



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Flex Ads
팀 명	둥글둥글
문서 제목	결과보고서

Version	2.2
Date	2019-05-28

팀원	민지수 (조장)
	유지원
	윤지영
	이성재
	황수진

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

CONFIDENTIALITY/SECURITY WARNING

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

문서 정보 / 수정 내역

Filename	최종보고서-Flex-Ads.pdf
원안작성자	민지수, 유지원, 윤지영, 황수진
수정작성자	민지수, 유지원, 윤지영, 이성재, 황수진

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-05-17	황수진	1.0	최초 작성	전반적인 내용 초안 작성
2019-05-19	민지수	1.1	최초 작성	부록 최초 작성, 1 개요, 2 목표 및 개발내용 수정
2019-05-20	유지원	1.2	내용 추가	2.2.3 - 2.2.5 내용 수정 및 추가
2019-05-23	윤지영	1.3	내용 추가	2.2.1 내용 추가
2019-05-23	황수진	1.4	내용 추가	5 내용 추가
2019-05-24	민지수	1.5	내용 추가 및 수정	2.2.1, 3, 2.2 내용 추가
2019-05-25	유지원	1.6	내용 추가 및 수정	2.2.4 수정 및 추가
2019-05-25	민지수	1.7	내용 추가 및 수정	2.2.1, 5.2, 5.3 내용 수정 및 사진 첨부
2019-05-26	유지원	1.8	내용 추가 및 수정	2.2.1 그림 수정
2019-05-26	황수진	1.9	내용 추가 및 수정	2.2.1 내용 수정
2019-05-26	이성재	2.0	내용 추가	5.3 내용 추가
2019-05-27	윤지영	2.1	내용 수정	문맥 및 맞춤법 검사
2019-05-28	황수진	2.2	완성본	보고서 완성 및 pdf 변환

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	5
1.2.2	오프라인 시장의 광고 현황	7
1.2.4	온라인 시장의 광고 현황	8
2	개발 내용 및 결과물	9
2.1	목표	9
2.2	연구/개발 내용 및 결과물	10
2.2.1	연구/개발 내용	10
2.2.1.1	고객의 얼굴 Detect 및 얼굴 이미지 생성	10
2.2.1.2	얼굴 데이터 분석을 통한 회원 인식	11
2.2.1.3	등록되지 않은 회원 인식	12
2.2.1.4	구매 내역 기반의 광고 추천	12
2.2.1.5	Serverless 환경에서의 광고 송출	18
2.2.2	시스템 기능 요구사항	20
2.2.3	시스템 비기능(품질) 요구사항	21
2.2.4	시스템 구조 및 설계도	23
2.2.5	활용/개발된 기술	24
2.2.6	현실적 제한 요소 및 그 해결 방안	25
2.2.6.1	하드웨어 제한 요소	25
2.2.6.2	소프트웨어 제한 요소	26
2.2.7	결과물 목록	28
2.3	기대효과 및 활용방안	29
3	자기평가	30
4	참고 문헌	31
5	부록	32
5.1	사용자 매뉴얼	32
5.2	운영자 매뉴얼	32
5.3	테스트 케이스	39

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

1 개요

1.1 프로젝트 개요

매일 많은 정보를 새롭게 접하는 현대인들에게, 오프라인 마켓의 종이 전단지는 관심을 끌기 어려우며 적절한 광고 효과를 얻기 힘든 상황이다. 몇몇 매장에서는 기존의 정적인 광고 방식에서 벗어나고자 디스플레이를 이용하지만, 무작위 다수에게 전달한다는 점에서는 여전히 효과가 작다.

Flex Ads 는 이러한 문제점을 해결하고, 기업과 고객 모두에게 이익을 주기 위해서 고안된 프로젝트이다. 온라인상에서 성공적으로 효과를 거둔 개인 맞춤형 광고를 오프라인 마켓으로 가져와 더 나은 광고 시스템으로 접목하는 것이 이 프로젝트의 목표이다. 시스템의 구현을 위한 기능은 크게 '회원의 구매 내역을 기반으로 한 광고 추천', '얼굴 detect 를 통한 회원 인식', 그리고 'Serverless 기반의 광고 송출'의 세 부분으로 구분할 수 있다.

먼저 광고 추천 시스템은 광고 송출 시점과 관계없이 클라우드 서버상에서 주기적으로 작동한다. 추천 알고리즘은 AWS RDS(이하 RDS)에 저장된 회원 구매 기록을 기반으로 작동되며, 축적된 구매 기록 데이터를 AWS EC2(이하 EC2)에서 읽어 들인다. 추천 알고리즘의 결과는 각 고객에게 어떤 광고를 송출할지에 대한 정보이며, 이는 AWS DynamoDB(이하 DynamoDB) 에 저장된다.

한편, 얼굴인식 기능을 구현하기 위해 모듈형 AI 슈퍼컴퓨터인 'NVIDIA Jetson TX1 / TX2'를 이용하여 실시간으로 얼굴을 촬영한다. 딥러닝을 이용해 얼굴 부분만을 Detect 및 Crop 하여 Jetson 에 저장한 후 클라우드 서버인 AWS S3(이하 S3)의 Bucket 에 이미지를 업로드한다. 업로드가 완료되면, 이미지 분석을 위해 AWS Rekognition 을 호출하는데 이때 분석하고자 하는 이미지의 S3 주소를 함께 넘겨준다. 그 결과로 해당 이미지에 대한 회원 id 가 JSON 형태로 반환된다.

마지막으로, 얼굴 인식을 통해 찾은 회원 id 와 매칭되는 정보를 DynamoDB 에서 읽어 웹으로 광고를 송출해야 하는데 이 과정은 모두 Serverless 로 작동된다. AWS API Gateway(이하 API Gateway)에 회원 id 를 request 하면 AWS Lambda(이하 Lambda)가 트리거 되어 해당 고객에게 보여줄 광고를 DynamoDB 에서 읽어와 그 값을 반환한다. 이 과정을 거치면 최종적으로 디스플레이를 통해 고객에게 광고가 보인다. 구체적인 시나리오는 2.2.4 에 기술되어있다.

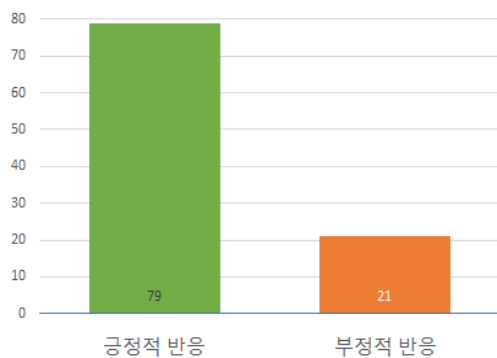
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

1.2 추진 배경 및 필요성

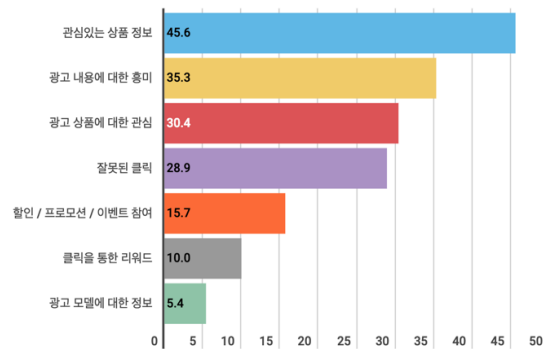
1.2.1 마켓에서의 광고의 역할과 중요성

광고는 소비자의 구매 행동에 영향을 미치는 중요한 역할을 한다. 특히, 마켓의 소비자는 많은 양의 정보를 접하며 동시에 구매 여부를 결정한다. 이때, 광고는 소비자들이 제품과 서비스에 대해 인식하고 제품의 구매를 유도한다. 'Dalia Susniene'의 연구 결과를 통해 155 명의 설문 참여자 중 79%가 광고에 긍정적인 반응을 보이는 것을 알 수 있다[그림 1]. 지루하지 않고 재미있는 광고는 소비자에게 제품을 쉽게 인지시킬 수 있도록 만들기 때문이다^[1].

광고에 대한 긍정적 반응은 맞춤형 광고를 접하였을 때 더욱 효과를 발휘한다. DMC 미디어의 2018 DMC 리포트 종합 보고서 '스마트폰 인앱 광고 접촉 태도'에 따르면, 대부분 사용자가 광고에 접근하는 가장 큰 이유는 '관심 있는 상품 정보가 나와서(45.6%)'라고 한다. 그리고 '광고 내용에 대한 흥미(35.3%)'나 '광고 상품에 대한 관심(30.4%)'이 광고 접근에 큰 영향을 미친 것으로 보아 맞춤형 광고 송출이 사용자(소비자)가 광고에 관심을 가지도록 하는 중요한 요인으로 보인다[그림 2].



[그림 1] 상품 광고에 대한 설문 참여자들의 반응



[그림 2] 사용자가 맞춤형 광고에 관심을 갖는 심리적 요인

마케팅 환경에서 광고는 더욱 중요한 의미가 있다. 상품의 생산자와 판매자는 사람들에게 제공되는 광고가 실제 구매까지 이루어지도록 창의적인 광고를 제작하여 마케팅을 관리한다. 그리고 광고는 소비자에게 제품과 서비스를 알리는 의사소통 기능을 수행한다. 즉, 광고는 제품과 서비스에 대해 긍정적 인식을 심어주고 구매로 이어질 수 있도록 유도하며, 실제로 해당 상품과 서비스의 수요를 증가시키는 역할을 한다^[2].

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

실제로 광고의 영향력에 대해 다룬 논문^[3]을 살펴보면, 제품 및 구매 행동에 대한 소비자의 관심을 높이는데 광고가 핵심 요소라는 점을 알 수 있다. 해당 논문에서 수행한 연구를 통해, 창의적으로 잘 만들어진 광고가 소비자의 구매에 강력한 영향을 미친다는 것을 확인했다. 우리나라의 대표적인 온라인 마켓 SSG닷컴의 광고를 살펴보면 이러한 경향성을 찾아볼 수 있다. SSG닷컴에서 '쓱'이라는 개념으로 진행된 광고는 독창적인 색감의 이미지와 유명 연예인의 재치 있는 내용 구성을 통해 매출을 32%까지 증가시킬 수 있었다. 이는 광고가 소비자의 구매에 미치는 막대한 영향을 증명한 셈이다^[4].

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

1.2.2 오프라인 시장의 광고 현황

오프라인 마켓의 광고를 살펴보면 온라인 광고보다 전통적인 마케팅이 이루어지고 있다. 예를 들어 대형마트인 '홈플러스'와 '이마트'의 경우 종이 형태의 할인쿠폰이나 전단지를 우편을 통해 전달하거나 매장 방문 고객에게 직접 배부하고, 해당 광고 내용을 다수의 고객이 볼 수 있도록 에스컬레이터 옆에 부착한다.

하지만 이러한 상품의 광고 방식은 비효율적이다. 매장에 방문한 고객이 [그림 3], [그림 4]와 같은 전단지를 마켓 내부에서 통행 중에 살펴보는 것은 드문 일이다. 또한, 제공되는 할인 쿠폰의 경우, 고객에게 제품에 대한 흥미를 제공하지 못한다. 할인 쿠폰을 통해 소비자가 할인된 제품을 구매하도록 유도하지만, 해당 제품에 대한 부가적인 정보가 없는 소비자들이 해당 쿠폰만 보고 제품을 실제로 구매할 가능성은 적다.



[그림 3] 홈플러스 전단지

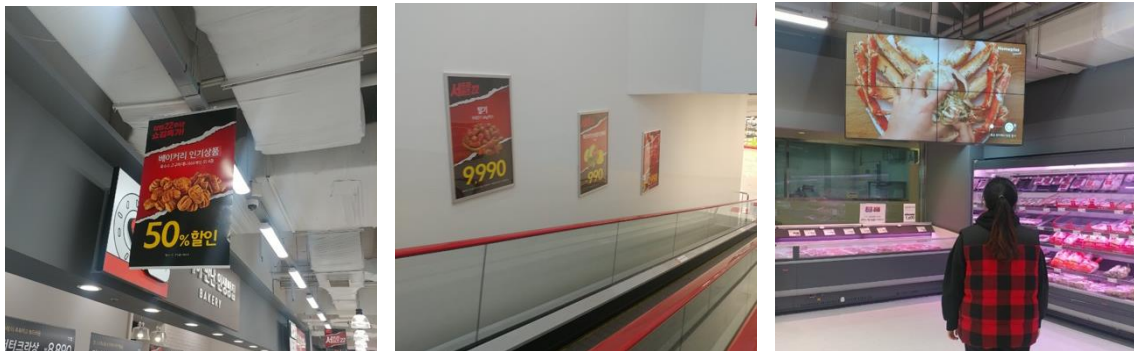


[그림 4] 이마트 전단지

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

1.2.3 오프라인 시장의 광고 기술 발전 현황 – 디스플레이 중심의 영상 광고

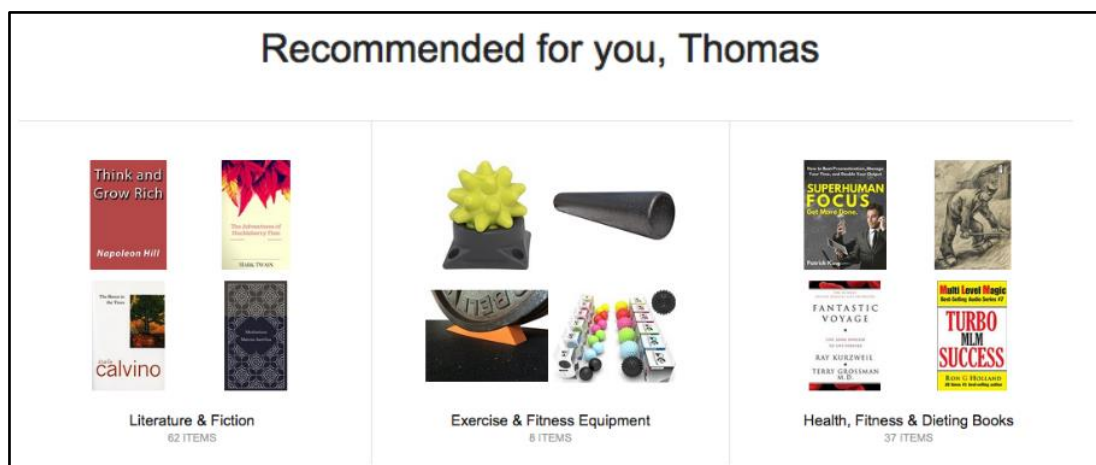
과거에 대형 마트에서 광고하는 방식은 앞에서 나타난 종이 전단지를 적극적으로 활용하는 방향으로 진행되어 왔다. 하지만 정적인 광고 방식의 한계를 느끼고 대형 마트에서는 구매욕을 더 강하게 일으킬 수 있는 유동적인 영상 광고를 채택하게 된다. 이러한 영상 광고는 특정 상품에 대한 사용법이나 활용 방안을 보여줌으로써 사용자에게 상품 가치를 전달하지만, 사용자와 특별한 상호작용을 하지 않는다는 점에서 기존의 방식과 차이가 없다[그림 5, 6].




[그림 5] 홈플러스에서의 광고 현황

1.2.4 온라인 시장의 광고 현황

한편, 아마존닷컴은 고객의 주문기록을 바탕으로 이메일 또는 홈페이지를 통해 고객에게 제품을 추천한다[그림 6]. 실제로 이 광고에 노출된 소비자의 35%가 구매까지 이루어진다^[5].



[그림 6] 온라인 고객 맞춤 광고

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


2 개발 내용 및 결과물

2.1 목표

이 프로젝트는 회원제 오프라인 마켓에서 개인 맞춤형 유동적 광고를 보여줌으로써 마케팅 효과를 증진하고, 판매량 증진을 통해 기업의 매출을 높이는 것을 목적으로 한다. 고객의 구매 내역 데이터 활용과 얼굴인식이 가능한 회원제 오프라인 마켓에서, 맞춤형 광고를 통해 매장 내 제품의 마케팅 효과를 극대화할 수 있다. Flex Ads 를 통해 송출되는 광고는 매장에 방문하는 고객이 흥미를 느낄만한 제품, 즉 실제로 구매 가능성이 큰 제품을 추천하게 된다. 일반 마켓이 아닌 회원제 오프라인 마켓이라는 제약조건에서의 구현 이유는 계획서 2.2.6 (현실적 제약요소 및 해결방안 : 3. 기타)에 상세히 기술되어 있다.

혼자 자취하는 대학생들에게 육아용품이나 어린이용 장난감 광고는 아무런 쓸모가 없으며, 평소 과일과 채소만 구매하는 채식주의자에게 삼겹살 세일 정보가 담긴 광고는 오히려 마트에 대한 반감을 갖게 할지도 모른다. 이는 개개인의 특성을 고려한 맞춤형 광고가 점차 중요해지고 있음을 의미한다.

기술적 측면에서 프로젝트의 목표는 딥러닝을 이용한 얼굴 인식 시스템과 고객의 구매 내역을 기반으로 한 광고 추천 시스템, 그리고 Serverless 환경에서의 광고 송출 시스템의 구현이다. 모든 작업이 클라우드에서 진행되기 때문에, 회원 정보와 구매 내역 데이터 등의 증가에 따른 컴퓨터 자원을 확장하여 사용할 수 있다는 점에서 성공적인 마케팅 솔루션의 제공이 가능하다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

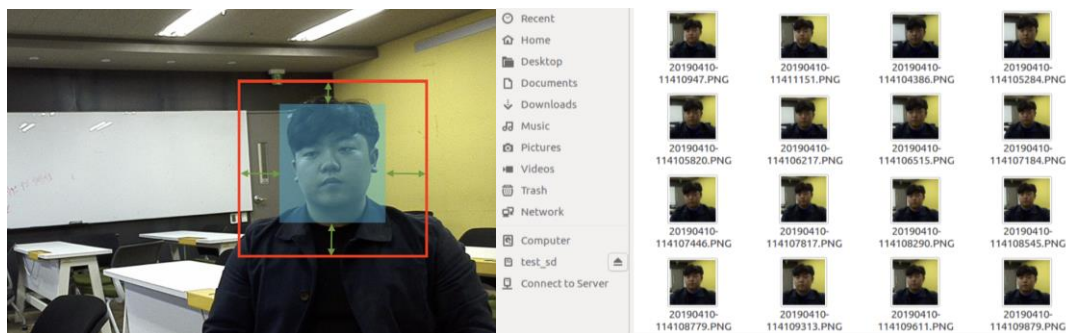
2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

2.2.1.1 고객의 얼굴 Detect 및 얼굴 이미지 생성

NVIDIA Jetson TX2 모듈에서 작동하는 딥러닝 모델에 의해 얼굴이 detect 되는 것, 얼굴을 인식하여 Cropping 하는 과정에서 FaceNet-120 모델을 사용하는 것, 그리고 네트워크 오버헤드를 최소화하기 위해 Crop 된 이미지를 전송하는 것은 최초 계획과 같다.

Jetson 모듈에서 얼굴을 detect 한 후 얼굴 영역만을 Bounding Box 처리한다. Rekognition 을 통해 해당 이미지에 포함된 얼굴이 누구인지 판별하는데, 이때 얼굴 이미지 input 은 얼굴이 화면을 꽉 채우지 않은 상태이고, 사람의 머리와 어깨가 포함되어야 한다. 따라서 저장할 시에 Bounding Box 처리된 영역을 좌우는 width * 0.6, 상단은 height * 0.3, 하단은 height * 0.6 만큼 확장하여 Cropping 하도록 구현하였다[그림 7].



[그림 7] Crop 된 얼굴 이미지

또한, 특정 구간을 지나갈 때만 얼굴을 detect 할 수 있도록 Cropping 시 얼굴 영역의 픽셀 사이즈에 제약 조건을 두었다. 해당 조건은 두 가지를 고려하여 결정된 것으로, 하나는 Rekognition 시스템에서 input 얼굴 크기 최소 조건을 충족시켜야 하는 점, 다른 하나는 너무 큰 얼굴 이미지가 저장될 경우 네트워크나 File I/O 의 오버헤드가 발생하여 인식 속도가 느려진다는 데에 있다. 이러한 점을 고려하여 detect 된 얼굴 영역의 크기가 120px 이상, 200px 이하일 경우에만 이미지를 저장하도록 하였다. 또한 얼굴이 detect 될 때마다 모든 얼굴 이미지가 저장된다면 한정적인 메모리 자원을 불필요하게 많이 사용하는 것이기 때문에 테스트 후 적절한 주기로 여겨진 1 초마다 이미지를 저장한다. filename 의 sec 부분과 flag 를 이용하여 1 초에 한 번씩 저장하도록 했다.

추가된 해당 작업은 LoadImage 코드에 Image Cropping 과 Save 함수 형태로 구현하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.1.2 얼굴 데이터 분석을 통한 회원 인식

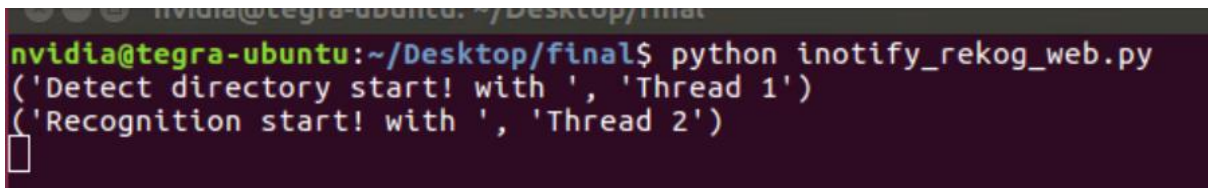
Crop 된 얼굴 이미지를 S3 에 저장한 후 Lambda 를 사용하여 Rekognition 으로 전달(S3 → Lambda → Rekognition) 하려고 계획했으나, 더 간단하고 빠른 속도를 위해 'Jetson TX2(Local) → python 을 이용한 Local Trigger → S3 업로드 → python 을 이용한 Rekognition 분석' 방향으로 Architecture 를 새롭게 구성하였다.

우선 인식하고자 하는 얼굴 이미지의 S3 주소를 Rekognition 의 Image 주소로 입력하여 response 값을 JSON 형태로 반환 받는다. 해당 JSON 데이터에는 회원 번호가 ExternalId 라는 형태로 저장되어 있으며, 해당 값을 추출하여 user_id 변수에 저장하게 된다. Local Trigger 인 inotify 는 생성된 이미지를 순차적으로 S3 에 업로드하기 때문에 Rekognition 에서 인식될 때까지의 지연 시간을 고려하지 않아 문제가 발생한다. 해당 문제를 예시를 통해 설명하고자 한다.

Crop 된 얼굴 이미지가 Local 에 1 초마다 생성된 시간의 파일명을 가지고 저장한 후, 저장된 이미지를 S3 에 업로드 시작하는 시점부터 Rekognition 을 통해 해당 얼굴이 누구인지 인식하여 user id 를 반환하는 것까지를 분석 과정이라 하고 3 초가 소요된다고 가정하자. 이때 마켓에 입장한 A 고객과 B 고객이 각각 1 초 ~ 2 초, 3 초 ~ 7 초 동안 카메라 앞에 있었다면 01.png ~ 02.png 는 A 고객의 얼굴 이미지, 03.png ~ 07.png 는 B 고객의 얼굴 이미지일 것이다. 01.png 가 저장되는 동시에 분석 과정이 시작된다면 4 초에 분석 과정이 종료될 것이다.


논리적으로 앞사람의 얼굴 이미지의 분석 과정이 완료된 후, 다음 이미지 분석 과정이 진행되어야 하지만 이미지가 저장된 순서대로 업로드를 진행하는 inotify 특성 때문에 05.png 가 아닌 01.png 다음에 저장된 02.png 의 분석 과정이 시작된다. 따라서 B 고객의 얼굴 이미지 분석 과정은 최소 8 초 이후에 시작되며 당연히 B 고객의 맞춤형 광고는 이보다 훨씬 뒤에 송출된다.

따라서 02.png 가 아닌 05.png 를 분석하기 위해서 [그림 8]과 같이 python 의 Multithreading 을 이용하여 같은 시간대에 서로 다른 Process 를 작동하게 함으로써 Rekognition 에서 인식된 user id 가 반환된 시간 이후의 얼굴 이미지를 분석하는 것이 가능하게 구현하였다.



```
nvidia@tegra-ubuntu:~/Desktop/final$ python inotify_rekog_web.py
('Detect directory start! with ', 'Thread 1')
('Recognition start! with ', 'Thread 2')
```

[그림 8] Multithreading 작동 모습

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	등글등글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.1.3 등록되지 않은 회원 인식

Detect 되었지만 Rekognition 에 등록되어 있지 않은 사람인 경우, 즉 비회원인 경우에는 'face match miss' 가 발생하게 된다. 이 경우 회원 등록을 유도하는 광고를 노출해 광고 시스템을 이용할 수 있도록 하였다.

2.2.1.4 구매 내역 기반의 광고 추천

구매 내역 기반의 광고 추천"에 대한 개발 내용은 최초 계획한 것과 동일하다. 따라서 수정 내용에 대한 설명 없이 광고 추천 알고리즘에 관해 기술하였고 데이터를 관리/저장하는지 나타내었다.

Instacart 에서 제공하는 약 300 만 개의 구매 기록과 제품 정보를 사용하였다. 이 구매 기록은 RDS 에 테이블 형태로 저장되어 관리된다[그림 9].

```
mysql> select * from aisles limit 10;
```

aisle_id	aisle
1	prepared soups salads
2	specialty cheeses
3	energy granola bars
4	instant foods
5	marinades meat preparation
6	other
7	packaged meat
8	bakery desserts
9	pasta sauce
10	kitchen supplies

```
mysql> select * from order_products_prior limit 10;
```

order_id	product_id	add_to_cart_order	reordered
2	33120	1	1
2	28985	2	1
2	9327	3	0
2	45918	4	1
2	30035	5	0
2	17794	6	1
2	40141	7	1
2	1819	8	1
2	43668	9	0
3	33754	1	1


[그림 9 -1] 제품이 속한 aisle(소분류) 정보

[그림 9 -2] 구매 정보(구매한 제품, 재구매 여부 등)

```
mysql> select * from orders limit 10;
```

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
1	112108	train	4	4	10	9
2	202279	prior	3	5	9	8
3	205970	prior	16	5	17	12
4	178520	prior	36	1	9	7
5	156122	prior	42	6	16	9
6	22352	prior	4	1	12	30
7	142903	prior	11	2	14	30
8	3107	prior	5	4	6	17
9	139016	prior	14	0	19	5
10	135442	prior	4	6	8	8

[그림 9 - 3] 주문 정보 (주문이 발생한 시간 및 요일, 이전 구매일로부터 몇 일 만에 방문하였는가 등)


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

RDS 에 저장된 주문기록을 바탕으로, EC2 인스턴스에서 boosting 으로 구현된 모델로부터 회원이 가장 구매할 것 같은 제품을 예측한다. 16 개 vCPU 와 128GiB 메모리를 가진 r5.4xlarge 타입을 사용하였다. 예측 모델로 python 에서 제공하는 XGboost 를 이용하며, 데이터를 약 321 만 개, 13 만 개, 7 만 5 천 개로 구분하여 train set, test set, validation set 으로 사용하였다.

이후, boosting tree 를 생성하기 위하여 20 개의 feature 를 생성하였다. 이때 지정한 feature 이름과 그에 대한 간단한 설명을 아래 명시하였다.

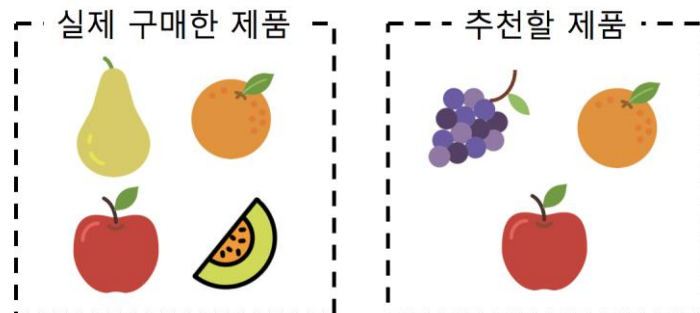
1. 회원을 기준으로 한 feature

- user_dep : 각 회원의 대분류별로 구매할 확률
- user_aisle : 각 회원의 소분류별로 구매할 확률
- user_dow : 각 회원의 요일별로 구매할 확률
- user_hour : 각 회원의 시간별로 구매할 확률
- organic : 각 회원이 유기농(organic) 상품을 구매할 확률
- reordered_latest : 회원이 구매한 상품이 바로 전 쇼핑 목록에서도 있었는지 여부
- reordered_count : 각 회원이 특정 상품을 총 몇 번 구매했는가
- reordered_sum : 각 회원이 특정 상품을 몇 번 재구매 했는가
- useritem_mean_cart : 각 고객이 구매한 제품의 평균
- useritem_min_cart : 각 고객이 구매한 제품의 최소값
- useritem_max_cart : 각 고객이 구매한 제품의 최대값
- useritem_median_cart : 각 고객이 구매한 제품의 중위값
- useritem_std_cart : 각 고객이 구매한 제품의 표준편차
- timezone : 상품을 구매한 시간에 따른 구간(아침, 낮, 밤, 새벽)
- together : 각 회원이 한 번 구매할 때 같은 제품을 몇 개 샀는가
- favor_product : 각 회원이 가장 선호하는 제품의 순위
- order_ratio_bychance : 각 회원이 특정 상품을 처음 구매했을 때와 마지막으로 구매했을 때의 기간 동안 해당 상품을 얼마나 구매했는가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

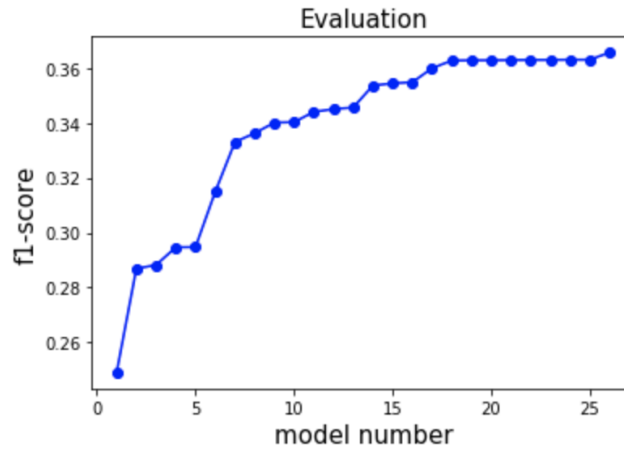
2. 상품을 기준으로 한 feature

- dowRank : 요일별로 많이 팔린 상품의 순위
- probCount : 상품별로 구매된 총합
- prod_rank : 많이 팔린 제품의 순위

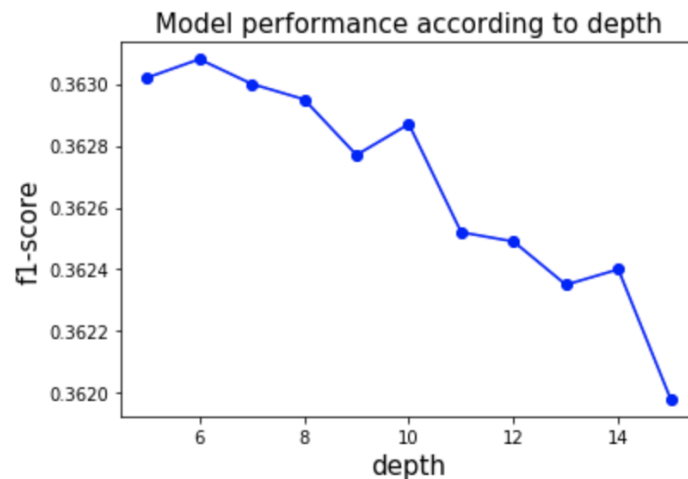


[그림 10] f1-score 설명을 위한 그림

위 feature로부터 생성된 모델은 f1-score 값이 높은 것으로 선택하였으며 이는 kaggle Instacart 대회에 제출하여 확인하였다. 여기서 f1-score 는 precision 과 recall 의 조화 평균값이다. 예를 들어, 우리가 고객에게 포도, 귤, 사과를 추천했다고 가정하자[그림 10]. 이때, 실제로 고객이 배, 귤, 사과, 멜론을 샀으면 $\text{precision} = 2/3$, $\text{recall} = 2/4$ 이다. 따라서 $\text{f1-score} = 2 * (2/4 * 2/3) / (2/4 + 2/3)$ 로 0.571 이 된다. 우리는 kaggle 대회에서 최고 기록인 0.4 를 목표로 하였으며, 구현한 feature 의 조합하여 17 번의 모델 개선을 통해 약 0.36 의 평가 결과를 보였다. 따라서 목표했던 모델 성능의 90%를 달성했다.




[그림 11] 모델에 따른 추천 알고리즘 성능



[그림 12] boosting tree 의 depth 에 따른 모델 성능 평가



[그림 13] boosting tree 의 threshold 에 따른 모델 성능 평가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

우리는 앞서 설명한 내용과 같이, f1-score 를 지표로 고객에게 예측된 여러 제품을 추천할 수 있다[그림 14-1]. 하지만, 실제 마켓에서 즉각적인 광고 송출을 목표로, 추천할 여러 제품 중 랜덤 1 개[그림 14-2]를 선택하여 추천하였다.

```

user_id,product_id
3,21903 47766 43961 39190 18599
11,14947 8670 27959 34658 35948 8309
12,13176 14992 7076 10863 28134
15,14715 11266 12427 10441 196 37710
16,21903 5134 21137 24852 17948 41950
19,17008 35123 2192 15599 34690 31487 15131 12108
20,9387 22362 13914 41400 6184 13575 46061
22,17794 22935 27845

```

[그림 14-1] user_id 에게 추천할 수 있는 product 목록

```

user_id,product_id
3,21903
11,8670
12,7076
15,196
16,24852
19,12108
20,41400
22,17794

```

[그림 14-2] user_id 에게 추천할 수 있는 product 중 랜덤 1 개를 선택한 모습

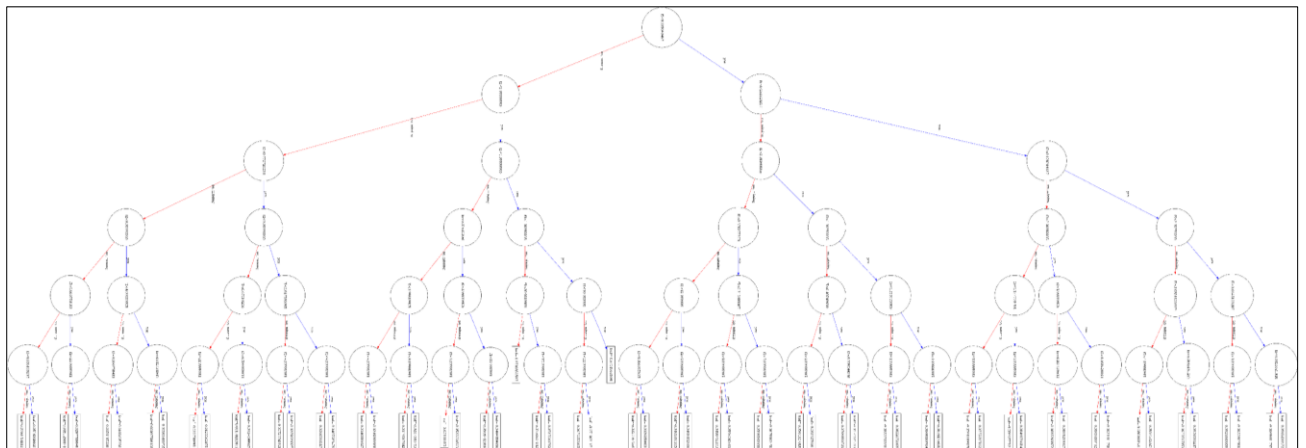
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	등글등글	
	Confidential Restricted	Version 2.2	2019-MAY-28

최종 모델을 통해 나온 결과(고객에게 추천할 제품)를 DynamoDB 에 저장한다. update_ver 을 기록하여 추천할 광고의 버전 트래킹 또한 가능하다[그림 15].

user_id	update_ver	aisle	bucket_url	product_name
2430	1	bears coolers	https://s3- /31716.png	312 Urban Wheat
11985	1	first aid	https://s3- /719.png	Scars Cream + SPF 30
21210	1	beers coolers	https://s3- /2054.png	Belgium Beer
31751	1	chips pretzels	https://s3- /16345.png	Spinach & Kale
58169	1	juice nectars	https://s3- /3181.png	Juice Cocktail, Cranberry Blood Orange

[그림 15] DynamoDB 에 추천 결과가 저장된 모습

최종 모델로 나온 boosting tree 는 아래와 같다

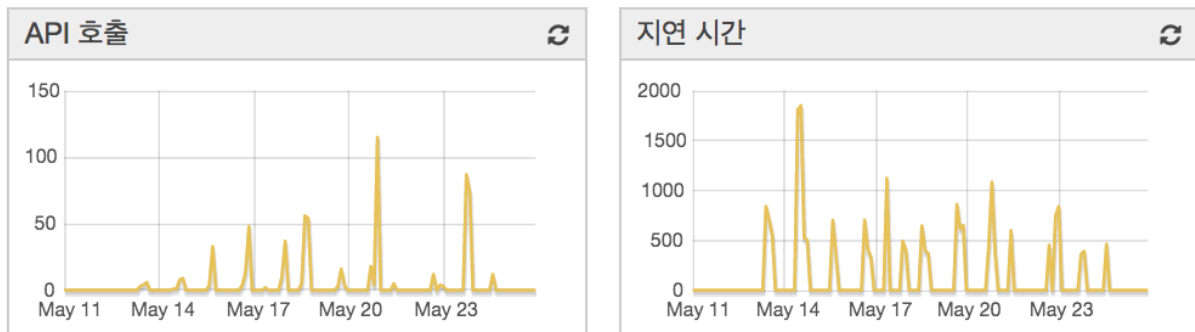


[그림 16] 추천 알고리즘으로 사용된 boosting tree

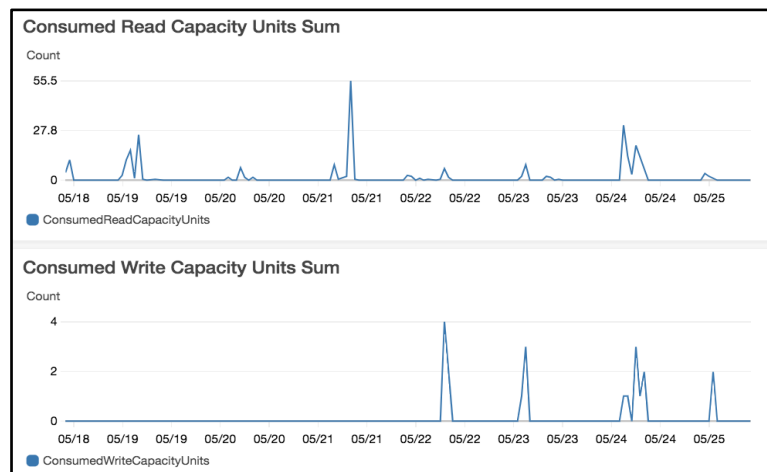
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	등글등글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.1.5 Serverless 환경에서의 광고 송출


송출하고자 하는 광고의 이미지는 S3 에 저장 되어있다. 2.2.1.2 에서 반환되는 회원 ID 를 이용하여 API Gateway 에 요청을 보내면, Lambda 는 트리거 되어 2.2.1.3 의 결과로부터 나온 추천할 광고 정보를 DynamoDB 에서 찾아 반환한다. 이 정보(광고)는 광고를 송출하는 디스플레이를 통해 고객에게 보인다[그림 19]. 한편, API Gateway 와 DynamoDB 에 요청한 내용은 AWS CloudWatch(이하 CloudWatch)를 통해 로그를 살펴볼 수 있다. [그림 17, 18]



[그림 17] 요청에 따른 API Gateway 상태



[그림 18] 요청에 따른 DynamoDB 상태

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28



[그림 19] 화면에 출력되는 광고

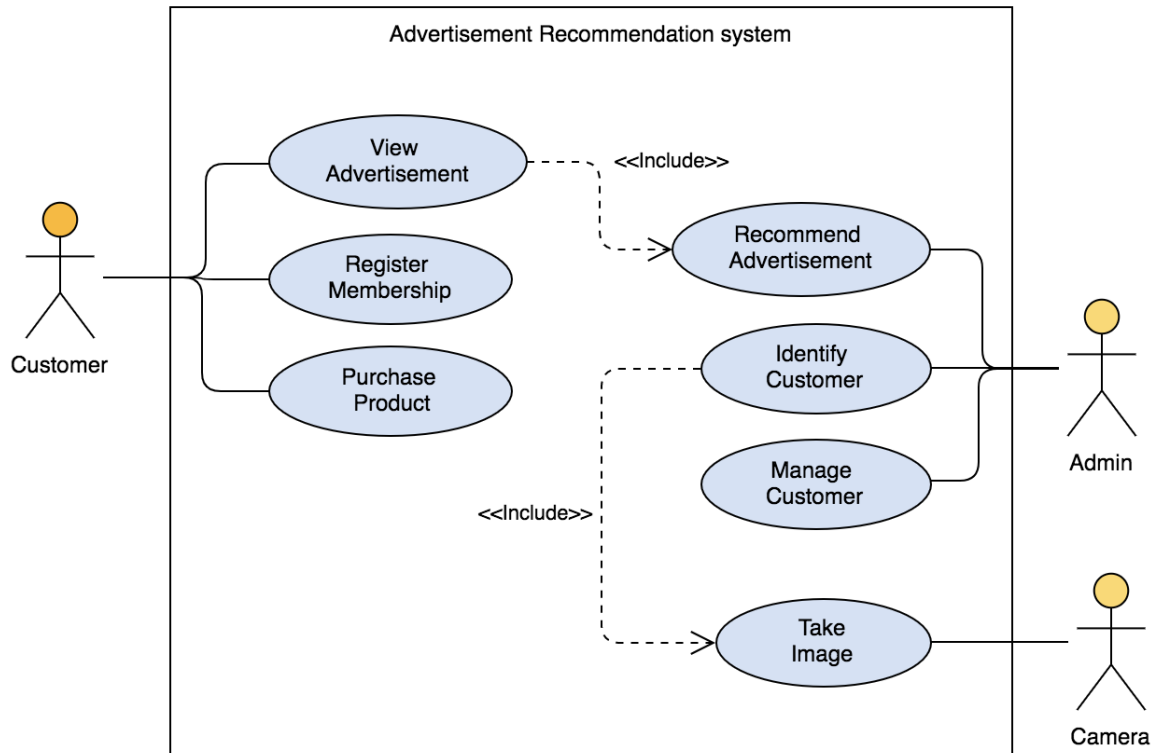
한편, 회원 등록을 하기 전의 예비 회원의 경우 주문 기록이 없으므로 추천 광고 또한 존재하지 않는다. 이 경우에는, [그림 20]와 같이 회원 등록을 유도하는 이미지를 띄운다.



[그림 20] 예비 회원에게 출력되는 화면


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.2 시스템 기능 요구사항



[그림 21] Use-Case Diagram

[그림 21]는 우리가 계획한 시스템 요구 사항을 use-case diagram 으로 나타낸 것이다. 시스템 관리자는 이 시스템을 통해 고객을 카메라로 탐지하고 인식하며 주문기록을 관리할 수 있다. 이를 바탕으로 추천 제품 또한 가능하다. 그리고 고객은 이 시스템을 통해 회원 등록을 하고, 주문 기록을 제공함으로써 개인 맞춤형 광고를 받는다. 이 요구사항들은 모두 완료되었으며 2.2 연구/개발 내용을 통해 확인할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.3 시스템 비기능(품질) 요구사항

(1) 얼굴 인식 대상

우리는 한 사람씩 시스템을 통과하는 상황과 다수의 사람이 동시에 시스템을 통과하는 상황, 두 가지 상황을 구현하는 것을 고려해볼 수 있다.

→ 여러 명이 인식되는 경우는 다중 얼굴인식과 여러 사람의 취향을 동시에 고려하는 추천 알고리즘이 필요하다. 이는 짧은 테스트 시간, 데이터를 통한 모델 성능 평가 불가능, 다중 인식으로 인해 발생할 수 있는 네트워크 오버헤드 등의 이유로 인식된 고객들에게 공통적인 추천 광고를 보여주는 것은 불가능하다고 판단하였다. 한편, 고객 1 명을 대상으로 광고 송출하는 것은 구현이 완료되었다. 이 기능을 통해 여러 명이 인식된 상황에서 먼저 인식된 1 명에 대해 개인 맞춤형 광고를 제공한다.


(2) 얼굴인식을 위한 딥러닝 모듈의 요구사항

정확도를 확인하기 위해 팀원 5 명과 교수님, 지인 총 13 명에 대해서 얼굴 인식 테스트를 421 회 진행한 결과, 384 회 성공하였다. 즉, 93.5%의 정확도를 보인다. 목표했던 90% 정확도를 초과하여 달성하였다.

(3) 추천 알고리즘 동작의 요구사항

실제 마켓에서 회원의 얼굴이 인식되고 바로 뒤에 추천 알고리즘이 실시간으로 동작한다면 누적된 구매기록을 바탕으로 새 모델을 구축해야 하는데 이는 매우 큰 시간을 요구한다. 따라서 광고 추천을 위한 알고리즘은 실시간이 아닌 미리 수행되는 것으로 정하였다.

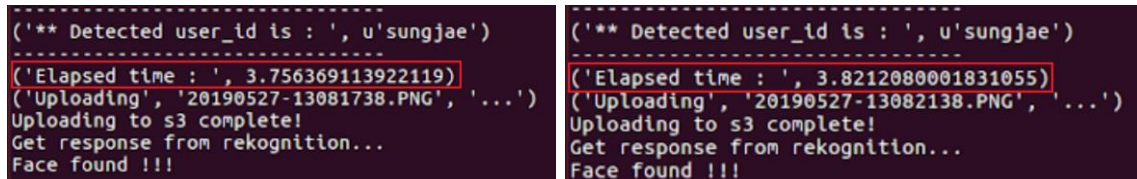
→ 알고리즘이 미리 수행되는 간격은 우리가 사용한 데이터에 각 사용자가 다음 구매까지 걸린 기간 정보를 이용하여 정했다. 고객의 제품 구매가 이루어지는 간격을 Sorting 하여, 상위 25%에 위치한 '4 일'을 기준으로, 알고리즘을 통해 추천할 광고 정보를 업데이트하였다. 즉, 4 일 간격으로 추천할 광고 정보를 업데이트한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

(4) 개인 맞춤형 Advertisement 제공에 걸리는 시간

Advertisement 를 제공하는 데 걸리는 시간은 Face Detection 에 걸리는 시간, Network I/O 로 인해 S3 에 업로드 하는 시간, Rekognition 을 호출하여 결과를 받을 때까지 걸리는 시간을 고려해야 하며 3 초를 목표로 계획했다.

→ 우리가 테스트한 환경에서 얼굴 인식 후 3 초에서 최대 5 초 이내에 개인 맞춤형 광고가 송출되었기 때문에 목표에 달성했다고 할 수 있다. (실행 초기 로딩 시간 제외) 와이파이와 유선 랜을 통한 데이터 전송속도가 다르므로, 유선 랜을 되도록 이용하였다 [그림 22].



```

(** Detected user_id is : ', u'sungjae')
('Elapsed time : ', 3.756369113922119)
('Uploading', '20190527-13081738.PNG', '...')
Uploading to s3 complete!
Get response from rekognition...
Face found !!!

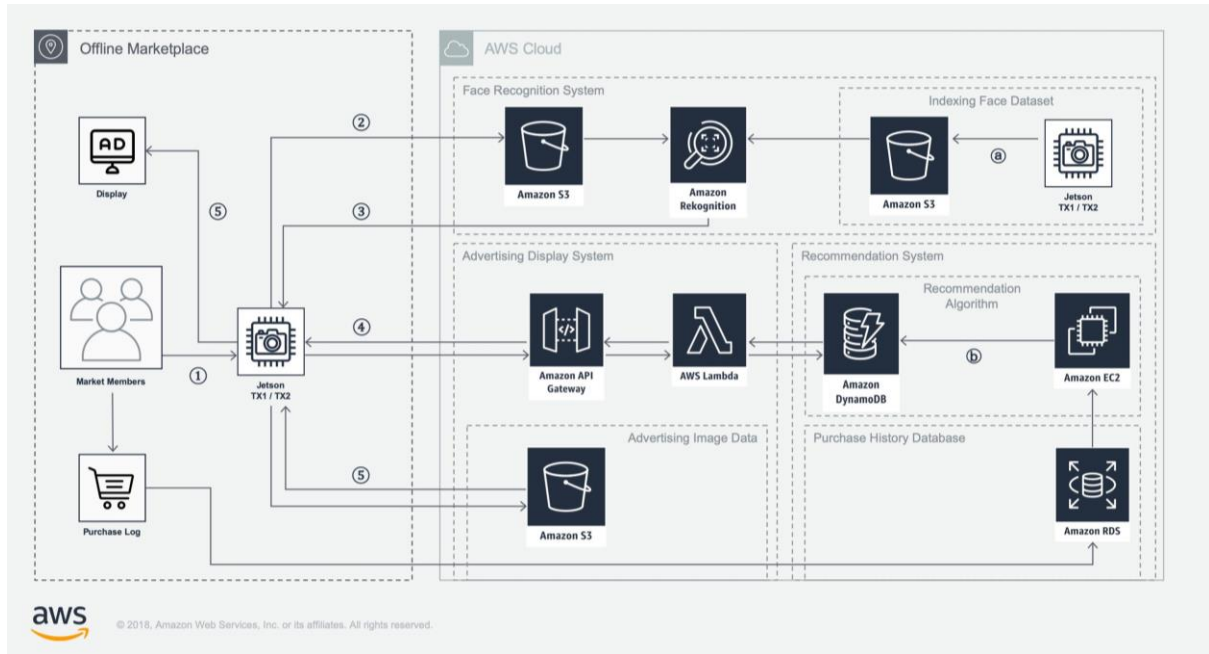
(** Detected user_id is : ', u'sungjae')
('Elapsed time : ', 3.8212080001831055)
('Uploading', '20190527-13082138.PNG', '...')
Uploading to s3 complete!
Get response from rekognition...
Face found !!!

```

[그림 22] 광고 나오기까지 걸리는 시간

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


2.2.4 시스템 구조 및 설계도



[그림 23]

Flex Ads 프로젝트는 일련의 과정이 AWS 클라우드 환경에서 이루어지며 그 구조는 [그림 23]과 같다. 시스템 구조에 대한 설명은 2.2의 연구/개발 내용 및 결과물에 기술되어 있으며 간단하게 정리하면 다음과 같다.

- ① Jetson에 연결된 카메라에 포착된다.
- ② 딥러닝을 통해 얼굴이 detect 되고, S3로 전송된다.
- ③ 클라우드에서 Rekognition을 통해 얼굴이 인식된다.
- ④ 인식된 회원 정보를 이용해 송출할 광고 정보를 API Gateway를 통해 DynamoDB에서 가져온다.
- ⑤ 해당하는 광고를 S3에서 가져와 디스플레이로 송출한다.
- ⑥ 일정 기간을 정하여 회원의 구매내역을 XGBoost를 통해 학습하여 가장 구매 가능성이 높은 상품에 대한 광고 정보를 입력한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.5 활용/개발된 기술

(1) Edge Computing

Edge Computing 방법을 통한 클라우드 기반의 얼굴인식 시스템을 개발하였다.

Edge Computing 은 클라우드 기반에서 IoT 기기를 제어하는데 사용되는 개념이다. 기존의 방법은 IoT 에서 인식된 데이터를 전처리 없이 그대로 클라우드를 통해 공유하여 네트워크 오버헤드가 컸다면, Edge Computing 에서는 데이터를 수집하는 IoT 단계에서 데이터 처리를 통해 핵심 데이터만 최소화하여 클라우드로 전송하는 방법이다. Flex Ads 시스템에서는 딥러닝 모듈 Jetson TX2 를 이용하여 촬영된 이미지에 딥러닝 기법을 적용하여 얼굴 주변부 이미지만을 클라우드로 전송하게 된다. 이를 통해 전송 데이터의 크기를 대폭 축소하였으며, 기존의 방법보다 빠른 속도로 처리가 가능해졌다.


(2) Facenet-120

지금까지 나온 얼굴 인식 시스템 가운데 가장 정확한 인식률을 실현한 것들 중 하나이다. 인공지능 신경망에 얼굴인식 능력을 갖게 하기 위해 Fddb 의 얼굴 데이터셋을 24,000 회 학습한 모델이다.

(3) XGBoost

회원에게 맞춤형 광고를 추천하기 위해 XGBoost 알고리즘을 사용했다.

XGBoost 는 병렬처리를 통해 Gradient Boosted Decision Tree 의 속도와 성능을 향상한 알고리즘이다. 회원들에게 어떤 상품을 추천할 것인지 예측하는 XGBoost 모델의 정확성을 높이는 데에 영향을 미치는 학습 데이터의 여러 feature 에 우선순위를 부여한다. 여기서 욕심쟁이 기법을 이용해 우선순위가 높은 feature 를 최상위 노드로 하여 가지를 나누는 기준을 정하고, 가지에 가중치를 부여해 만들 수 있는 약한 모델(tree)을 여러 개 생성한다. 각 모델에서 나온 오류를 줄이기 위해 또 다른 약한 모델을 만들고 이전에 만든 트리와 더하는 과정을 반복한다. 이 결과, 최종적으로 나온 모델(tree)의 말단 노드들의 값을 통해 회원이 해당 상품을 살 것인지 예측 결과를 얻을 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.6 현실적 제한 요소 및 그 해결 방안

2.2.6.1 하드웨어 제한 요소

(1) NVIDIA Jetson TX1 / TX2 Onboard Camera 사양에 따른 현실적 제한 요소

Jetson TX2 의 카메라 사양은 [12 lanes MIPI CSI-2 | 2.5 GB/sec per lane | 1400 megapixels/sec ISP] 으로 1400 만 화소의 고화질 영상 이미지를 통해 사람이나 물체를 인식하는 데 문제가 없는 상태이다.


또한 데모 시 추가로 사용하는 카메라는 logitech C920 와 Microsoft Lifecam Studio 이고 이 두 카메라의 비디오 해상도는 1920 x 1080(FHD), 비디오 프레임은 30fps 이다.

(2) NVIDIA Jetson TX1 / TX2 GPU 사양에 따른 현실적 제한 요소

Jetson TX2 의 GPU 사양은 [256-core Pascal @ 1300MHz]로 딥러닝을 활용한 고화질 영상의 처리에 문제가 없는 수준이다. 하지만 동시에 수십 ~ 수백 명의 Face Detection 이 이루어질 경우 S3 로 이미지를 업로드하는 과정에서 네트워크 오버헤드가 크게 발생하여 처리 속도가 늦어지는 문제가 발생할 수 있다. 이러한 제한 요소를 해결하기 위하여 Face Detection 가능한 인원수의 한계를 지정하는 것이 필요하다고 판단하였고 한 번에 한명만 detect 하는 것으로 제한하였다.

(3) 메모리의 현실적 제한 요소

얼굴이 detect 될 때마다 모든 얼굴 이미지가 저장된다면 한정적인 메모리 자원을 불필요하게 많이 사용하는 것이기 때문에 테스트 후 적절한 주기로 여겨진 1 초마다 이미지를 저장한다. 또한 Image File Save 과정에서 Disk Write 가 발생하여 Frame Drop 을 확인하였으며, 이에 대한 해결방안으로 SD Card 에서의 프로그램 실행을 구현하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


2.2.6.2 소프트웨어 제한 요소

(1) Face Detection 을 위한 오픈소스 딥러닝 모델 FaceNet-120 의 라이선스

회원의 얼굴을 인식하고 Crop 하는 과정에서 딥러닝 모델 FaceNet-120 이 사용되며, 이는 MIT 라이선스가 부여되어 있으므로 복사, 수정을 포함한 상업적 사용이 가능하다. 하지만 회원의 얼굴 정보 수집에 관하여 오프라인 마켓 측에서 적절한 개인정보 수집 동의가 필요할 것으로 보인다.

(2) 대용량 구매 내역 데이터를 이용한 추천 시스템의 한계

300 만 개의 고객 구매 내역 데이터를 이용하여 추천 시스템을 구축해야 하는 상황에서, 현재는 Single machine 을 이용해 python 언어로 구현하였다. 복잡한 알고리즘을 실제 오프라인 마켓의 대규모 데이터에서 구현하기 위해서는 더 많은 컴퓨터 자원이 필요하거나 분산 처리가 필요하다고 판단했다. 현재 EC2 단일 instance 에서 추천 알고리즘이 동작한다. 메모리 기반의 인스턴스를 사용할 경우 대용량 메모리가 사용할 수 있다. 따라서 필요에 따라 인스턴스의 성능을 조절하면 대용량 데이터에 대한 고민을 줄일 수 있다. 여러 인스턴스를 클러스터화하여 사용할 수 있지만, 불필요라고 생각한다.

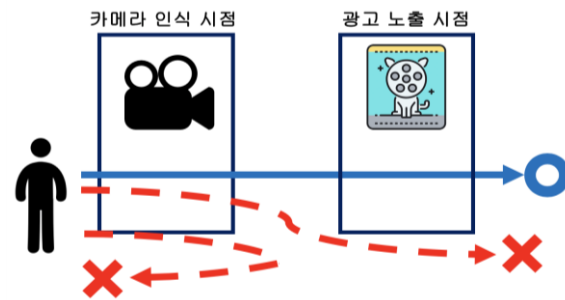
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.2.6.3 기타 제한 요소

(1) 프로젝트의 상용화 시 공간적 제약조건

이 프로젝트가 실제로 시장에 상용화될 때, 공간적인 제약사항이 존재한다. 우리는 얼굴 인식을 하고 인식된 고객들이 관심을 가질 만한 제품의 광고를 고객에게 제공한다. 하지만 얼굴을 인식하는 시점과 광고가 이루어지는 시점이 명확해야 하는데, 고객의 행동 변화 및 마트의 공간 배치 등의 변수 발생할 수 있다.

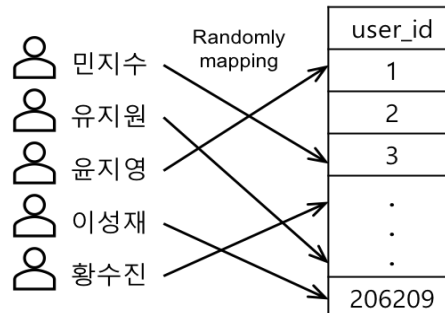
→ 우리는 고객의 얼굴을 인식하였지만, 광고가 송출되는 지점까지 도달하지 못하고 고객이 화장실 또는 동선을 변경하여 실질적으로 광고에 노출되지 못하는 경우를 고려해볼 수 있다. 하지만 우리는 고객이 카메라에 인식되었다면 반드시 광고를 마주하게 되는 상황에 놓이는 것을 가정했다. 그 외의 돌발적인 행위에 대해서는 고려하지 않았다[그림 24].



[그림 24] 실제 상용화를 위한 제약조건

(2) 얼굴 데이터의 수집 시 제약사항

현재 데이터로 확보한 고객의 수는 206,209 명이다. 프로젝트의 진행을 위해서는 실제 고객의 얼굴 데이터를 수집해야 하지만, 현재 가지고 있는 주문 기록에 존재하는 실제 고객의 얼굴 데이터는 존재하지 않는다. 따라서 5 명의 얼굴 정보를 랜덤으로 user_id 에 매핑하여 실제 고객이라고 가정하고 프로젝트를 진행했다[그림 25]. 추가로 지도 교수님 및 지인의 얼굴 데이터를 확보하였다.



[그림 25]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


(3) 고객 데이터 사용 시 제약 사항

프로젝트를 진행하기 위해서는 마켓에 방문하는 고객의 얼굴 사진 데이터와 고객의 구매 내역 데이터가 필수적이다. 두 가지 모두 개인 정보이기 때문에 이를 활용하기 위해서는 사전에 고객의 동의가 필요하다. 그렇기 때문에 비교적 고객의 동의를 얻는 방법이 쉬운 회원제 오프라인 마켓에서 프로젝트를 진행하는 것으로 시나리오를 설정했으며, 고객의 마켓 회원가입 시 '개인정보 수집·제공 동의서'를 작성하는 것을 통해 고객의 동의를 얻고 데이터를 활용할 수 있다.

2.2.7 결과물 목록

2.2 연구/개발 내용 및 결과물을 통해 결과물을 살펴볼 수 있다.

중복된 내용 기술을 피하기 위해 이는 2.2 로 대체한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2.3 기대효과 및 활용방안

마켓을 운영하는 수많은 기업[그림 26]가 저마다의 회원 시스템으로 고객을 끌어들이려는 이유가 무엇일까. 이들은 회원 시스템을 통해 더 나은 서비스와 품질을 제공하며 궁극적으로는 이윤 증가를 목표로 한다. 우리는 이 특성을 이용하여 회원 데이터를 수집하여 이윤을 극대화하는 방법을 논의해볼 수 있다.



[그림 26] 다양한 회원제 오프라인 마켓

현재 쿠팡, 11 번가, 아마존^[6]을 포함한 대부분의 온라인 마켓에서는 구매자가 이전에 검색 및 구매하였던 정보를 바탕으로 연관되는 상품들을 웹 배너에 띄워 광고하고 있다. 이러한 맞춤형 광고는 사용자에게 꼭 필요한 상품에 대한 정보를 알려주기 때문에 사용자는 관심을 가지고 해당 내용을 확인한다. 이처럼, 온라인에서만 진행되던 개인 맞춤형 광고 추천 시스템을 오프라인에 도입하여 오프라인 마켓에서도 개인 맞춤형 광고를 접하도록 한다. Flex Ads 는 회원의 구매기록을 바탕으로 회원에게 딱 맞는 맞춤형 광고를 보여준다. 고객들은 회원으로서 더 많은 서비스를 받고 있다고 느끼는 동시에, 마켓은 충성 고객을 유치와 이윤 증대도 기대할 수 있다.

이는 단순히 회원제 오프라인 마켓에만 제한되지 않고, 회원제를 적용하고 있는 다양한 업종과 분야에서 활용할 수 있다. 패스트푸드점, 카페 등에서 활발하게 이용되고 있는 무인 주문기(키오스크)를 통해 개인 맞춤형 광고를 제공할 수 있다. 또한, 고객의 구매 기록 및 얼굴 데이터와 같은 개인 정보 이용에 대한 동의가 있다면 스타필드, 코엑스와 같은 대형 복합쇼핑몰에서 활용할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	등글등글	
	Confidential Restricted	Version 2.2	2019-MAY-28

3 자기평가


꾸준히 프로젝트 Architecture 를 계획하고, 수정하여 실제 사용 가능한 시스템을 구성하였다. 기본적인 구현을 완료(얼굴인식, 광고 추천 알고리즘, 광고송출)하고, 부가적으로 시스템 기능을 향상할 수 있는 기능(고객 등록, 회원으로 등록되지 않은 고객에게 광고 송출)을 적극적으로 추가하였다.

운영자 매뉴얼을 통해 이 시스템을 관리하는 마켓 운영자는 스스로 시스템을 구축하여 사용할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

4 참고 문헌

번호	종류	제목	출처	발행 년도	저자	기타
1	Article	The Psychological Impact of Advertising on the Customer Behavior	Communications of the IBIMA.	January 2008	Sandra, Jakštienė & Susniene, Dalia & Valdas, Narbutas	
2	Journal	Importance of Creative Advertising and Marketing According to University Students' Perspective	International Review of Management and Marketing	Vol. 4, No. 3, 2014, pp.239-246	Remziye Terkan	
3	Journal	Impact of Advertising on Consumers' buying behavior through Persuasiveness, Brand Image, and Celebrity endorsement	GLOBAL MEDIA JOURNAL	VOL. 6(2):149 · DECEMBER 2013	Shumaila Ahmed and Ayesha Ashfaq	
4	Online news	SSG닷컴, '쓱(SSG)' 광고 인기에 매출 30% 급증	http://www.etoday.co.kr/news/section/newsview.php?idxno=1298782	2016-03-16		
5	Web	The Amazon Recommendations Secret to Selling More Online	http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/			
6	Web	AWS 제품	https://aws.amazon.com/ko/#			

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

5 부록

5.1 사용자 매뉴얼

서비스 대상자는 고객이며, 고객이 이 시스템을 사용하기 위해서는 주문기록과 얼굴 데이터 사용에 대한 동의서가 필요하다. 추가적인 가이드는 필요하지 않다.

5.2 운영자 매뉴얼

이 시스템은 '얼굴 인식', '추천 알고리즘', '광고 송출'의 3 개 부시스템으로 나뉜다. 따라서 통합 시스템을 구축하기 위해서 아래에 명시되어 있는 운영자 매뉴얼이 수반된다.

1. 회원 등록 방법 (회원 등록 명령어 이용)

회원 등록을 위한 일련의 과정을 구현한 코드는 다음 위치에서 확인할 수 있다.

/src/complete/registration/member_registration.sh

먼저, 시스템 운영자는 고객을 등록시키기 위해서 2.2.6.3 (2)에 기술한 것과 같이, 회원과 임의의 user_id 를 매핑해야 한다.

```

=====
Flex Ads Memeber Registration System
=====

Enter user-name for registration
Sujin ← 관리자 입력 부분

Enter user-id for registration
12120 ← 관리자 입력 부분

'Sujin' directory is created!

```

[그림 27] 터미널에서 이름, user_id 입력하는것

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


그 다음 Jetson 에 연결된 카메라를 이용하여 촬영 및 생성해야 한다. 정면 이미지, 얼굴을 왼쪽으로 돌린 이미지(요 45 도 이내), 얼굴을 오른쪽으로 돌린 이미지(요 45 도 이내), 얼굴을 아래로 기울인 이미지(피치 30 도 이내), 얼굴을 위로 젖힌 이미지(피치 45 도 이내)를 이용해 학습시키는 것을 권장한다[그림 28, 29].



[그림 28] 여러 각도의 얼굴 사진



[그림 29] Jetson 에서 촬영한 회원 등록을 위한 데이터 셋

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

2. 광고 이미지를 저장하고 공유하기






S3 Bucket 을 만든 후 Bucket 에 대한 permission 을 수정한다. 외부에서 이미지를 읽을 수 있도록 한다. 하지만 객체(광고 이미지)를 수정하고 변경할 수 없도록 제한한다.

퍼블릭 액세스

그룹 	객체 목록 생성 	객체 쓰기 	버킷 읽기 권한 	버킷 쓰기 권한 
<input type="radio"/> Everyone	-	-	예	-

[그림 30]

광고 이미지는 마트에서 적극적으로 관리해야 한다. 광고 이미지는 [광고할 제품 id].png 의 파일명 규약을 따른다. 이는 DynamoDB 에 item 을 추가할 때, 부가적인 작업을 추가하지 않기 위함이다.

 1006.png	5월 18, 2019 10:53:27 오전 GMT+0900	64.2 KB	스탠다드
 116.png	5월 18, 2019 10:53:24 오전 GMT+0900	252.0 KB	스탠다드
 1179.png	5월 18, 2019 10:53:27 오전 GMT+0900	227.8 KB	스탠다드
 1317.png	5월 18, 2019 10:53:27 오전 GMT+0900	79.9 KB	스탠다드
 13176.png	5월 18, 2019 10:53:29 오전 GMT+0900	1.1 MB	스탠다드

[그림 31]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28


3. 고객과 매칭된 광고 정보 저장하기

추천 알고리즘을 통하여 나온 결과는 [고객 id]-[추천할 제품 id]로 매핑되어 csv 파일로 저장된다. 이를 DynamoDB 에 추가하게 되는데, /src/recommendation/aws_dynamodb/create_table.py 를 이용하여 테이블을 생성하고, /src/recommendation/aws_dynamodb/add_item_to_dynamodb.py 를 이용하여 필요한 정보와 함께 DB 로 전송한다.

[고객 id] - [업데이트 version] - [제품이 위치한 코너명] - [광고 이미지 주소] - [광고할 제품]
 - [고객명]의 정보를 가진다.

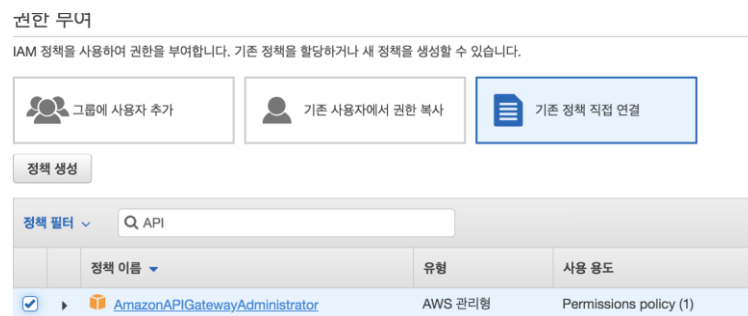
user_id	update_ver	aisle	bucket_url	product_name	user_name
115757	1	milk	https://s3-us- /33664.png	2% Reduced Fat Milk	jiwon
91185	1	mint gum	https://s3-us- /1179.png	Polar Ice Sugar Free Gum	Sung Jae
21210	1	beers coolers	https://s3-us- /2054.png	Belgium Beer	Su Jin
98765432	1	canned meals beans	https://s3-us- /45871.png	mixed bean	Ji Young
12345678	1	ice cream ice	https://s3-us- /33477.png	Bunny Tracks Ice Cream	Ji Soo

[그림 32]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

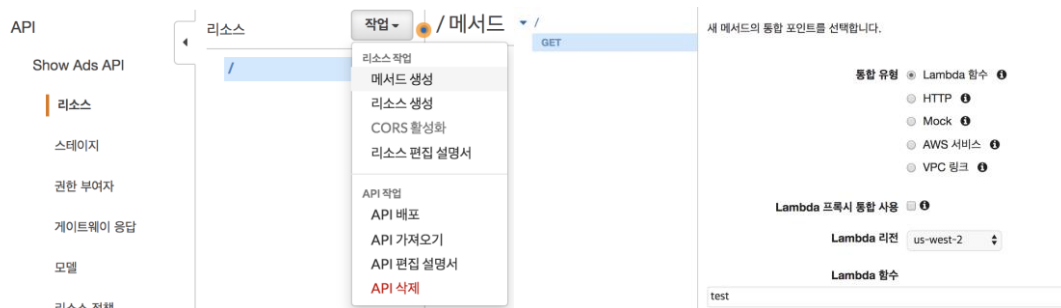
4. 요청에 의해 고객에게 맞는 광고 정보를 반환하는 API Gateway 설정하기

먼저 /src/recommendation/aws_apigateway/create_apigateway.sh 를 이용하여 API Gateway 를 생성한다. API Gateway 가 Lambda 를 트리거 할 수 있도록 IAM 을 이용하여 Lambda 에 권한을 부여한다.



[그림 33]

요청을 수행하기 위해서 resource 에서 GET 메소드를 생성한다. 인식된 고객의 id 를 입력으로 받아, 이 고객에 대한 광고 정보를 반환하는 Lambda 를 GET 메소드에 연결한다.



[그림 34]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 2.2	2019-MAY-28

GET 메소드의 통합요청에서 Mapping Templates 를 추가한다. json 의 형태로 request 를 받도록 설정한 것이다.

▼ Mapping Templates

Request body passthrough ☒ When no template matches the request Content-Type header ⓘ

☐ When there are no templates defined (recommended) ⓘ

☐ Never ⓘ

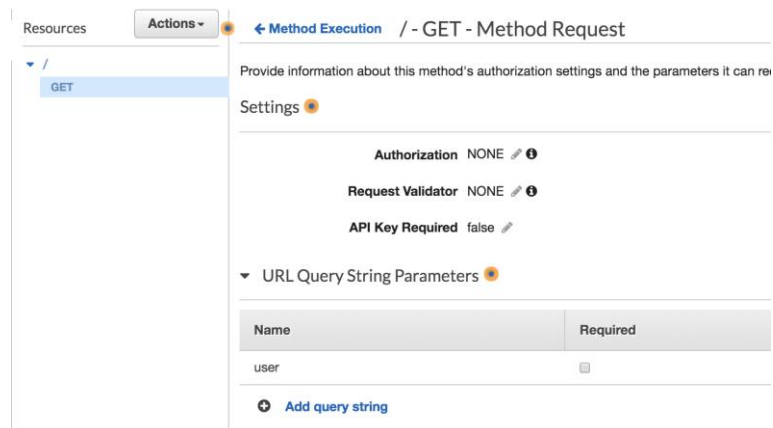
Content-Type

application/json

+ Add mapping template

[그림 35]

url 에서 query 를 받도록 GET 메소드의 메소드 요청에서 query string 을 수정한다.



[그림 36]

stage 를 생성하여 API 를 배포한다.



[그림 37]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

5. 추천 알고리즘 실행하기

작성한 광고 추천 알고리즘은 EC2 에서 구현되어 있으며 AMI 를 통해 배포할 수 있다. AMI 를 관리자에게 공유하면, 관리자는 이 AMI 로부터 EC2 인스턴스를 생성할 수 있다. (현재 보안 및 요금 문제로 프라이빗으로 설정) 회원들의 구매 기록, 상품 정보 등 사용할 데이터를(현재 Instacart data) RDS 에 저장했다면 /src/recommendation/run/main.py 를 이용하여 EC2 환경에서 광고 추천 모델을 실행하고 결과를 얻을 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 2.2	2019-MAY-28

5.3 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
스크립트	python	boosting 모델을 생성한다.	main.py 를 실행한다.	모델이 생성되고 예측 값이 나온다.	성공
스크립트	명령어	추천 알고리즘의 성능 평가	고객이 방문했을 때 추천할 제품들이 나열된 목록이 csv 로 저장되어있다. 이를 kaggle 에 전송하여 성능을 평가할 수 있다.	추천 알고리즘의 성능을 평가하고 개선할 수 있다.	성공
스크립트	python, C++ 실행파일	얼굴을 detect 하고 이미지를 Crop 하여 Rekognition 을 통해 고객을 인식한다.	카메라를 설치하고, python 스크립트와 detectnet 실행 파일을 실행한다.	얼굴이 Crop 되는 장면과, 실행 시간을 확인할 수 있다. 최종적으로 광고가 보여진다.	성공
url	url	고객 아이디를 입력으로 받아, 해당 고객에게 추천할 광고에 대한 정보를 획득할 수 있다.	API Gateway deploy url 을 통해 user_id 를 넘겨주면 해당 정보가 웹에 출력되는데, 이것을 통해 정보를 가져왔는지 확인할 수 있다. 오류가 발생하면 오류가 출력된다.	user_id 에 해당하는 정보가 웹에 출력된다. 요청된 정보는 CloudWatch 에서 확인할 수 있다.	성공
스크립트	shell script	회원 얼굴을 촬영하여 데이터셋을 만들고, Rekognition 에 학습시킨다.	카메라가 설치된 상태에서, 회원을 앞에 앉히고 shell script 를 실행한다. 회원 이름과 번호를 입력한다.	생성된 회원 얼굴 데이터셋과 Rekognition 의 학습 결과, DynamoDB 의 등록 결과를 확인할 수 있다.	성공