

DeepFake Detection System

운영 기관명 : 멀티캠퍼스

훈련 과정명 : 딥러닝 기반 AI 엔지니어링

팀 명: CHIMAC

팀 원: 이찬호

이정철

정소현

정용주

황지민

목차

I. 프로젝트 배경

II. 프로젝트 팀 구성 및 역할

III. 프로젝트 수행절차 및 방법

IV. 프로젝트 수행 결과

V. 느낀 점

I. 프로젝트 배경

프로젝트 명

DeepFake Detection
System
(딥페이크 판별 시스템)

프로젝트 목적

악의적인 목적으로 제작
및 배포되는 딥페이크
영상이 심각한 초상권
침해와
인권 침해를 일으키고
있다. 육안으로 구분 할 수
없을 정도로 정교하게
만들어진 딥페이크
영상을 판별하는
시스템을 만들어
범죄 피해를 줄이고자
한다

프로젝트 개요

사이트에 딥페이크로
의심되는 영상을 업로드
하면, 해당 영상이
딥페이크 여부를 확률로
판별하는 기능을
수행하는
시스템 개발

프로젝트 구조

교육 과정 중 배운 CNN을
이용하여 영상의
프레임을 학습시켜
딥페이크 여부를
판별하는 모델 생성

기대 효과

인터넷에 배포 된 영상의
진위 여부를 판단하여,
사용자들의 의구심을
해결하고, 딥페이크
영상으로 인한
피해자들이 없도록 한다

I. 프로젝트 배경



원본 영상

https://www.youtube.com/watch?v=yjahRHch0_c&feature=youtu.be



딥페이크 영상

<https://www.youtube.com/watch?v=MFiw78aVKKE&feature=youtu.be>

II. 프로젝트 팀 구성 및 역할



이찬호 (팀장)

- WBS를 통한 팀 일정 관리
- 딥러닝 모델 생성, 학습, 검증, 예측 구현



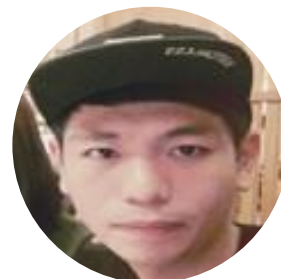
이정철 (팀원)

- 백엔드 개발
- DB 구축 설계 및 제어
- 서버 환경 설정 및 서버 구성 관리



정소현 (팀원)

- 데이터 전처리
- 데이터 샘플링
- 팀 주간 일지 작성



정용주 (팀원)

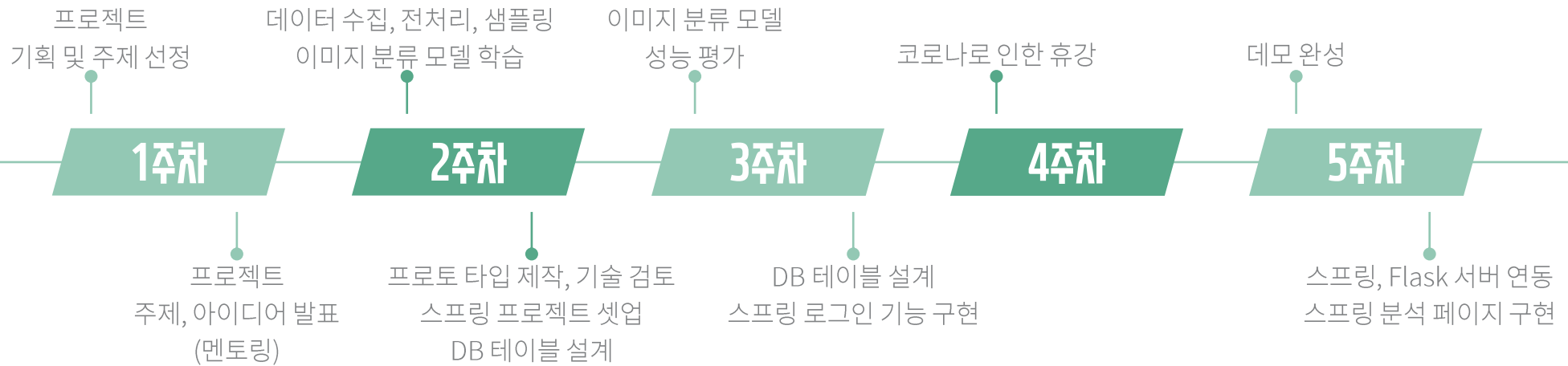
- 프론트엔드 개발
- UI/UX 기획 / 설계
- 웹 검증 및 테스트



황지민 (팀원)

- 딥러닝 서버 백엔드 개발
- 서버 연결
- 스프링 서버 마이페이지 구성
- 인프라 구성 및 배포

III. 프로젝트 수행절차 및 방법



III. 프로젝트 수행절차 및 방법

테스트 케이스 작성
기획서와 매핑
UI 인터랙션 표현

6주차

멘토링 피드백 반영하여
데이터 전처리, 모델
보완 및 개선

멘토링 피드백 반영하여
웹페이지 기능 추가

마이페이지 구현
서비스 UI 코드 구조화

7주차

중간 보고

배포

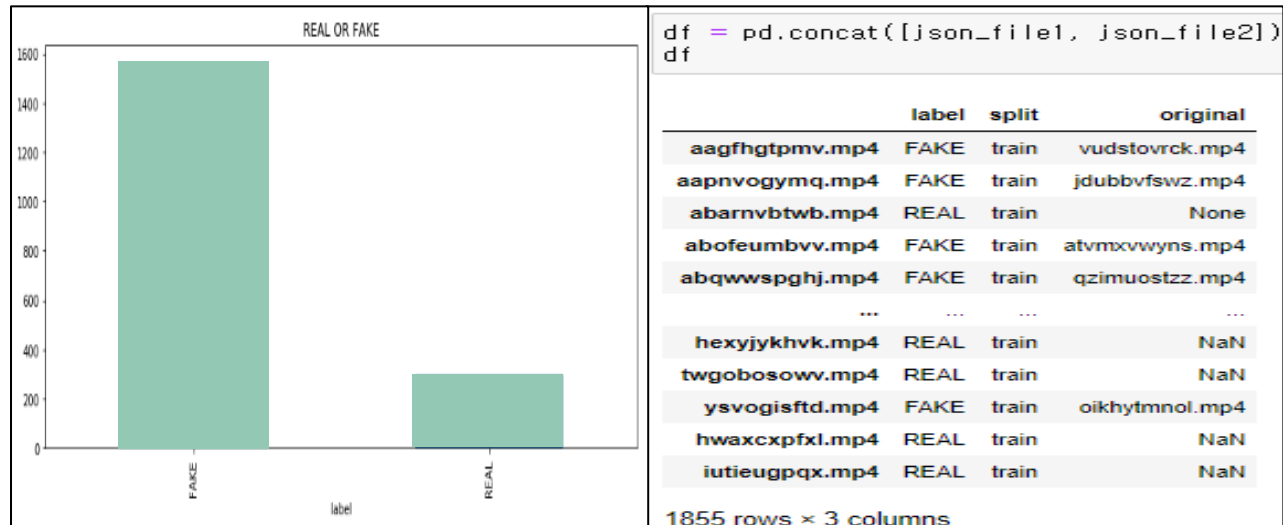
코드 정리

8주차

결과 보고서 작성
발표

IV. 프로젝트 수행 결과

- 탐색적 분석 및 전처리



- Kaggle 대회 'DeepFake Detection Challenge' Dataset 일부 사용
- 26명의 사람, 1855개로 이뤄진 영상 (REAL 296개, FAKE 1559개)

IV. 프로젝트 수행 결과

- 탐색적 분석 및 전처리

```
def chimac(videos, num, label):  
    count=0  
    file = videos.split("/")[-1]  
    fileName = file.split(".")[0]  
    mtcnn = MTCNN(margin=50, keep_all=True, post_process=True,  
                  device='cuda:0', thresholds=[.9, .9, .9])  
    v_cap = cv.VideoCapture(videos)  
    v_len = int(v_cap.get(cv.CAP_PROP_FRAME_COUNT))  
  
    for i in tqdm(range(v_len)):  
        success = v_cap.grab()  
        if i % 1 == 0:  
            success, frame = v_cap.retrieve()  
        else:  
            continue  
        if not success:  
            continue  
        frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)  
        face = mtcnn(frame, save_path = f'C:/ai/workspace/sh/newtrain #  
                                       /{label}/{num}/{fileName}{count}.jpg')  
        count=count+1  
  
    v_cap.release()
```

OpenCV 라이브러리

- 영상을 1개의 프레임 씩 나눠 불러오기
- BGR2RGB를 이용하여 RGB 색상으로 변경

MTCNN 라이브러리

- 얼굴 자동 검출하는 라이브러리
- Margin을 두어 얼굴 일부가 잘리는 것을 방지
- Keep all, Post process를 사용하여 영상에 여러 명이 나올 시 여러 명의 얼굴을 검출
- Device = 'cuda:0'을 사용하여 GPU를 이용하게 함으로 얼굴 검출을 더욱 빠르게 함
- Thresholds를 사용하여 최대한 얼굴인 것만 검출

IV. 프로젝트 수행 결과

- 탐색적 분석 및 전처리

01

```
real_dir = ('C:/ai/workspace/sh/newtrain/REAL/')
real_dir_list = [real_dir + x for x in os.listdir(real_dir)]
real_dir_list = list(map(lambda x: x+"/", real_dir_list))
```

```
real_count=[]
for lists in real_dir_list:
    real_list = [lists + x for x in os.listdir(lists)]
    real_count.append(len(real_list))
```

02

```
fake_dir = ('C:/ai/workspace/sh/newtrain/FAKE/')
fake_dir_list = [fake_dir + x for x in os.listdir(fake_dir)]
fake_dir_list = list(map(lambda x: x+"/", fake_dir_list))
```

```
fake_count=[]
for lists in fake_dir_list:
    fake_list = [lists + x for x in os.listdir(lists)]
    fake_count.append(len(fake_list))
```

03

```
balance_fake = ('C:/ai/workspace/sh/newtrain/fakesamepic/')
balance_fake_list = [balance_fake + x for x in os.listdir(balance_fake)]
balance_fake_list
```

```
np.random.seed(33)
for path in range(len(real_dir_list)):
    real = [real_dir_list[path] + x for x in os.listdir(real_dir_list[path])]
    fake = [fake_dir_list[path] + x for x in os.listdir(fake_dir_list[path])]
    bal_fake = np.random.choice(fake, len(real), replace=False)
    for bal in bal_fake:
        shutil.copy(bal, fakesamplefoldlist[path])
```

04

```
for lists in real_dir_list[:-2]:
    real_list = [lists + x for x in os.listdir(lists)]
    real_list = np.random.choice(real_list, len(real_list), replace=False)
    for real in real_list[:int(len(real_list)*0.11)]:
        shutil.copy(real, 'C:/ai/workspace/sh/newtrain/val/real')
```

```
for lists in fakesamefoldlist[:-2]:
    balance_fake_list = [lists + x for x in os.listdir(lists)]
    balance_fake_list = np.random.choice(balance_fake_list, len(balance_fake_list), replace=False)
    for fake in balance_fake_list[:int(len(balance_fake_list)*0.11)]:
        shutil.copy(fake, 'C:/ai/workspace/sh/newtrain/val/fake')
    for fake in balance_fake_list[int(len(balance_fake_list)*0.11):]:
        shutil.copy(fake, 'C:/ai/workspace/sh/newtrain/train/fake')
```

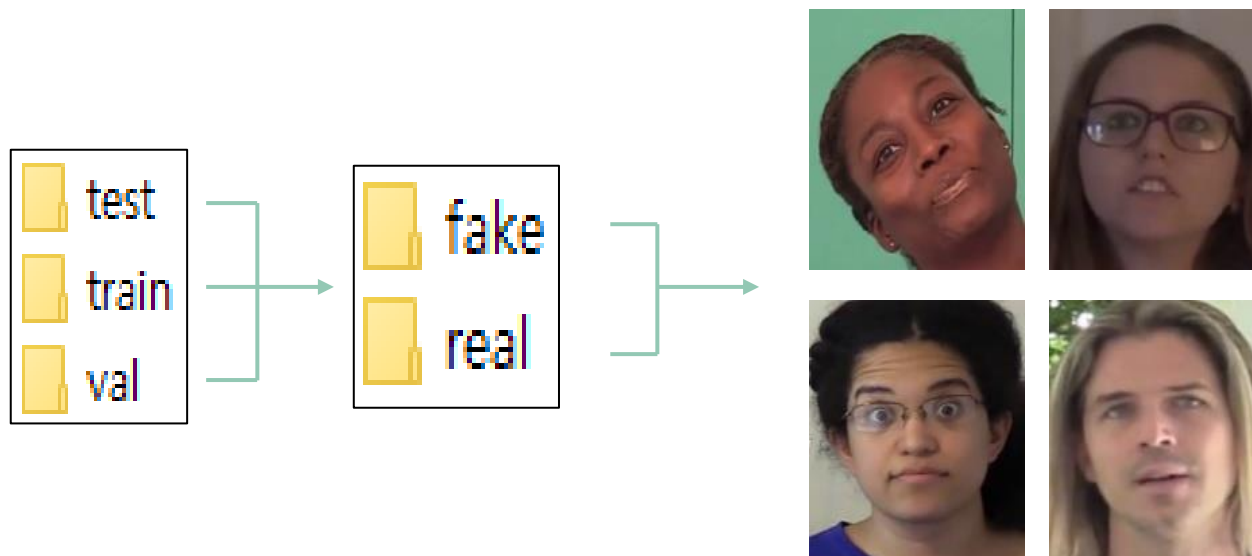
UnderSampling

- Fake 개수가 Real 보다 많아 Overfitting 발생
- Overfitting 방지를 위해
1번 코드로 Real 사진을,
2번 코드로 Fake 사진을 불러 온 뒤
3번 코드로 미리 생성된 폴더에 Fake 사진을 Real 개수에 맞춰 랜덤으로 파일 복사
- 4번 코드로 Fake와 Real 사진을 Train, Validation, Test로 분류

IV. 프로젝트 수행 결과

- 탐색적 분석 및 전처리

UnderSampling의 결과물



IV. 프로젝트 수행 결과

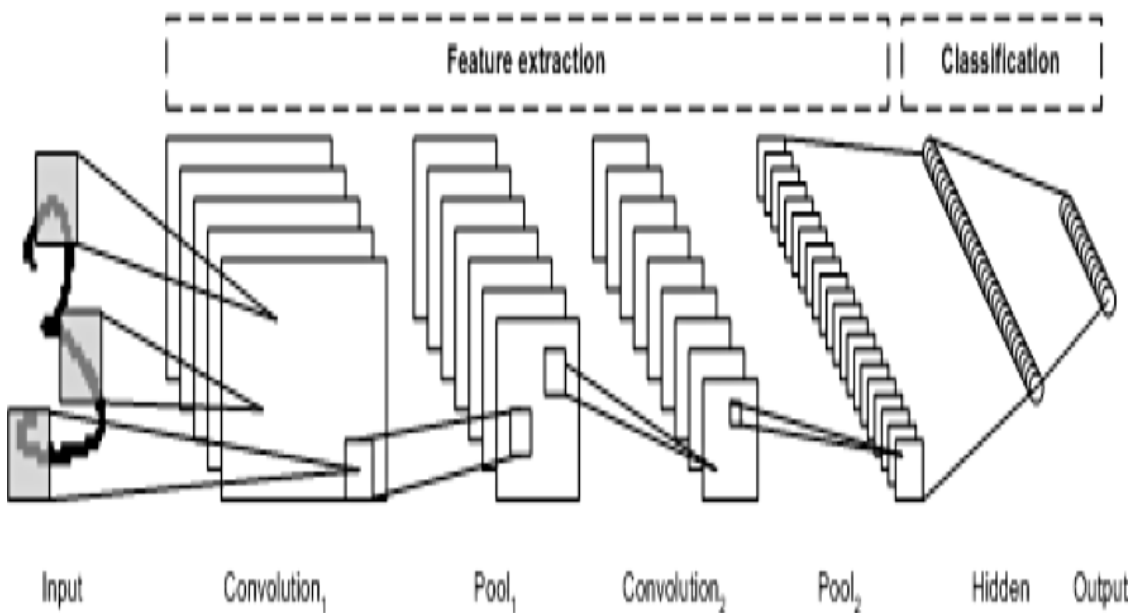
- 모델 개요

CNN (Convolutional Neural Network)

- 이미지를 인식을 위해 패턴을 찾는데 특히 유용
- 데이터에서 직접 학습하고 패턴을 사용해 이미지를 분류
- 특징을 수동으로 추출할 필요가 없음
- 자율주행자동차, 얼굴인식과 같은 객체인식이나 Computer Vision이 필요한 분야에 많이 사용 됨

CNN 주요 용어

- Convolution (합성곱)
- 채널 (Channel)
- 필터 (Filter)
- 커널 (Kernel)
- 스트라이드 (Strid)
- 패딩 (Padding)
- 피쳐 맵 (Feature Map)
- 액티베이션 맵 (Activation Map)
- 풀링 (Pooling) 레이어



IV. 프로젝트 수행 결과

- 모델 개요

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved
Feature

CNN 주요 용어 - 합성곱

- 두 함수 f, g 가운데 하나의 함수를 반전 (Reverse), 전이(Shift)시킨 다음, 다른 하나의 함수와 곱한 결과를 적분하는 것을 의미

IV. 프로젝트 수행 결과

- 모델 개요

RED Channel



Green Channel



Blue Channel

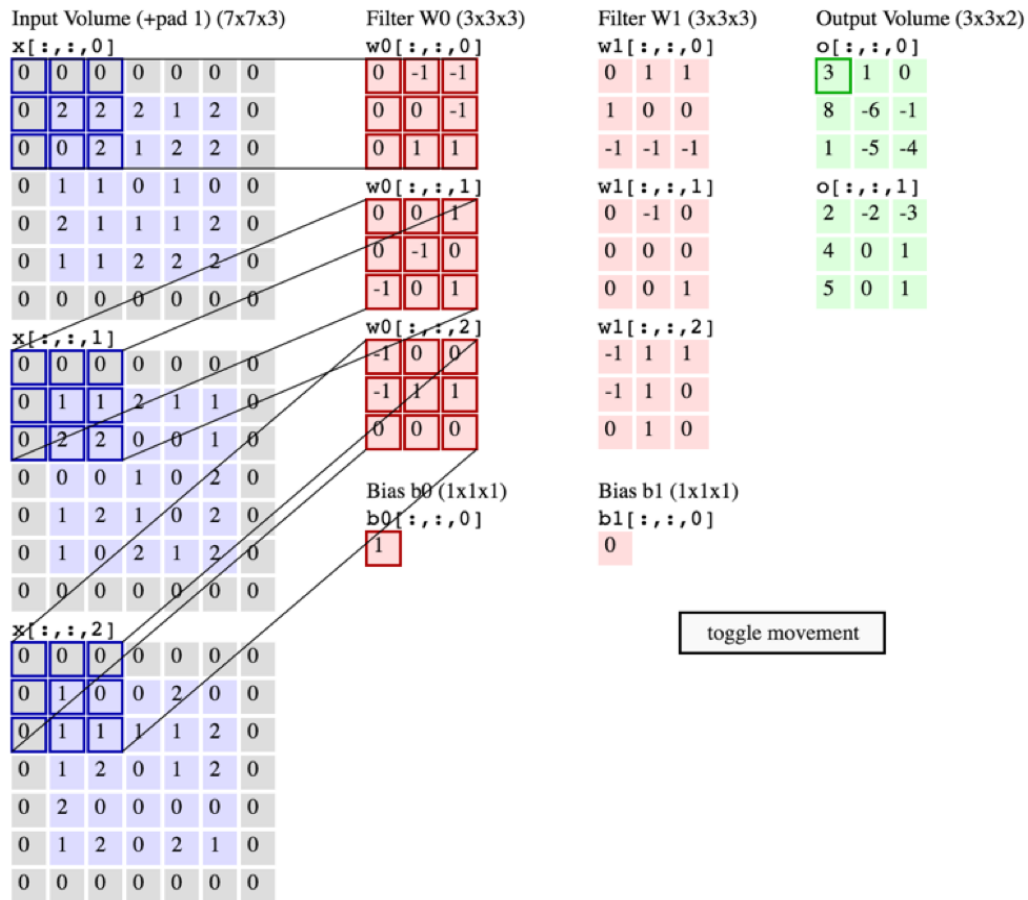


CNN 주요 용어 - 채널

- 이미지 픽셀 하나하나는 실수
- 컬러 사진은 천연색 표현을 위해 각 픽셀을 RGB 3개의 실수로 표현한 3차원 데이터
- 컬러 이미지는 3개의 채널로 구성
- 흑백 명암만을 표현하는 흑백 사진은 2차원 데이터로 1개 채널로 구성

IV. 프로젝트 수행 결과

- 모델 개요



CNN 주요 용어 - 필터

- 이미지의 특징을 찾아내기 위한 공용 파라미터
- Filter를 Kernel이라고 하기도 함
- CNN에서 Filter와 Kernel은 같은 의미

IV. 프로젝트 수행 결과

- 모델 개요

1	2	3	0
4	5	6	0
7	8	9	0
0	0	0	0

CNN 주요 용어 - 패딩

- Convolution 레이어에서 Filter와 Stride에 작용으로 Feature Map 크기는 입력데이터 보다 작음
- Convolution 레이어의 출력 데이터가 줄어드는 것을 방지하는 방법
- 패딩은 입력 데이터의 외각에 지정된 픽셀만큼 특정 값으로 채워 넣는 것을 의미
- 보통 패딩 값으로 0으로 채워 넣음

IV. 프로젝트 수행 결과

- 모델 개요

Activation Map

12	20	30	0
8	12	2	0
34	70	37	7
112	100	22	12

Max Pooling

20	30
112	37

Average Pooling

13	8
79	18

CNN 주요 용어 – Pooling 레이어

- 풀링 레이어는 컨벌루션 레이어의 출력 데이터를 입력으로 받아서 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용
- 풀링 레이어를 처리하는 방법으로는 Max Pooling과 Average Pooling, Min Pooling이 있습니다.

IV. 프로젝트 수행 결과

- 모델 개요

학습에 사용한 모델 소개

모델	사이즈	상위-1 정확성	상위-5 정확성	매개변수	깊이
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
Xception	88 MB	0.790	0.945	22,910,480	126

출처 : <https://keras.io/ko/applications/>

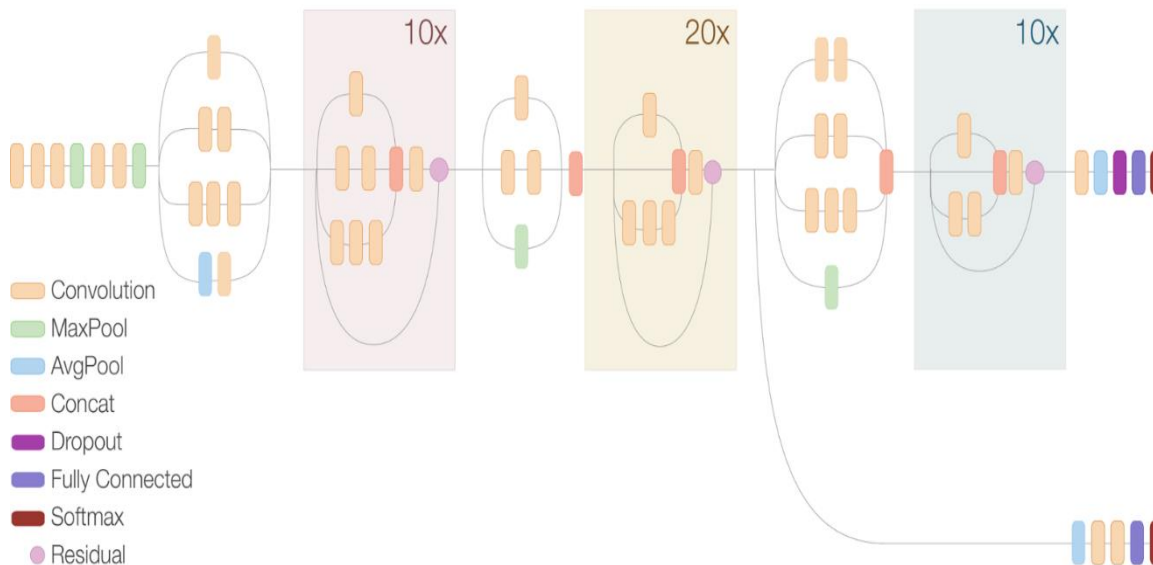
IV. 프로젝트 수행 결과

- 모델 개요

Inception Resnet V2 Network



Compressed View



InceptionResNetV2

- ImageNet 데이터베이스의 1백만 개가 넘는 영상에 대해 훈련된 컨벌루션 신경망
- 164개의 계층이 있으며, 영상을 마우스, 연필, 각종 동물 등 1,000가지 사물 범주로 분류할 수 있음
- 다양한 영상을 대표하는 다양한 특징 학습
- 네트워크의 영상 입력 크기는 299x299입니다.

IV. 프로젝트 수행 결과

- 모델 개요

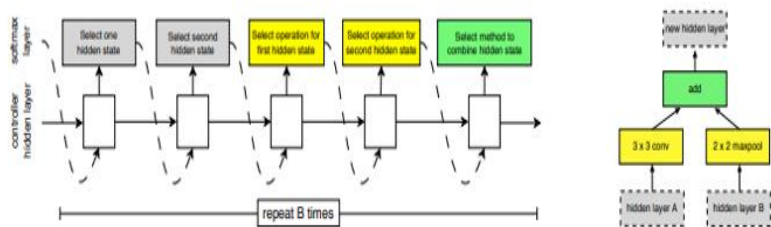
Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$28^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times k$	conv2d 1x1	-	k	-	-

MobileNetV2

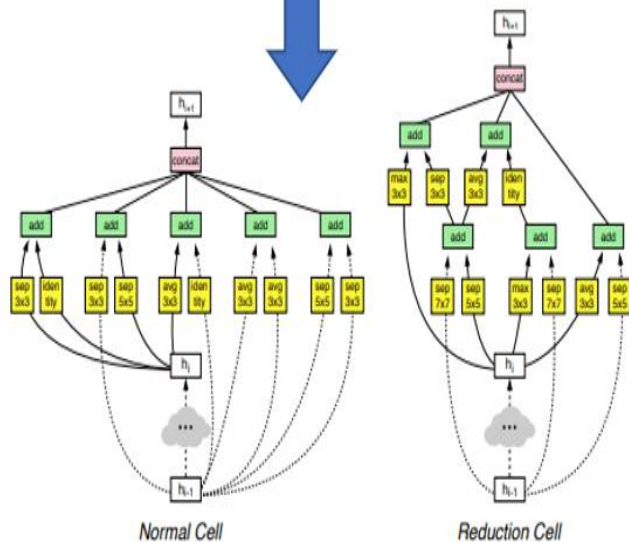
- 모바일이나, 임베디드에서도 실시간을 작동할 수 있게 모델이 경량화 되면서도, 정확도 또한 많이 떨어지지 않음
- 속도와 정확도 사이의 트레이드 오프 문제를 어느정도 해결한 네트워크

IV. 프로젝트 수행 결과

- 모델 개요

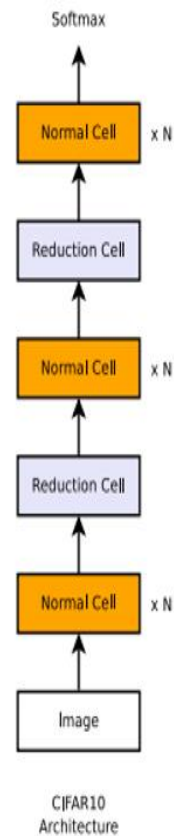


Block



Normal Cell

Reduction Cell



NASNetMobile

- 새로운 검색 공간을 디자인하여 트랜스퍼 할 수 있는 것이 가장 큰 특징
- CIFAR-10 데이터셋에서 최고의 컨볼루션 레이어 (또는 셀)을 찾고, 이 레이어를 ImageNet 데이터셋에 적용하는 방식으로 트랜스퍼 함
- 작은 데이터셋에서 학습한 최고의 결과를 더 큰 데이터셋에 적용
- 네트워크의 영상 입력 크기는 224x224

IV. 프로젝트 수행 결과

- 모델 개요

유형	입력 크기	출력 크기	커널 크기	횟수
입력	(299,299,3)			
Conv	(299,299,3)	(149,149,32)	(3,3)	
Conv	(149,149,32)	(147,147,64)	(3,3)	
Separable Conv	(147,147,64)	(147,147,128)		
Separable Conv	(147,147,128)	(147,147,128)		
Maxpool	(147,147,128)	(74,74,128)		
Add	(74,74,128)	(74,74,128)		
Separable Conv	(74,74,128)	(74,74,256)		
Separable Conv	(74,74,256)	(74,74,256)		
Maxpool	(74,74,256)	(37,37,256)		
Add	((37,37,256)	(37,37,256)		
Separable Conv	(37,37,256)	(37,37,728)		
Separable Conv	(37,37,728)	(37,37,728)		
Maxpool	(37,37,728)	(19,19,728)		
Add	(19,19,728)	(19,19,728)		
Separable × 3 + Add	(19,19,728)	(19,19,728)		× 8
Separable Conv	(19,19,256)	(19,19,728)		
Separable Conv	(19,19,728)	(19,19,1024)		
Maxpool	(19,19,1024)	(10,10,1024)		
Add	(10,10,1024)	(10,10,1024)		
Separable Conv	(10,10,1024)	(10,10,1536)		
Separable Conv	(10,10,1536)	(10,10,2048)		
Average pool	(10,10,1536)	(2048,)		
Dense	(2048,)	(1000)		

Xception

- 2016년에 논문을 통해 발표된 CNN 모델
- Inception에 그 기초를 두고 있음
- InceptionV1(GoogLeNet)에서 Inception module을 이용하여 노드들 간의 연결을 줄이고자 했다는 것을 이전에 학습
- Xception은 거기서 더 나아가 채널 간의 관계를 찾는 것과 이미지의 지역 정보를 찾는 것을 완전히 분리하고자 했다

IV. 프로젝트 수행 결과

- 모델 선정 및 분석

모델 테스트 정확도

(Dropout 80%)

모델	LOSS	ACC	VAL_LOSS	VAL_ACC	TEST_ACC
InceptionResNetV2	0.0589	0.9803	7.3138e-15	0.9777	86.47%
MobileNetV2	0.0887	0.9635	6.1913e-09	0.9539	80.57%
NASNetMobile	0.1267	0.9440	2.3950e-04	0.9475	79.86%
Xception	0.0429	0.9842	9.8779e-11	0.9780	80.49%

IV. 프로젝트 수행 결과

- 모델 선정 및 분석

모델 테스트 정확도 (Dropout 50%)

모델	LOSS	ACC	VAL_LOSS	VAL_ACC	TEST_ACC
InceptionResNetV2	0.0126	0.9954	1.0681e-13	0.9835	89.98%
MobileNetV2	0.0636	0.9733	0.0023	0.9639	79.71%
NASNetMobile	0.1149	0.9492	0.1183	0.9514	79.77%
Xception	0.0807	0.9653	2.6611e-06	0.9646	84.46%

모델 8개 중 Test Acc가 89.98%로 가장 높은 [InceptionResNetV2 \(Dropout 50%\)](#)로 선정

IV. 프로젝트 수행 결과

- 원본 개발 소스(github) 및 배포 URL



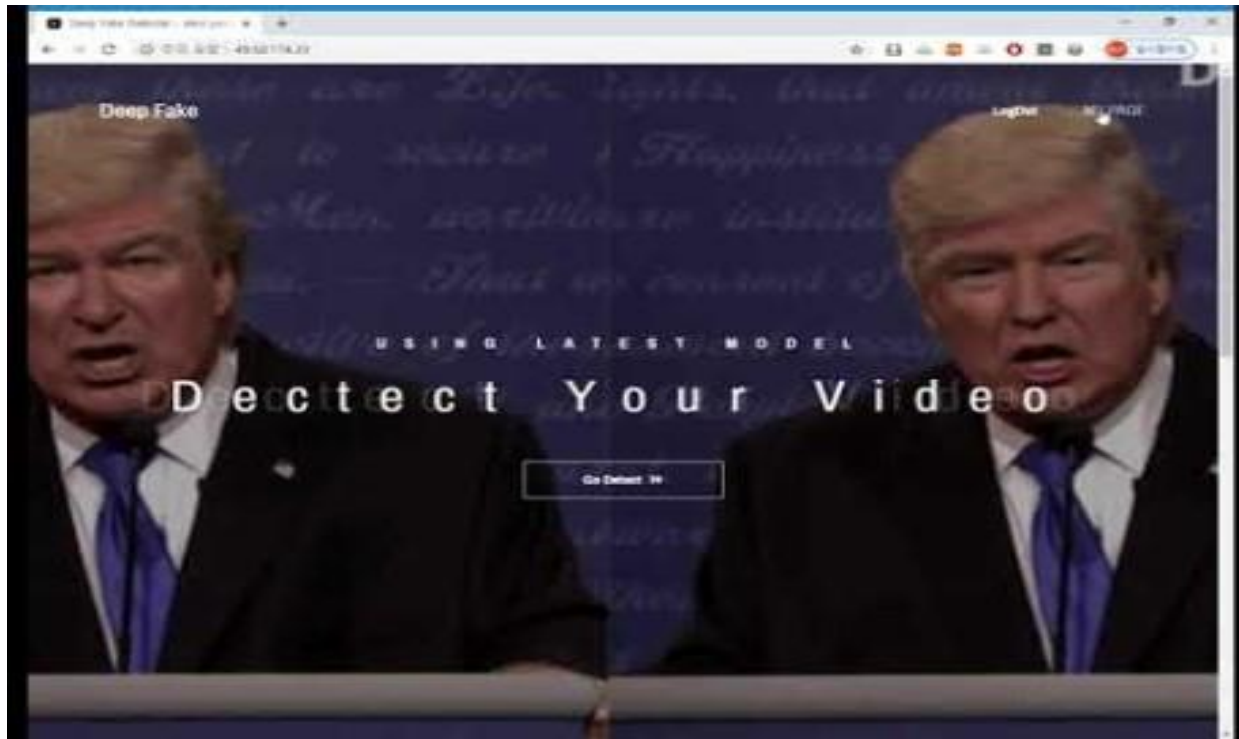
https://github.com/Garrinch95/Final_Project



<http://49.50.174.33/>

IV. 프로젝트 수행 결과

- 시연 영상



<https://www.youtube.com/watch?v=uKsX4jroHeA&t=6s>



이찬호

V. 소감문

프로젝트 수행 상 어려움 극복 사례

얼굴 검출 라이브러리 선정과 Keras에서 지원하는 많은 모델을 선정하는데 많은 어려움을 겪었다. Kaggle 대회인 DeepFake Detect Challenge에선 9시간의 커널 세션 연결 내에 4000개 영상을 예측해야 하는 시간 압박을 벗어 나기 위해 얼굴 검출 성능과 GPU를 가속할 수 있는 MTCNN 라이브러리를 채택하였으며, 많은 모델 중 4가지를 1차 선별 하여 학습을 진행 후 가장 좋은 모델인 InceptionResNetV2 모델을 선정하였다

프로젝트에서 잘한 점

GPU 가속하여 학습을 진행하였을 때 한 Epoch당 40~50분이 소요 됐다. 아마 CPU로 학습을 진행했으면, 더욱 느렸을 것이다. 모델 테스트 정확도 약 90%인 모델을 만든 것이 가장 잘한 점이라고 꼽고 싶다

프로젝트에서 아쉬운 부분

Kaggle 대회 기간 내에 목표로 했던 손실함수(log loss)0.5에 도달하지 못한 0.7까지 밖에 못한 점 그리고 github을 처음 사용하다 보니 미숙한 점이 아쉬웠다.

프로젝트를 통한 진로 설계, 취업분야 탐색 및 결정 등 도움

평소에 데이터 분석 및 관리에 관심이 있어서 이번 프로젝트를 통해 효율적인 데이터 관리 및 분석 방법을 터득 하였고, 이 경험을 살려 데이터 분석, 데이터 관리 업무를 진행을 잘 할 것 같다.



이정철

V. 소감문

프로젝트 수행 상 어려움 극복 사례

AI 엔지니어링 과정을 처음으로 코딩에 접했고 백엔드 부분도 물론 처음이었기 때문에 시작하는 것이 가장 큰 어려움이 있었다. 스프링 프레임 워크를 사용하는 것에 대해서 거부감이 있었다. 그래서 시작이 더욱 힘들었지만, 팀 웹 어플리케이션의 한 부분을 담당하고 있다는 책임감에 교재를 따로 구입하고, 과정 시간 이외에 따로 공부하다 보니 자연스럽게 스프링 프레임 워크에 대한 거부감이 사라지고, 구현하고 싶었던 기능을 구현 했을 때는 큰 성취감을 느끼게 되었다.

프로젝트에서 잘한 점

회원가입 로직을 MVC모델을 이용하여 구현하였고, mysql 과 연동하여 사람들이 등록한 회원정보를 DB 에 저장 될 수 있도록 하였다. 또한 DB유출을 고려하여 비밀번호 정보를 encoding 하여 알 수 관리자 또한 알 수 없는 형태로 저장되게 구현하였다. 로그인 부분 구현 또한 로그인에 인코딩 된 비밀번호 정보를 해독 하여 대조하도록 구현 하였다.

프로젝트에서 아쉬운 부분

협업툴인 깃허브를 온전하게 사용하지 못한 것 이 가장 아쉬웠다. sql의 계정 정보나, navercloudplatform의 개인 키를 분리하지 못했고, 이 때문에 브랜치가 대량으로 생기고, 서로의 버전이 맞지않는등 어려움이 있었다.

프로젝트를 통한 진로 설계, 취업분야 탐색 및 결정 등 도움

내가 하고 싶은 분야가 백엔드 개발자라는 것에 대한 확신이 설 수 있는 시간 이었다. 물론 이번 프로젝트 로 경험한 백엔드 개발이 아주 작은 부분이겠지만 개발을 하며 기능이 구현되었을 때의 기쁨을 알게 되었고, 사용자의 눈에 크게 띄지는 않지만 핵심적인 부분들을 개발하는 것에 큰 흥미를 느끼게 되었다. 이후 취업 분야는 java spring framework 를 이용한 백엔드 개발과 필요하다면 nodejs 나 django 등 백엔드의 전문가가 되기위해 노력 할 것이다.



정소현

V. 소감문

프로젝트 수행 상 어려움 극복 사례

데이터 샘플링 과정에서 무작위로 언더 샘플링을 진행하였는데, 오버 피팅이 일어나 결과가 만족스럽지 못했다. 높은 정확도를 얻기 위해 데이터 전처리 과정에서 영상에 등장하는 사람 별로 나누어 얼굴을 검출하고 언더 샘플링을 진행하였다. 그 결과 오버 피팅이 일어나지 않고 만족스러운 결과를 얻을 수 있었다.

프로젝트에서 잘한 점

데이터 전처리 및 샘플링 과정에서, 계획했던 일을 끝까지 수행해낸 점을 잘한 부분이라 생각한다. 수십만 장의 이미지를 수작업으로 분류해야 했는데, 마우스 클릭 실수로 인해 수행 도중 수십 차례 일을 반복해야 했지만 끝까지 평정심을 잃지 않고 일을 수행하였다.

프로젝트에서 아쉬운 부분

Kaggle 대회에 모델과 예측 코드를 제출했는데, score 값인 log loss값이 0.7으로, 목표했던 점수인 0.5보다 높았던 점이 아쉽다. 당시 Kaggle 마감 시간이 얼마 남지 않았고, 컴퓨터 속도와 용량의 한계로 전체 데이터 셋을 학습 모델로 이용하지 못했던 점이 이유였던 것 같다. 그래서 Kaggle 대회가 끝난 후에도 데이터 전 처리와 학습 데이터 양을 보완하며 좋은 모델을 만들기 위해 노력했다.

프로젝트를 통한 진로 설계, 취업분야 탐색 및 결정 등 도움

멘토님들과 면담 결과, 대학원이 필수는 아니지만 어떤 데이터를 다루고 싶은 지는 명확하게 정해야 한다는 것을 알게 되었다. 졸업 후의 진로에 고민이 많았는데, 다양한 데이터를 분석해보며 어떤 데이터에 가장 흥미가 가는지를 우선적으로 찾고, 그에 대한 전문성을 높여야겠다고 생각했다.



정용주
V. 소감문

프로젝트 수행 상 어려움 극복 사례

이번 프로젝트를 통해 프론트 엔드 개발을 쉽게 봤던 과거를 되돌아보게 되었다. 과거엔 HTML, CSS는 쉽게 보았고 Javascript는 조금만 공부하면 된다고 생각했다. 그러나 공부해도 CSS로 원하는 UI를 구성하는 데 어려움을 느꼈고 Javascript도 처음 배우다 보니 어려움을 느꼈고, jQuery, Ajax 등 JS를 쉽게 사용할 수 있게 해주는 라이브러리가 있었지만, 모든것을 처음 배우다 보니 그것마저 힘들었다. 그래도 다른 팀 프론트 엔드 개발을 맡은 정혁진, 최민창, 김우희 씨의 도움을 받아 포기하지 않고 끝까지 해낼 수 있었다.

프로젝트에서 잘한 점

프로젝트를 수행하며 작성한 테스트 케이스와 기능테스트를 자동화해 놓은 것을 잘한 부분으로 뽑을 수 있다. 요즘 TDD가 급부상하며 개발을 진행하며 유닛 테스트를 작성하며 진행하는데, 그것에 더해 기능테스트를 자동화함으로써 모듈을 수정하거나 function을 바꿨을 때 혹시 모를 문제에 대비할 수 있었고, 리그레션 과정도 금방 마무리 된 것 같다. 다음번에는 프론트 부분만 아니라 백 엔드 개발도 함께 유닛 테스트를 진행하고, 자동화 테스트 부분을 더 증가시킬 수 있다면 더 좋은 프로젝트가 될 수 있을 것 같다.

프로젝트에서 아쉬운 부분

프로젝트를 진행하면서 QA 과정을 정식으로 거치지 않았다는 점이다. 보통 프로젝트를 수행할 때 개발자들에 유닛 테스트에 더불어 단계적 블랙박스 테스트와 화이트 박스 테스트를 거쳐야 하는 것으로 알고 있다. 이 중에 QA 업무에서 대부분 실행하는 블랙박스 테스트 등을 실제로 수행할 수 있었으면 좋았을 것 같다. 또한, QA를 진행하면서 고객의 명세서에 따라 작성하고 실행하는데, 이것을 역할분담을 맡아 명세서와 그것에 맞는 테스트케이스 작성을 해보았으면 더 좋았을 것 같다.

프로젝트를 통한 진로 설계, 취업분야 탐색 및 결정 등 도움

이번 프로젝트를 통해 웹 개발에서 프론트 엔드 개발자와 백 엔드 개발자의 역할 분담에 대한 이해를 증진시킬 수 있었던 것 같다. 예전에는 프론트 엔드랑 사용자에게 입장에 가깝고, 백엔드란 서버의 입장에 가깝다고만 알고 있었다. 그러나 정확한 경계에 대한 이해가 부족했던 것 같다. 어디까지가 일의 경계가 분명하지 않으니 백엔드 개발자들과 일을 떠 미루기 일 수였고, 프로젝트가 끝나감에 따라 조금씩 서로의 역할을 이해할 수 있게 되었다.



황지민

V. 소감문

프로젝트 수행 상 어려움 극복 사례

인프라에 관련된 경험이 없어서 처음에 많은 시행착오를 겪었다. 책도 사고, 개인시간을 투자하여 공부를 하면서 해결하여 뿌듯했다.

프로젝트에서 잘한 점

프로젝트 전반적으로 참여해서 팀원 각자의 고충을 이해할 수 있었다. 팀원들의 의견을 공유하고 반영하여 프로젝트 진행에 도움을 줄 수 있었다.

프로젝트에서 아쉬운 부분

과정 전에 생각하던 AI와 현실을 많이 다름을 알게 되었고, 딥 러닝 알고리즘 개선이 어렵다는 것을 알았다. 많은 시행착오와 충분한 리소스(하드웨어 스펙, 시간)이 필요하다는 것을 배웠다. 그리고 처음에 체계적으로 협업툴 관리를 했었다면 더 빠르게 개발이 가능하지 않았을까 생각이 든다.

프로젝트를 통한 진로 설계, 취업분야 탐색 및 결정 등 도움

클라우드 서버 관리와 인프라설계에 흥미가 생겼다. 기존에 관심있던 임베디드 분야와 유사한 점이 있어서 진로 결정에 좋은 영향을 줬다. 리눅스 커널에 대해 좀더 깊이 공부하고 싶다. 프로젝트는 끝났지만 좀 더 발전시켜서 배포하여 완전히 독립된 서버를 만들고 싶다.

V. 느낀 점



이찬호 (팀장)

열악한 자원으로 프로젝트를 진행해 아쉽지만, 메모리 관리 및 여러가지를 배웠고 CNN 모델에 대해 많이 공부를 해서 데이터 분석에 진로 방향을 정하는데 도움이 됨



이정철 (팀원)

백엔드의 모든 부분이 처음이었기 때문에 시작하는 것이 가장 큰 어려움이었지만, 수업 외에 개별적으로 공부하며 제작해 나가니 성취감을 느낄 수 있었다. 아직 부족한 점이 많지만 추후 지속적인 공부를 통해서 원하는 것을 무엇이든 만들어낼 수 있는 백엔드 전문가가 될 수 있게 노력할 것이다.



정소현 (팀원)

시간과 자원의 한계로 아쉬운 점도 존재하지만 이번 프로젝트가 데이터 공부와 진로설계에 큰 도움이 되었다.



정용주 (팀원)

이번 프로젝트를 진행하면서 백엔드와 프론트 엔드 개발의 구분점을 알 수 있었고, QA를 수행한 것은 잘한 것 같으나 QA를 명세서를 가지고 전문적으로 하지 못한 점은 아쉬웠던 것 같다. 추후 웹 개발 직무나 QA직무에 지원 할 때 큰 도움이 될 것 같다.



황지민 (팀원)

텐서플로우 버전과 GPU 사용여부에 따라 서버실행이 안 될 때가 있었다. 도커라는 가상환경을 구성해서 시행착오를 빠르게 진행했다. 또한 프로젝트의 전반적인 부분에 참여할 수 있어서 좋은 경험이었다.