔잔한 노래인 '슬픔(1)음악', 너무 슬퍼서 울고싶은 노래인 '슬픔(2)음악', 기분전환을 할 수 있는 밝은 노래인 '슬픔(3)음악'이다.

ㄴ. 이때 '슬픔(1)음악'인 잔잔한 노래는 다른 감정 결과일 때도 사용될 수 있으므로 추천 상황에

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19


따라서 활용한다. 예를 들어, '공포'감정에서 잔잔한 노래를 추천해줄 때 Database 의 슬픔테이블에 접근해 노래를 가져올 수 있다. (Database 의 슬픔 테이블에는 앞에서 기술한 3 종류의 분위기에 따라 노래들이 소분류되어 있다.)

ㄷ. 분석된 감정이 복합적으로 나올 수 있으므로 경우에 따라 2 순위 감정과 3 순위 감정까지 고려할 수 있다. 만약 감정이 3 순위까지 나오지 않아 노래추천에 문제가 생길 시, 1 위 혹은 2 위로 추천된 노래의 root 로 가 랜덤 추천을 한다.

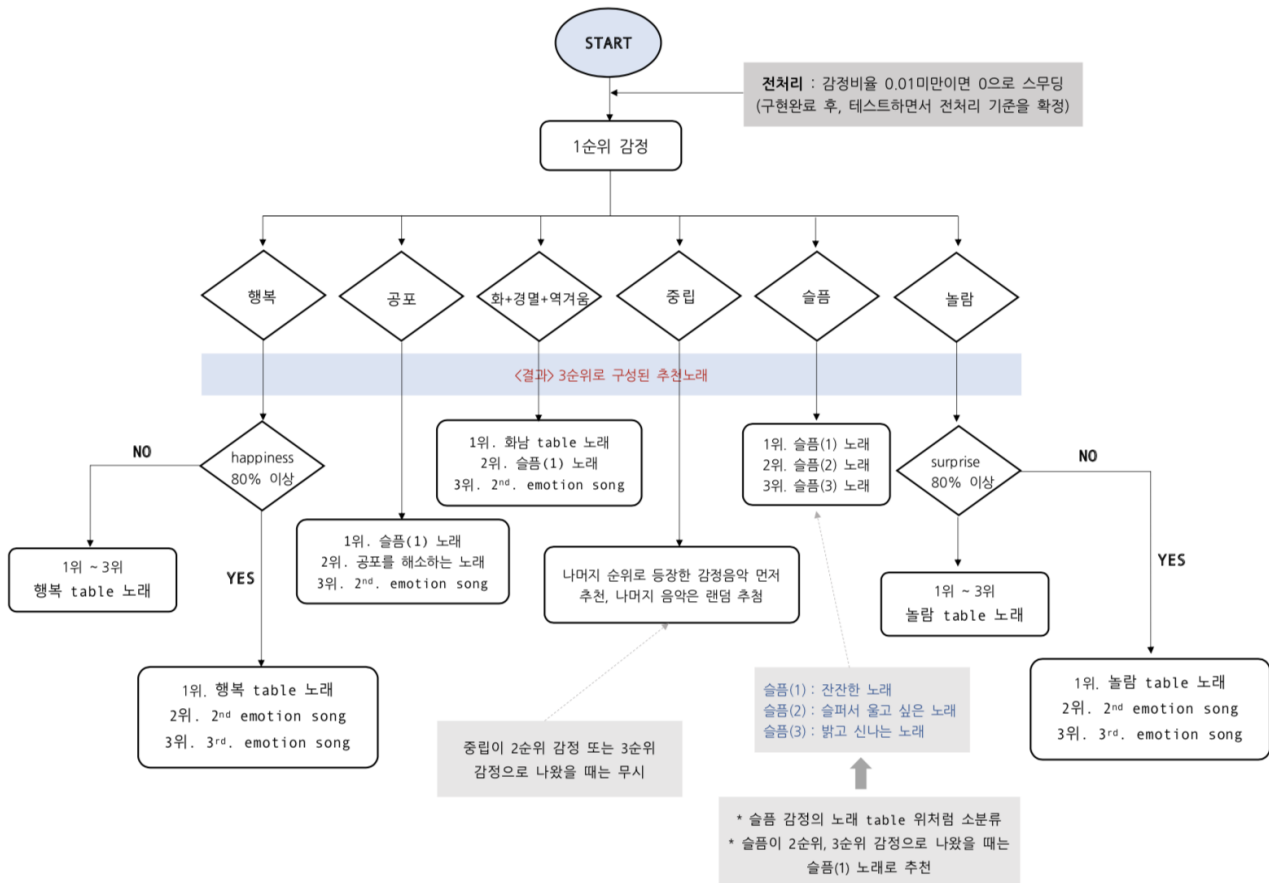
ㄹ. 다양한 추천노래 중 사용자의 선택에 도움을 주기 위해, 추천된 노래 옆에 노래의 분위기를 알 수 있는 해시태그(#)를 넣기로 한다.


위의 사항을 고려해 추천 알고리즘을 설계한다면, 감정마다의 추천 노래는 다음과 같다. 아래에 첨부된 알고리즘 구성도에서 부연 설명이 필요한 것만 기술하였다.

- '화+경멸+억거움' 감정일 때 추천 노래들은, 1 위 : Database 의 화남 테이블의 웃긴 분위기의 노래(이하 '화남음악'), 2 위 : 화남을 진정시킬 수 있는 Database 의 슬픔 테이블의 잔잔한 노래(이하 '슬픔(1)음악'), 3 위 : 2 순위 감정음악
- '공포' 감정일 때 추천 노래들은, 1 위 : 슬픔(1)음악, 2 위 : 공포를 해소하는 노래, 3 위 : 2 순위 감정음악
- '중립' 감정일 때 추천 노래들은, 나머지 순위로 등장한 감정음악을 먼저 추천해주고 나머지 음악은 랜덤으로 추천한다. 추가적으로, '중립' 감정이 최종감정(입력)에서 2 순위 감정 또는 3 순위 감정으로 나왔을 때는 무시한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

다음은 추천 알고리즘 정책에 따른 알고리즘 구성도이다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

(3) 예상 결과물

아래는 추천 알고리즘을 통해 나온 예상 결과물을 표현한 도식이다.

1순위



[MV] Onestar(임한별) _ A tearful farewell(사랑 이따 거)

#슬픔 #이별 #상처 #눈물

2순위



김예림 Lim Kim - 잘 알지도 못하면서 Without Knowing It All (Official Audio)

#슬픔 #위로 #공감 #잔잔

3순위



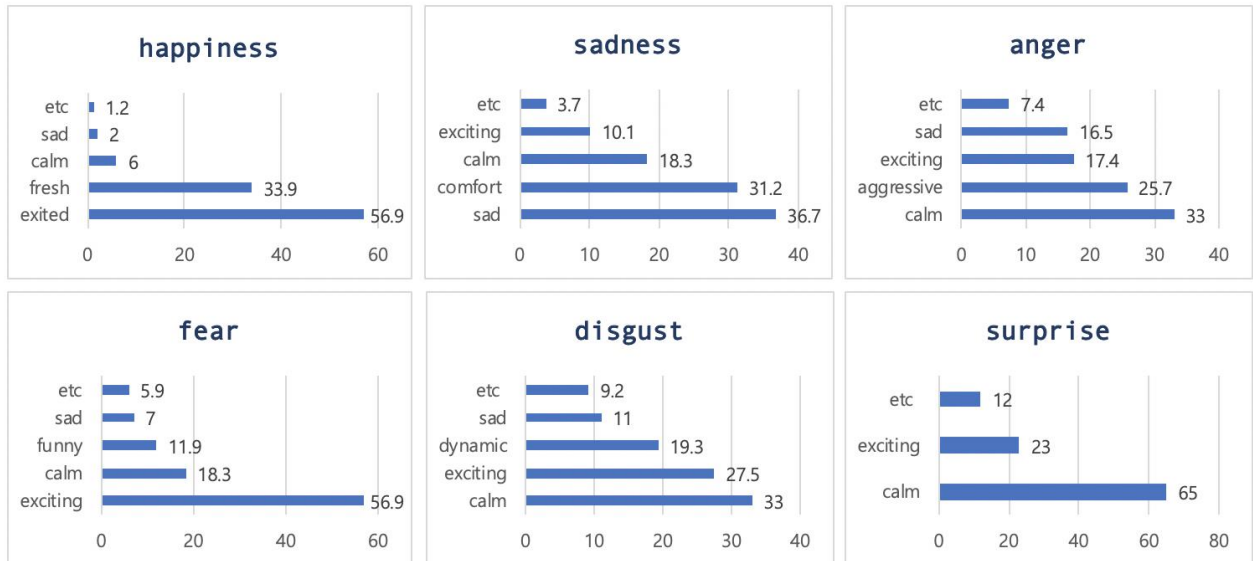
2NE1 - GO AWAY M/V

#슬픔 #기분전환 #댄스곡

(image reference : www.youtube.com, tag reference : www.melon.com)

e. 노래 분류


노래를 감정 테이블에 맞게 분류하기 위한 정책을 정하기 위해 여러모집단을 대상으로 '각 감정의 상황에서 어떤 노래를 듣는지'에 대한 설문조사를 진행하였다. 설문조사에 대한 결과는 다음과 같다.



위의 설문조사의 결과를 토대로 6 개의 감정 테이블에 들어갈 노래의 종류 및 분위기를 다음과 같이 정하였다.

- (1) Happiness : 신나는 노래, 발랄하고 상큼한 노래
- (2) Sadness : 슬픈 노래, 위로가 되는 노래
- (3) Anger : 진정이 되는 노래, 공격적인 노래(화를 표출할 수 있는 노래)
- (4) Fear : 상황을 반전시켜주는 밝은 노래, 진정이 되는 노래
- (5) Disgust : 잔잔한 노래, 신나는 노래
- (6) Surprise : 진정이 되는 노래

노래 리스트는 음원 스트리밍 사이트 멜론(melon), 벅스(bugs), 지니(genie)의 tag 를 활용하여 분위기에 맞는 노래 데이터를 가져 올 것이다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

2.2.2 Front-End

A. 3D 캐릭터

3D 캐릭터의 modeling 과 rigging, controller, skin bind, uv texturing 과 같은 기본 사항은 작업한 상태다. 추가적으로 어플 시작시 필요한 깨어나는 애니메이션에 대해 작업하고 있다.

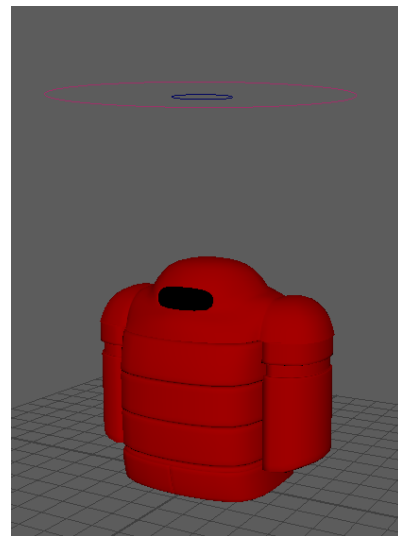
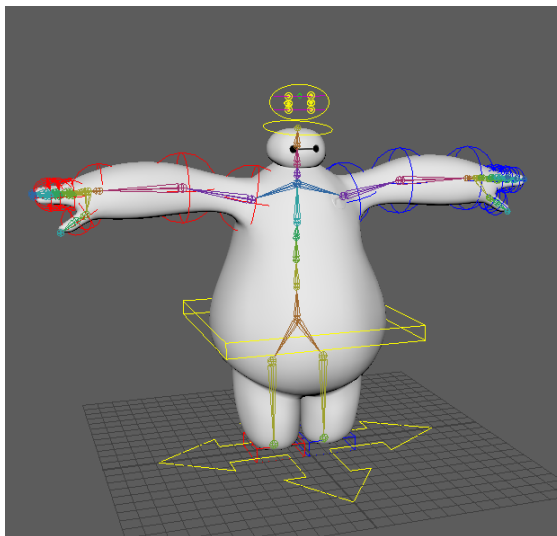
먼저 깨어나는 애니메이션을 케이스 안에 들어가서 잠든 상태에서 케이스가 열리면서 모델이 깨어나는 형식으로 제작할 것이기 때문에 필요한 모델링을 작업했다.


해당 texturing 은 나중에 하고 연결하면 되기 때문에 먼저 케이스가 열리는 애니메이션을 작업했다. 추가적으로 모델과 함께 연관지어서 추가적인 애니메이션 작업을 해주면 된다. 해당 작업은 blend shape 와 driven key 들을 활용해서 in-between 으로 작업할 예정이다.

추가적으로 3D 모델은 그저 모델일 뿐 collider 가 적용이 안되어있는데 원래는 마야에서 제공해주는 muscle 이나 delta mush 를 사용하려 했으나 해당 기능은 면마다의 연산이 들어가서 충돌체 기능을 해주는 것이기 때문에 어플에 사용하기엔 연산량이 많아져 사용하지 않기로 했다. 대신 필요한 충돌에 대하여 blend shape 을 각각 만들어서 처리해 주기로 했다.

눈 깜빡이는 애니메이션도 작업을 했다. facial rig 으로 blend shape 을 이용해서 morph target 으로 만들었다. 그래서 어플 상에서 idle 로 눈을 깜빡이는 애니메이션을 적용할 예정이다.

skin weight 가 controller 를 조작하면서 조금 어긋나는 부분들이 있어서 전면 수정을 했다. 하지만 또 애니메이션을 제작하다가 어긋나는 부분이 있으면 계속 수정을 해줘야한다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

B. application

splash image 를 처음에 구현을 했다가 splash image 에 추가적으로 더 필요한 기능을 사용하지 않을 예정이어서 unity 에 존재하는 간단한 splash 기능을 사용해서 로고와 배경이 뜨는 방향으로 구현했다. 이후에 로고와 splash image 작업을 해서 이미지를 수정할 예정이다.


카메라 permission 을 받아서 카메라 사용 허가를 받고 WebCam 을 이용해서 개발용으로 카메라가 화면 상에 뜨도록 구현했다. 핸드폰을 돌렸을 때 카메라도 rotation 되어서 표시될 수 있도록 구현했다. 캡처한 사진을 PNG 형식으로 인코딩해서 사진이 잘 찍혔는지 확인이 가능하게 구현해놓았다.

ImultipartFormSection 을 사용해서 Post 를 날려서 서버에서 정보를 잘 받는지 확인하였다.


RawImage 에 Video player 를 사용해서 영상과 소리가 함께 출력되는 것을 확인하였다.

audio source 를 이용해서 음성 정보를 저장할 수 있도록 했다. 마이크를 인식한 뒤 record 버튼을 눌러서 음성을 받고 save 를 눌러서 .wav 파일로 저장이 되는 것을 확인하였다.

대분류	소분류	기능	진행상황	진도 비교
Back-End (server)	이미지를 이용한 감정분석 (Microsoft Face API)	어플에서 입력받은 이미지를 분석하여 감정에 대한 결과를 리턴	이미지를 입력, api 에서 받아온 결과를 HttpResponse()로 출력, 이미지 분석 결과 확인 및 테스트	완료
	음성을 이용한 감정분석 (Speech-Emotion- Analyzer)	학습 모델을 활용하여 사용자 대답을 통한 감정 분석	로컬에서 모델을 사용하여 데이터 입력에 대한 결과 확인	(계획 수정 후) 완료
	Database	사용자 정보, 코멘트, 추천할 노래를 저장하는 공간	DB 구축	완료

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

	노래 추천 알고리즘	분석된 감정 결과에 추천되는 노래를 매칭하는 알고리즘	추천 알고리즘 설계	80% 완료
Front-End	모델링&애니메이 션	어플리케이션에 디스플레이 해줄 모델	모델링, 간단한 uv texturing, 리깅, skin bind, facial rig 로 얼굴 표정 리깅 완료 케이스 모델링, 애니메이션 완료	80% 완료
	application	서버에 데이터를 보내고, 받은 정보를 디스플레이	서버에 Post 보내기 카메라로 사진 촬영 splash 화면 구현 동영상 재생 음성 저장	80% 완료

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

3 수정된 연구내용 및 추진 방향

3.1 수정사항

3.1.1 Back-End

a. 음성을 이용한 감정분석


추가- github 에 올라온 Speech_Emotion_Analyzer 를 사용하기로 한다. readme 에 명시된 정확률은 약 70%정도이므로 우려했던 모델의 정확률 문제가 어느정도 해결되었다 생각하여 어플에 적용하기로 한다.

삭제- 원래는 음성-감정모델을 직접 학습시켜 감정분석 결과를 얻기로 했다. 31 일 피드백을 받은 결과 데이터셋 약 2000 개를 가지고 학습을 하였을 때 얻게되는 정확률이 부족할 것으로 예상되었다. 예상되는 정확률이 약 3-40%정도 되어 최종 감정 결과를 도출할 때에 이미지 인식 결과를 무디게할 수 있다고 생각하여 직접 학습시킨 모델을 빼고, API 를 통한 이미지 인식과 Speech_Emotion_Analyzer 의 활용성을 더욱 높일 수 있는 방향으로 간다.

보완- 사용한 모델은 감정이 섞인 영어로 recording 된 데이터를 사용하여 학습되었다. 따라서 일단 새로운 영어 데이터를 넣었을 때 나오는 정확도를 측정하고, 한국어 데이터를 넣었을 때 어떻게 인식되는지를 확인하여 그 차이를 어떻게 보완해 나갈 것인지 전략을 세워야 한다. 인식률을 위해 영어로 대답을 들을 경우, 어플에서 사용자에게 대답을 어떻게 유도 할 것이냐에 대한 캐릭터의 질문을 고민한다. 이때에 모델의 특성상 사용자의 대답이 3 초이상 지속될 수 있도록 해야 할 것이고 그렇게 되지 못했을 때 학습 모델에서 발생하는 오류를 서버에서 어떻게 처리하여 감정 결과를 도출 할 것인지 결정한다.

b. database

계획서 상의 목표는 MUSIC TABLE 에 모든 노래를 넣어놓고 분석된 감정 값과 맞는 노래들을 관계 테이블을 통해 선정하는 것이다. 하지만 노래의 리스트들을 감정 별로 나누어 담아놓는 것이 더 효율적이라고 판단하여 RELATION TABLE 을 삭제한 뒤 HAPPINESS TABLE, SADNESS TABLE, ANGER TABLE, FEAR TABLE, DISGUST TABLE, SURPRISE TABLE 을 생성하여 각 테이블에 맞는 노래를 넣어두기로 하였다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

c. 노래 추천 알고리즘

계획서에 따르면, 추천 알고리즘은 단순히 노래 리스트 중 하나를 랜덤함수로 가져오는 것이었다.

추천 알고리즘에서 수정된 연구내용의 핵심은 다음과 같다.

- (1) 사용자의 복합적인 감정을 고려한 1 위 ~ 3 위의 노래 추천이 이루어진다.
- (2) 해시태그 기능을 이용해 추천된 노래들의 분위기를 알 수 있어, 사용자의 노래선택에 도움을 준다.
- (3) 이용할 감정 결과는, 'anger', 'contempt+disgust', 'fear', 'happy', 'sad', 'surprise' 로 새롭게 분류한다.
(2.2 의 현재 수행 내용에서 변경된 내용이다.)

변경 내용과 이유에 대해 자세히 기술하면 아래와 같다.

본 프로젝트에서 추천 알고리즘을 우리 시스템의 정책에 따라 설계하려고 하였다. 기존의 노래 리스트 중 하나를 랜덤으로 가져오는 것에서, 사용자의 다양한 감정을 고려한 노래 추천이 이루어지게 하고자 하였다. 복합적인 감정 결과를 고려하기 위해 3 순위 감정 결과까지 나왔다는 전제하에, 모든 경우의 수를 수형도로 표현해보았다. 데이터가 많은 관계로 원본([추천알고리즘_설계도_ver2](#))을 참고할 수 있다.


프로젝트 진행에 관하여 교수님 몇 분과 면담한 결과, 우리 시스템의 정책을 가지고 가도 좋을 것 같다는 의견과 타당성을 부여해야한다는 의견의 극명히 다른 조언을 들었다. 따라서 기존의 정책대로 가되, 설문 조사 결과를 통해 정책을 보강하기로 하였다.

현재까지의 수행 내용에서는 감정을 'anger+contempt+disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise' 으로 분류하였다. 그런데 우리의 행동을 지시하고 신속하게 움직일 수 있게 하는 등 중요한 역할을 하는 감정들 중, 가장 중요한 감정인 놀라움, 혐오, 공포, 행복, 슬픔, 그리고 화를 기본 감정이라고 부른다. ([reference](#))

이 지침을 따라서 disgust 를 분리하고 neutral 을 제거한 'anger', 'contempt+disgust', 'fear', 'happy', 'sad', 'surprise' 로 감정을 새로이 분류한다. 이 내용은 중간보고서 상에서 '노래 분류'와 'Database'에는 반영하였지만 '추천 알고리즘'에는 아직 반영을 안한 상태이다. 이 주제는 항목 '4. 향후 계획'에 추가적으로 기술한다.

d. 노래 분류

계획서 상의 노래 분류 방법으로는 가사의 단어를 분석하여 각각의 감정에 해당하는 노래로 분류하는 TF-IDF 알고리즘을 사용하는 것이었는데, 기간 내의 실현 가능성의 문제로 이를 일단 보류한 상태이다. 노래 분류를 하는 새로운 방법으로는 음원 스트리밍 사이트 멜론(melon), 벅스(bugs), 지니(genie)의

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

tag 를 사용하여 노래를 가져오는 것으로 생각한다.

3.1.2 Front-End

A. 3D 캐릭터

Maya 에서 arnold 나 vray 를 이용해서 subsurface scattering 로 export 를 하려했으나 .fbx 로 export 하기 위해서는 maya 에서 지원해주는 기본 material 인 lambert 나 blinn, phong 등으로만 해야 링크가 끊기지 않고 export 되기 때문에 maya 상에서 하지않고 기본 텍스처만 뽑아온 다음 유니티 상에서 렌더를 돌리는 방식으로 한다.

B. application

삭제- 얼굴을 인식함에 있어서 사용자에게 카메라로 사진을 찍을 것인지 갤러리에 존재하는 이미지를 선택할 것인지 선택할 수 있게 하려 했으나 우리는 당장의 감정에 대한 분석과 그에 따른 노래 추천이 목적이었기에 갤러리 기능은 좀 더 나중의 부가적인 기능이라 제하기로 한다.

보완- 카메라로 직접 사진을 찍을 경우 내가 찍히는 얼굴에 대해서 인식이 되기 때문에 자연스러운 얼굴 데이터를 수집할 수 있도록 어플 화면 상에는 나타나지 않게 backend 로 이미지 수집이 이루어질 예정이다.


4 향후 추진계획

4.1 향후 계획의 세부 내용

4.1.1 Back-End

A. server

AWS Elastic Beanstalk 를 사용하여 애플리케이션을 배포한다. Elastic Beanstalk 는 애플리케이션을 배포할 때, 선택된 지원 가능 플랫폼 버전을 구축하고 Amazon EC2 인스턴스 등의 AWS 리소스를 하나 이상 프로비저닝하여 애플리케이션을 실행하여 쉽게 배포 가능할 것 같다. 현재 WSGIPath 문제 때문에

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

잠시 멈춰있는 상태지만, 이 문제를 해결한 후 배포완료할 예정이다. 그렇게 하면 현재 데이터베이스 테이블 엔트리를 로컬에만 가지고 있는 것이 아니라 AWS 리소스를 사용하여 DB 에 쌓아둘 수 있어 편할 것 같다.

어플에서 이미지와 음성 정보를 보내주면, 그 정보들을 이용해 감정분석을 한 후 최종결과를 도출하는 과정을 거친다. Microsoft Face API 를 사용한 결과는 최종 3 순위까지만 고려하기로 결정하였고, 음성을 이용한 감정분석 결과는 예를 들어 'female_angry' 이런 형식으로 1 개만 나오는데, 이러한 결과들을 어떻게 적절하게 처리할 것인지 정한다.


그리고 현재까지는 CSRF(Cross Site Request Forgery protection) 이슈를 없애기 위해 프로젝트 settings.py 에 csrf.middleware 를 주석처리해놓고 request 를 처리했는데, 그렇게 하면 보안에 문제가 생기는 것이기 때문에 이 이슈를 해결한다. 단순 Django 에서는 템플릿에서 폼을 작성해 csrf 토큰을 전달하면 되지만, RESTful API 구현을 위해 template 을 쓰지 않기로 했으므로 Django REST Framework 의 Serializer 를 사용할 것이다. 또한 성능을 최적화 하기 위해 Serializer 가 중첩된 필드를 쿼리할 때 @staticmethod 로 마킹된 setup_eager_loading 메소드를 시리얼라이저에 추가한다.

Django 의 settings.py 에 명시되어 있는 SECRET_KEY 는 주로 쿠키데이터 해시, 암호화 등 임시적인 일에 사용되고, 변경 시 로그인 세션 등의 데이터가 사라질 수 있다. SECRET_KEY 는 공개되면 안 되는 정보이기 때문에 settings.py 파일에서 분리해야 한다. secrets.json 파일을 작성에 JSON 파일에 키 정보를 입력하고, settings.py 파일에서는 참고만 하도록 설정한다.

Django 에서는 View 라는 것에서 로직이 처리가 된다. 즉, 모델에서 사용자에게 건네줄 데이터를 추출한 뒤 View 에서 최종적으로 응답을 보내주는 형식이다. Django REST Framework 에서는 이러한 View 들의 집합인 ViewSet 을 사용한다. ViewSet 을 router 에 등록하면 자동으로 URL 패턴을 생성해줘서 매우 편리한 점을 확인하였다. 현재는 view.py 파일에 함수형으로만 구현했는데, 향후에는 class 형류, ViewSet 을 사용하는 방법 등을 더욱 다양하게 적용해보도록 한다.

a. 이미지를 이용한 감정분석

위에서 언급한 대로 API 결과가 중립이나 행복 위주로 나오는 문제점이 있었다. 사람 눈으로 판단한 것보다 결과가 더 잘 나오지 않을 경우에 어떻게 할 것인지 정해야 한다. 다른 방법을 찾아본 결과, 현재까지는 github 에 자바스크립트 프로젝트로 오픈되어 있는 face-api 를 찾게 되었다. (<https://github.com/justadudewhohacks/face-api.js>) Microsoft Face API 정확률을 확인하기 위해

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

넣었던 이미지를 똑같이 넣어본 결과, 감정분석 성능이 조금은 더 향상될 수 있을 것 같다. 나중에 성능을 조금 더 높이고 싶다면 이 face-api 를 사용하는 방법을 고려해 본다.

b. 음성을 이용한 감정분석

어플리케이션에서 음성데이터를 입력받으면 서버를 통해 결과를 도출한다. 따라서 음성-감정 모델이 서버에 구현되어야 한다. Django project 의 views.py 에 음성-감정 모델의 코드를 삽입한다. 이때에 중요한 점은 결과값을 어떻게 리턴하는 가이다.

```
label = [
    "female_angry", "female_calm", "female_fearful", "female_happy", "female_sad", "male_angry", "male_calm",
    "male_fearful", "male_happy", "male_sad"
]
```


지금 로컬에서는 string 값으로 해당하는 index 의 감정을 리턴한다. 성별에 대한 정보는 지을 것이며 이미지 분석을 통한 감정 결과와 적절히 양상불이 되기 위해서 결과값의 형태를 통일 한다.

입력으로 들어오는 음성 데이터의 이름을 'input'으로 고정하여 어플에서 받아올지 함수의 파라미터로 들지 또한 정한다. 이때에 어플에서 전송된 음성데이터의 형태를 wav 로 하여 사용된 모델의 데이터 입력 형태를 유지한다.

c. database

노래 분류를 통해 분류가 된 노래들을 알맞은 테이블에 업데이트한다. HAPPINESS TABLE, SADNESS TABLE, ANGER TABLE, FEAR TABLE, DISGUST TABLE, SURPRISE TABLE 의 테이블 각각에 200 개의 노래를 업데이트하는 것을 목표로 한다. 연령별 업데이트 될 노래 비율은 20 대 타겟의 노래가 가장 많을 것이고, 30-10-40 대 순으로 비율을 줄인다.

본 어플리케이션에서는 분석된 감정에 맞는 노래를 간단한 코멘트와 함께 보여준다. 따라서 COMMENTS_LIST TABLE 에 선정된 노래들과 같이 나갈 코멘트들을 업데이트할 것이다. 코멘트는 100 개를 업데이트하는 것을 목표로 한다. 또한 어플리케이션의 구성이나 정책이 바뀔 경우에는 그에 맞는 database 를 즉각적으로 수정한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

d. 최종 감정 결과 도출


이미지 분석을 통한 감정분석 결과와 음성 학습모델에서의 감정분석 결과를 적절한 weight 값을 바꾸어 가며 최종 감정으로 도출해 낸다. 이때 weight 값은 음성-감정모델과 서버가 연동이 될 때 바꾸어가며 테스트 한다.

e. 노래 추천 알고리즘

- (1) 3.1 에서 기술한 대로, 논문에 기재된 감정의 분류인 6 개 ('anger', 'contempt+disgust', 'fear', 'happy', 'sad', 'surprise') 로 추천 알고리즘 설계를 수정한다. 설문한 결과를 바탕으로 추천 알고리즘 설계를 보충한다. 이를 바탕으로 추천 알고리즘을 구현한다.
- (2) Database 에 저장된 연령 정보를 가져와 추천 알고리즘에 적용한다. 각 연령 별로 노래 추천시 향수를 불러일으킬 수 있다는 점에서 어플의 감성적인 효과를 높인다. 이는 어플의 실용성과도 연관이 될 수 있다. (연령 정보가 필요한 감정은 'happy', 'sad', 'fear'이다.)
- (3) 최종 감정 결과가 특별한 양상을 보인다면 추천 알고리즘에서 따로 빼서 처리할 필요가 있다. 예를 들어, 사용자가 웃는 얼굴을 하고 슬픈 목소리로 대답을 하는 등의 모순된 상황에서의 노래 추천이 문제가 될 수 있다. 이를 역으로 이용해, 이런 경우에는 '거짓말'에 관련된 노래를 추천하는 등 사용자에게 재미있는 요소로 작용될 수 있도록 한다.
- (4) 사용자의 만족도 조사를 통해 설문 조사와 같은 효과를 얻을 수 있다. 만족도 조사를 바탕으로 추천 알고리즘을 보충한다.

f. 노래 분류

감정 TABLE 각각에 200 개의 노래를 넣는 것이 목표이기 때문에, 약 1200 개의 노래를 수집해 분류할 것이다. 또한 TF-IDF 알고리즘뿐만 아니라 노래를 분류하는 데 있어 활용할 수 있는 기계적인 방법을 계속 찾아볼 예정이다. 만약 적절한 방법을 찾을 경우 알고리즘을 기반하여 노래를 분류하는 방법과 음원 스트리밍 사이트의 태그를 이용하는 방법을 조합하여 노래를 분류하는 것으로 한다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

4.1.2 Front-End

A. 3D 캐릭터

케이스의 애니메이션까지 제작을 해놓은 상태이다. 이제 케이스와 모델을 결합해서 깨어나는 최종 애니메이션을 작업한다.

그리고 uv texturing 작업도 한다. 케이스 애니메이션이 끝나고 나면 리깅되어있는 모델로 target 을 바꿀 수 있도록 한다.

maya 기본 material 로 export 를 한 후 unity 상에서 subsurface scattering 이 되는지 확인해보고 불가능하다면 기본 material 로 작업한다.

추가적으로 필요한 애니메이션이 있다면 작업한다. 팔이 움직여서 충돌했을 경우 blend shape 를 이용해서 충돌했을 때의 target 을 바꿔줘서 자연스럽게 보일 수 있도록 한다.

B. application


splash image 를 바꿔주는 등 전체적인 UI 디자인을 한다.

현재 기능이 구현 가능한지 테스트를 거의 다 한 상태이니 테스트했던 내용들을 토대로 전체적인 어플 흐름을 구현한다.

- 카메라가 어플 상에 표시되지 않도록 하고 카메라를 통해 가져온 이미지 정보를 서버로 보낸다.
- 저장된 .wav 음성 파일을 서버로 보낸다
- 서버를 통해 받은 정보를 팝업으로 띄워준다

기능 테스트 중 해결 못한 문제가 하나 있다. 동영상 출력은 가능하지만 유튜브 url 을 replace 해줘야하는데 아직 가능한 방법을 찾지 못했다. 그래서 youtube extractor 나 url replace 방법들을 더 찾아보고 url 로 영상을 띄울 수 있는지 추가적 테스트 한다.

만약 url 로 해결되지 않는다면 유니티 에셋 스토어에 존재하는 youtube player 에셋을 가져와서 영상을 띄울 수 있게 하거나 thumbnail 이미지만 불러와서 이미지를 클릭했을 때 하이퍼링크로 유튜브 앱으로 연결될 수 있게 구현할 예정이다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	My_Mood_Music #마무무	
	팀 명	포춘텔러	
	Confidential Restricted	Version 1.6	2019-APR-19

5 고충 및 건의사항

1. 캡스톤 디자인을 수강하는 학생들을 위한 전용 강의실 필요
 - 캡스톤 디자인을 수강하는 학생들은 학교에 오래 남아있거나, 팀별 활동을 할 경우가 많은데 이를 수행할 수 있는 강의실이 자율주행스튜디오밖에 없어 공간이 부족하다. 또한 팀원 각각의 산출물을 놓을 공간이 마땅하지 않다. 따라서 학생들에게 캡스톤 디자인 전용 강의실을 제공하여 이러한 문제를 해결해줬으면 하는 바람이다.
2. 지원항목 규제에 대한 완화
 - 범용적인 소프트웨어가 프로젝트에서 필요한 상황임에도 불구하고 지원항목에서 제외되었다. 학생들이 더 효율적이고 유연하게 프로젝트를 진행할 수 있도록 이 규제 사항을 완화할 필요가 있다고 판단된다.