

ISSN 2765-5482



# 데이터과학연구

## Journal of Data Science

제11권



2022년 9월

데이터과학연구소

The Research Center for Data Science



ISSN 2765-5482

# 데이터과학연구

Journal of Data Science

---

제11권

---

---

2022년 9월

---

데이터과학연구소

The Research Center for Data Science

## **데이터과학연구소 임원 및 연구부**

- 소장 : 곽일엽 교수 (응용통계학과, ikwak2@cau.ac.kr)
- 간사 : 이주영 교수 (응용통계학과, jooylee@cau.ac.kr)
- 연구부

박상규 교수 (응용통계학과)  
김삼용 교수 (응용통계학과)  
이재현 교수 (응용통계학과)  
성병찬 교수 (응용통계학과)  
박재현 교수 (컴퓨터공학과)  
임창원 교수 (응용통계학과)  
김원국 교수 (응용통계학과)  
황범석 교수 (응용통계학과)  
임예지 교수 (응용통계학과)  
이재우 교수 (산업보안학과)

## **데이터과학연구 편집위원회**

편집위원장 : 곽일엽 교수 (응용통계학과)  
편집 위원 : 이주영 교수 (응용통계학과)

# Journal of Data Science

---

2022. 9

---

Vol.1

---

## Contents

- ◆ Deep Learning Based Growing Environment Optimization .....  
.....이정국, 임석원, 황채원, 김현준 / 1
- ◆ Classification of Images for Makerere Fall Armyworm Crop and Applying  
Grad-CAM .....강지원, 손동현, 이승현, 이재용 / 17
- ◆ Code Similarity Determination Using SBERT .....  
.....김희민, 노지현, 황재원 / 32
- ◆ Comparison of Algorithm Performance for Handling Data Imbalance  
.....김병찬, 김희원, 최지은, 차호영 / 44
- ◆ Leaf area forecasting Using CNN .....  
.....임성호, 류재욱, 이경서, 강진모 / 65
- ◆ US Patent Phrase to Phrase Matching Using DeBERTa V2 Model  
.....최주원, 박지원, 지평진, 강태인 / 75
- ◆ Imbalanced Anomaly Detection Using MVtec Data .....  
.....구지윤, 김민경, 이수미, 이호준 / 85
- ◆ Factors Affecting Social Media Usage using Propensity Score Matching  
and Survival Analysis .....강진모, 김희원 / 97

---

The Research Center for Data Science

---

## Deep Learning-based Growing Environment Optimization

이정국<sup>1)</sup>, 임석원<sup>2)</sup>, 황채원<sup>1)</sup>, 김현준<sup>1)</sup>

### Abstract

The fourth industrial revolution is a digital revolution that has occurred since the mid-20th century due to the convergence of information and communication technologies. Technological innovations are emerging in various fields, such as artificial intelligence (AI), genetic modification, robots, and augmented reality. These technologies have been applied to multiple industries and have increased agricultural productivity. Efficient crop cultivation has been achieved through information technology, such as increased agricultural research to respond to climate change and smart farms in agriculture following the 4th industrial revolution. In this study, the leaf area prediction algorithm is developed using the image classification deep learning model of the bok choy image, and an AI model is implemented for deriving the optimal growth environment. Based on AI, it is possible to derive the optimal environment for the efficient growth of crops. It aims to promote smart agriculture and contribute to stabilization by establishing growth environment information data. A total of five models were used: Each ResNet, Xception, VGG16, MobileNetV2 and four ensemble models, and the performance of all models were checked and compared. This study was submitted using data from the Optimization of Growing Environment Contest hosted by DACON.

- 
- 1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea
  - 2) All authors have equal contribution, Department of Animal Science and Technology, Chung-Ang University, Anseong-si, Gyeonggi-do 17546, Korea

## 1. 서론

4차 산업혁명은 정보통신 기술의 융합으로 20세기 중반부터 이뤄지고 있는 디지털 혁명이다(Xu et al., 2018). 4차 산업혁명은 Artificial Intelligence(AI), 유전자 변형, 로봇, 증강현실 등 다양한 분야에서 새로운 기술 혁신이 나타나고 있다. 이러한 기술들은 다양한 산업에 적용되었고 대표적으로 농업 생산성 증대에 기여하였다. 기후변화에 대응하기 위한 농업분야의 연구 증가와 4차 산업혁명에 따른 농업 분야의 스마트팜 등 정보 기술 활용을 통한 효율적인 작물재배가 이루어졌다.

본 연구에서는 데이콘(DACON)에서 주최하는 ‘생육환경 최적화 경진대회’에 참여하여 대회에서 주어진 청경채 이미지를 이미지 분류 딥러닝 모델을 활용하여 일면적 예측 알고리즘을 개발하고 최적의 생육 환경 도출을 위한 AI모델을 구현한다. AI를 기반으로 작물의 효율적인 생육을 위한 최적의 환경을 도출할 수 있으며 생육환경 정보 데이터 구축을 통한 스마트 농업 활성화 및 안정화 기여를 목표로 한다. 각각의 ResNet, Xception, VGG16, MobileNetV2 모델과 네 가지 모델을 ensemble한 총 5가지 모델이 사용되었고 모든 모델의 성능을 확인하고 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구에서 사용된 각 모델들에 대해서 설명한다. 3장에서는 대회에서 주어진 데이터에 대해 설명하고 모델이 어떻게 구현되었는지에 대해 설명한다. 마지막으로 4장에서는 실험의 결과와 한계점에 대해 논의한다.

## 2. 모형 설명

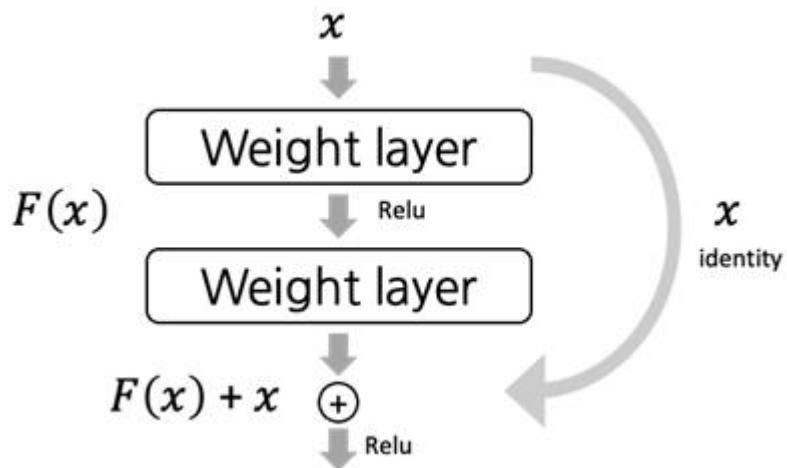
### 2.1. ResNet

ResNet 모델(He 등, 2016)이 개발되기 전에는 많은 레이어를 바탕으로 성능이 좋은 모델을 선정하는 추세였다. 하지만 레이어가 깊어질수록 결과값을 도출하는 과정에서 곱해지는 미분값의 크기가 작아지면서 입력값에 더해지는 중요도가 작아지는 문제가 생겼는데, 이를 Vanishing Gradient Problem이라고 한다. 이러한 문제를 해결하기 위해 소개된 모델이 ResNet 모델이다(He et al., 2016).

기존의 Neural Network에서는 input(x)을 output(y)으로 mapping하는  $H(x)$ 를 찾았다면, Residual Network는 잔차  $H(x) - x$ 를 최소화하는 방향으로 모델을 구

현한다. 그림[1]에서  $F(x) = H(x) - x$  라고 하였을 때,  $F(x)$ 를 residual learning을 통해 학습시킨 후 identity mapping으로 산출된  $x$ 를 더해주면 output으로 똑같이  $H(x)$ 를 얻게 된다. 이때 모든 layer에서의 gradient는  $F'(x) + 1$ 이기 때문에 최소값으로 1을 갖게 되어 Gradient Vanishing Problem을 해결할 수 있다.

[그림 1] Residual Block(BottleNeck Architecture)



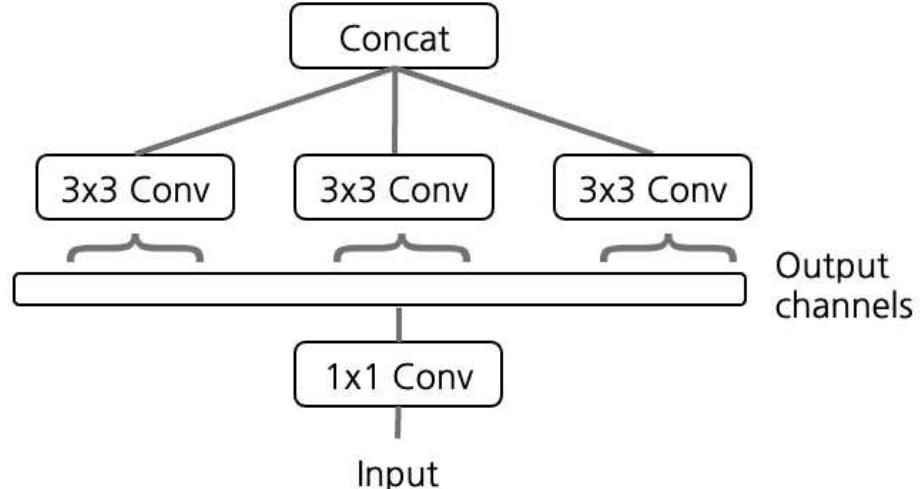
## 2.2. Xception

Inception은 이미지 파일의 너비와 높이인 spatial dimension과 channel dimension을 3개의 1X1 convolution으로 적당히 분리하여 channel correlation을 먼저 구한 후 3X3 혹은 5X5 의 일반적인 convolution으로 spatial correlation에 대해 연산을 진행하고 concat으로 합쳐서 결과값을 구하는 기법이다.

Xception (Chollet et al., 2017)은 Extreme Inception의 약자로 기존 Inception 기법을 강화한 모델이다. Inception에서는 spatial convolution과 channel convolution을 3개의 segment로 나눴다면, Xception에서는 모든 spatial과 channel correlation을 depthwise separable convolution으로 완전히 분리하여 계산한다. Depthwise separable convolution의 과정은 [그림 3]처럼 input channel 각각에 대하여 spatial convolution인 depthwise convolution을 진행한 후 1X1

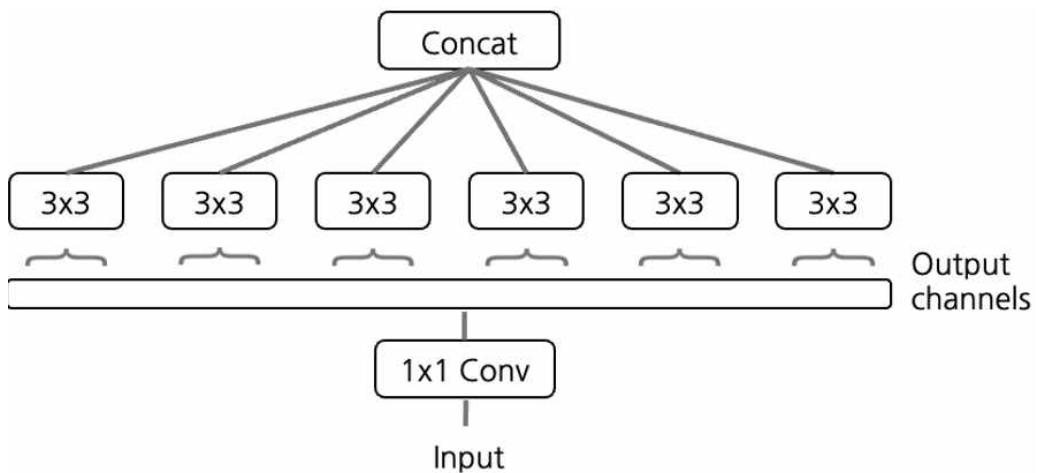
convolution을 통해 앞선 과정에서의 결과값을 새로운 channel space로 할당한다.

[그림 2] simplified Inception model

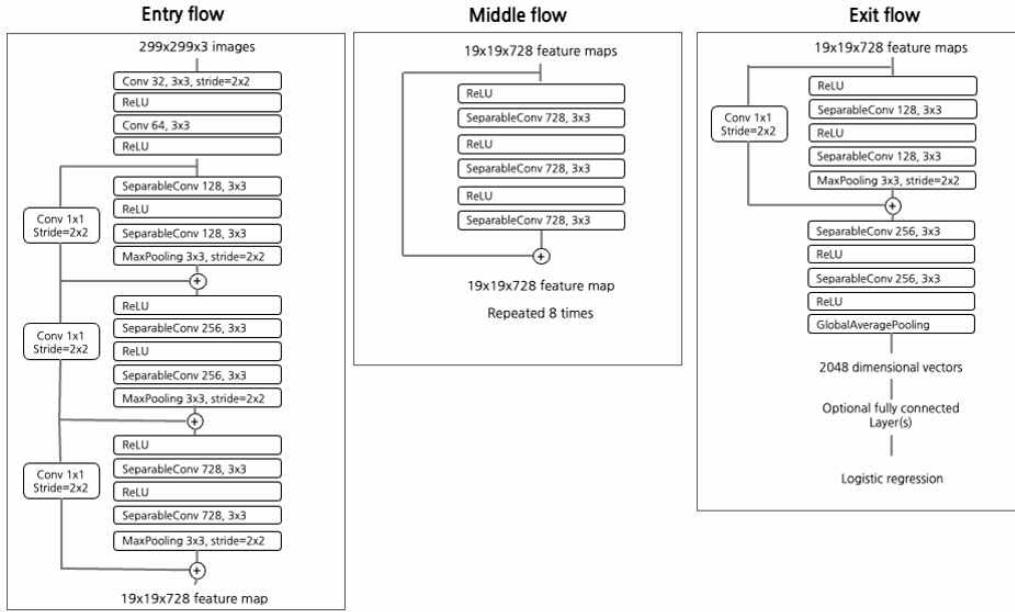


전체 Xception 모델은 이러한 block을 여러개 쌓아 완성이 되며 전체 모형은 [그림 4]와 같다.

[그림 3] Simplified Xception model



[그림 4] Structure of Xception model



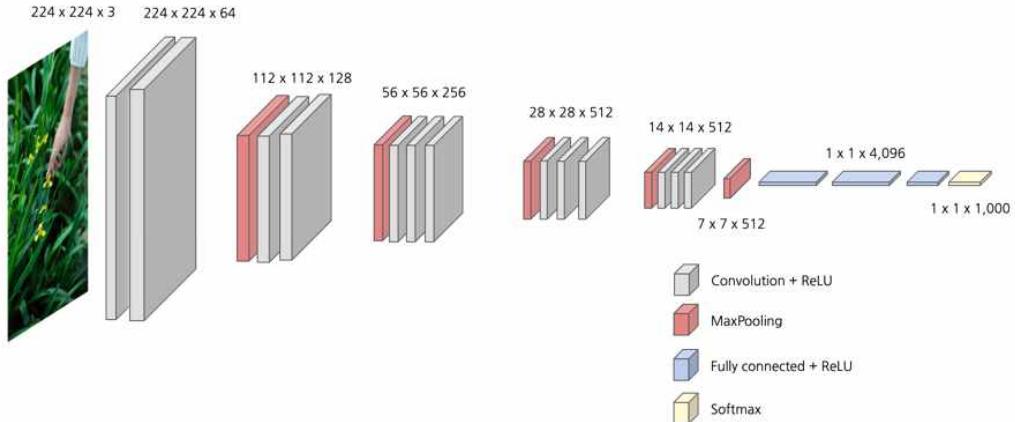
### 2.3. VGG16

VGGNet 모델(Simonyan & Zisserman, 2014)은 네트워크의 깊이가 모델의 성능에 미치는 영향을 연구하고자 개발되었다. VGGNet 모델에서는 convolution 필터 사이즈를 상/하/좌/우/중앙을 확인할 수 있는 필터 중 가장 작은 사이즈인 3 X 3으로 통일하는데, 이는 이미지의 크기가 줄어드는 속도를 조절 함으로써 네트워크의 깊이를 늘리는데 용이하게 한다. 또한 큰 사이즈의 필터를 사용하는 것보다 가중치의 개수가 줄게 되어 모델의 학습 속도를 향상시킨다. [표 1]은 ConvNet의 구성을 보여주는 표인데 이중 칼럼 D가 VGG16에 해당한다. VGG16은 16개의 레이어로 구성되어 있으며, 각 층에서는 서로 다른 크기의 필터로 특성맵을 convolution 해준다.

[표 1] Configuration of ConvNet

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input(224x224 RGB Image)					
Conv3-64	Conv3-64 LRN	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64
MaxPooling					
Conv3-128	Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128
MaxPooling					
Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv1-256	Conv3-256 Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv3-256 Conv3-256
MaxPooling					
Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv1-512	Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv3-512 Conv3-512
MaxPooling					
FC-4096					
FC-4096					
FC-1000					
Soft-Max					

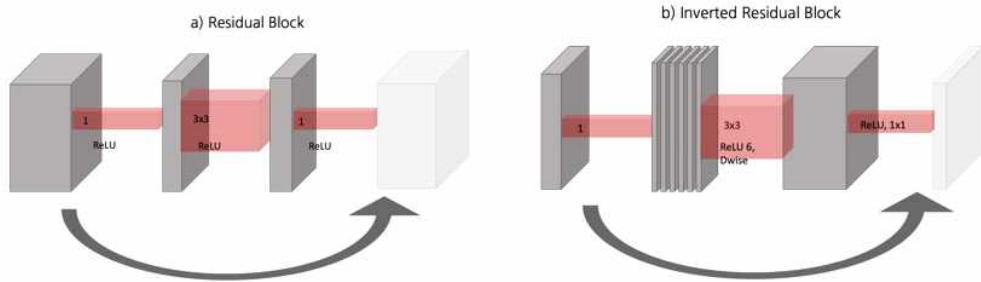
[그림 5] Structure of VGG16



## 2.4. MobileNet V2

MobileNet V2 모델(Sandler 등, 2018)은 기존 MobileNet 모델을 linear bottleneck과 inverted residual을 도입하여 성능을 향상시킨 모델이다. Linear bottleneck이란 만약 manifold of interest가 RELU 변환 이후 0이아닌 volume으로 남아 있다면, 이것을 선형변환에 부합한다고 보고 convolutional blocks 에 linear bottleneck 레이어를 추가함으로써 비선형 변환으로 인한 정보 손실을 예방하는 기법이다. 이 과정에는 input manifold가 input space의 저차원의 subspace에 있을 때 RELU 변환이 input manifold의 완전한 정보를 보존할 수 있기 때문이다. 여기서 이 저차원의 subspace에는 입력값의 중요한 정보들이 모두 들어있다고 보기 때문에 [그림 6]과 같이 inverted residual 을 통해 이 정보들을 network의 깊은 곳까지 전달해 준다.

[그림 6] Residual Block and Inverted Residual Block



MobileNet V2의 구조는 [표 2]에서 볼 수 있듯이 32개의 필터를 갖는 fully convolutional 레이어와 19개의 residual bottleneck 레이어로 이루어진다.

[표 2] Structure of MobileNet V2

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
$14^2 \times 64$	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	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

### 3. 데이터 구성 및 모델 적용

#### 3.1. 데이터 구성 및 전처리

[표 3] Summary of train, validation and test set

	Train set	Validation set	Test set
Image ( $N$ )	900	373	460
Label	Leaf weight	Leaf weight	-

[표 3]은 데이터 구성 요약표이다. Train set은 900장, validation set은 373장으로 random shuffle되었고 test set은 460장이 존재한다. 이미지에 해당하는 label은 일무게로, train과 validation set에서 존재한다.

다양한 크기를 가지는 이미지를 feature로 사용하기 위해 동일한 차원으로 사이즈를 변경해야 한다(Salimans et al., 2016). 이미지 크기의 표준화를 통해 메모리 사용량을 줄이며 네트워크 모델 사이즈와 동일하게 맞춰주기 위해 전처리 과정을 진행하였다. 데이터 이미지 사이즈 변경 및 표준화 과정을 진행하기 위해 모든 데이터의 이미지 사이즈를 224 x 224로 고정시켰다.

[그림 7] Image data augmentation

(a) Raw image



(b) Flipping image



(c) Rotation image

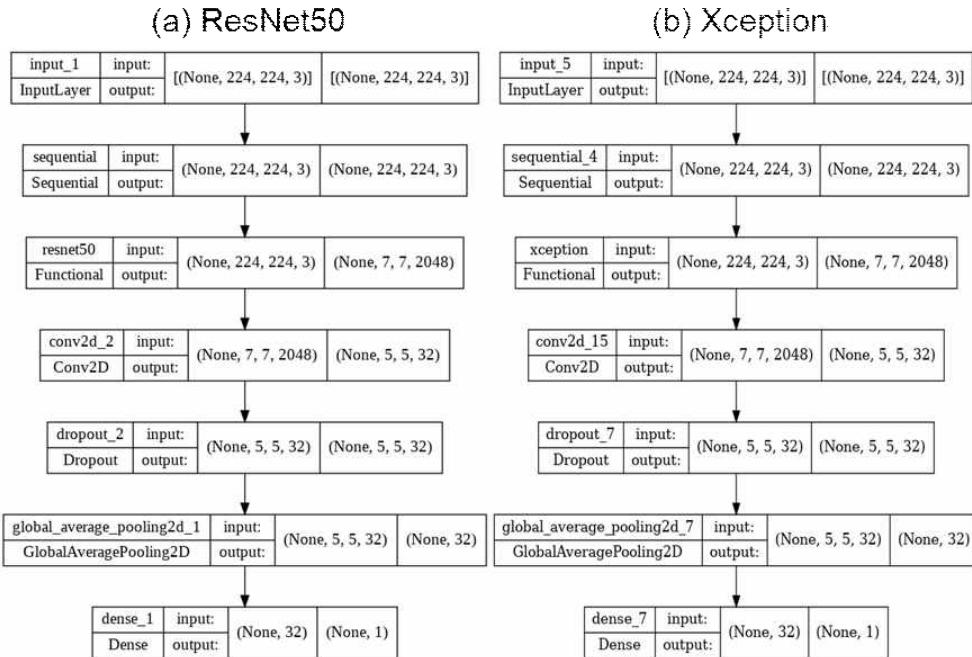


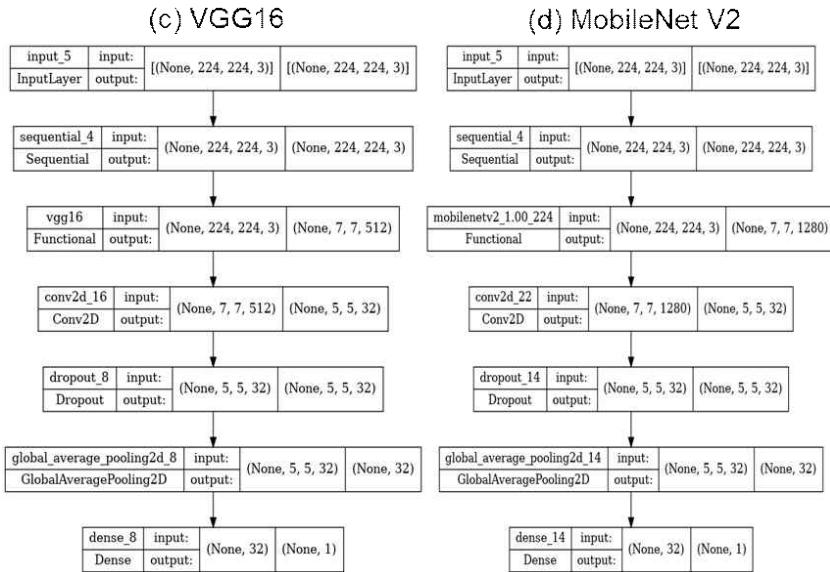
데이터의 수가 많지 않을 때 CNN을 통한 모형 학습이 어려울 수 있다. 이를 보완하기 위해 이미지의 색깔, 각도 등을 변형하여 데이터의 수를 늘려주는 데이터 Augmentation을 활용할 수 있다(Perez et al., 2017). [그림 7]과 같이 데이터는 수평 및 수직 방향으로 이미지를 뒤집어주는 randomflip과 이미지의 각도를 변환시켜주는 randomrotation 기법의 Augmentation을 활용하여 전처리 과정을 진행하였다.

### 3.2. 모델 적용

학습이 수행된 모델의 속도 및 성능 향상을 위해 이미 학습된 모델의 가중치를 미세하게 조정하여 학습시키는 Transfer learning 과 Fine tuning 방법을 활용하였다(Bengio and Yoshua, 2012; Too et al., 2019).

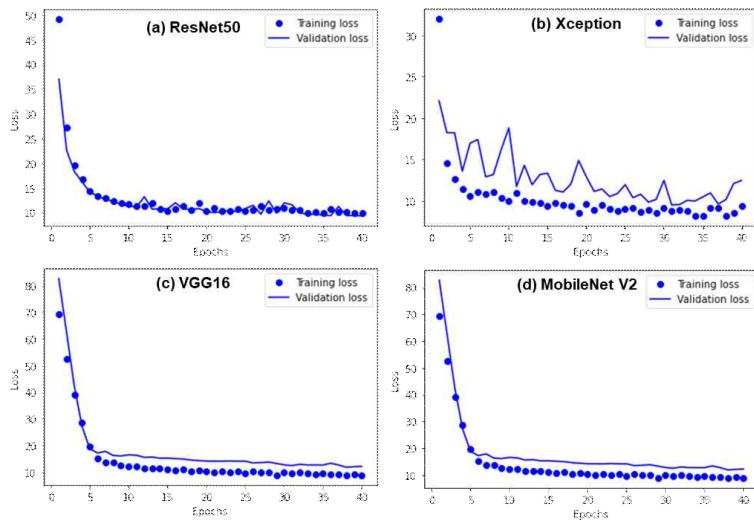
[그림 8] Hyperparameter of pre-trained Models





Pre-trained 단계에서는 ResNet50, Xception, VGG16, MobileNet V2 4가지 모델 [그림 8]과 Ensemble 모델이 사용되었다.

[그림 9] Training and validation loss



[표 4] Summary of total and trainable parameters by each model

	ResNet50	Xception	VGG16	MobileNetV2
Total parameters	24,177,601	21,451,369	14,862,209	2,626,689
Trainable parameters	589,889	589,889	147,521	368,705

[그림 9]는 Pre-trained 단계에서 사용된 4가지 모델의 train 및 validation loss를 보여준다. 각 모델별 사용된 total parameter와 trainable parameter는 [표 4]에 나타난다.

#### 4. 실험 결과

[그림 10] NMAE(Normalized Mean Absolute Error)

$$\text{NMAE} = \frac{\text{MAE}}{\text{ABSOLUTE OF TRUE}}$$

실험 결과의 성능 평가 지표는 측정된 데이터의 평균으로 정규화를 적용하여 척도가 다른 데이터 세트의 Mean Absolute Error(MAE)를 비교하는 Normalized Mean Absolute Error(NMAE)이다(Qi et al., 2020).

[표 5] Summary of model results

Model	Validation_NMAE	Test_set_MAE	Submission_NMAE
ResNet50	12.13	12.74	0.1477

Xception	11.24	11.47	0.2467
VGG16	15.63	13.33	0.259
MobileNet V2	12.09	13.63	0.2702
Ensemble	-	-	0.2476

실험에서 사용한 모든 모형에 대한 결과값은 [표 5]와 같다. MobileNetV2 모델은 ShuffleNet, NasNet-A 등과 같은 경량화 모델 사이에서는 성능이 좋은 것으로 알려졌지만, 연산량이 적다는 특징과 실험에서 사용한 trainable parameter 수가 적은 문제 때문에 사용된 다른 모델들에 비해 성능이 낮게 나왔다. VGG16모델 역시 trainable parameter 수가 타 모델보다 세 배 이상 적었기 때문에 MobileNet V2모델에 비해 크게 개선되지 않았다. Xception 모델은 validation set과 test set에서의 MAE 값은 낮았지만 대회에 제출 후 평가 점수를 내는 실제 test set에 적용했을 때 ResNet50 모델이 크게 개선된 점수를 기록했다.

## 5. 결론 및 개선방안

대회 참여기간 동안 총 57번의 결과를 제출했고, 최종 스코어 0.1477로 총 136 팀 중에 17위를 기록하고 대회를 마무리했다.

이번 대회에서 제공해주는 데이터 중에 청경채가 자란 환경의 메타데이터가 있었는데, 이를 활용할 방법을 찾지 못해 메타데이터를 활용하지 않고 실험을 진행했다. 대회 종료 후 일부 공개된 상위권 참가자들의 코드를 보았는데 임의의 기준에 따라 다른 값들과 차이가 크게 나는 이상치 메타값들을 분류해서 고의로 누락시키거나, 메타데이터는 그대로 유지한 채 청경채 성장 케이스를 지우는 등 실험을 진행하면서 생각하지 못했던 여러 방법론을 접하게 되었다. 메타데이터를 활용하더라도 상태에 따라서 과적합이 발생해 사용하지 않는 것이 오히려 나을 수 있다는 점

도 배울 수 있었다. 딥러닝 프로젝트 경험이 적어 모델 사용방법에 익숙해지는데 시간이 들어 다양한 모델과 방법론을 사용하지 못했다는 아쉬움이 남았지만, 강의에서 배운 지식을 실제 프로젝트에서 어떻게 활용하는지 배울 수 있는 경험이 됐다.

## 참고문헌

- [1] Xu, M., David, J. M., & Kim, S. H. (2018). The fourth industrial revolution: Opportunities and challenges. *International Journal of Financial Research*, 9(2), 90–95.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- [3] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1251–1258).
- [4] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [5] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 4510–4520).
- [6] Salimans, T., & Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*, 29.
- [7] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- [8] Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (pp. 17–36). JMLR Workshop and Conference Proceedings.
- [9] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272–279.

- [10] Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C. H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, 27, 1485–1489.

데이터 과학 연구 제11권

2022년 pp.17-31

## Classification of Images for Makerere Fall Armyworm Crop and Applying Grad-CAM

강지원<sup>1)</sup>, 손동현<sup>1)</sup>, 이승현<sup>1)</sup>, 이재용<sup>1)</sup>

### Abstract

Fall armyworm is a devastating pest in Africa and is causing a massive loss of maize production annually. Since maize accounts for most of the grown crop in Africa and is a staple for many people around the world, this problem is in urgent need of solution. In this work, we approached this problem as a binary image classification. We classified if a plant has been affected by the pest with its leaf's image. Two pre-trained models, ResNet-50 and EfficientNet-B0 were used for transfer learning. They achieved AUC score of 0.9843 and 0.9981 respectively. We also examined which part of the image the models focused on by the heatmaps obtained by the Grad-CAM technique. EfficientNet-B0 performed better in terms of both AUC score and the interpretability of the heatmaps.

---

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

Fall armyworm은 아프리카에서 연평균 31%의 옥수수 손실을 야기하는 파괴적인 해충이다. 옥수수는 벼, 밀과 더불어 세계 3대 식량작물 중 하나로 벼나 밀과는 달리 재배된 역사는 비교적 짧지만, 다양한 쓰임새로 인해 세계 경제에 미치는 파급력이 매우 큰 작물이다(백성범, 2011). 세계의 옥수수 재배면적은 약 1억 5,900만 ha, 생산량은 8억 1,900만 톤이며 아프리카에서도 가장 널리 재배되는 작물이다. 옥수수는 가치 있는 부산물과 높은 생산량으로 지속적인 식량 위기를 겪고 있는 아프리카 국가들의 주요 식량이지만, Fall armyworm과 같은 해충에 취약하기 때문에 이러한 특정 작물에 대한 의존은 식량 위기와 빈곤의 주요 원인이 된다. 따라서 아프리카의 농작물 손실을 예방하기 위한 방법의 설계가 식량 위기와 빈곤 문제를 해결할 수 있는 측면에서 중요하다.

본 연구에서는 ZINDI에서 주최하는 ‘Makerere Fall Armyworm Crop Challenge’에 참여하여 작물의 이미지를 통해 Fall armyworm에 의한 손상 여부를 예측하는 딥러닝 모델을 구현한다. ‘Makerere Fall Armyworm Crop Challenge’는 이진 분류 과제로 Fall armyworm이 작물을 횡폐화시키기 전 작물의 손상 여부를 진단하여 문제에 개입하고 손실을 예방하는 데 목적이 있다. 이미지 이진 분류를 위해 사용된 모델은 ResNet-50와 EfficientNet-B0이다. 두 모델의 성능은 AUC score로 평가를 한다. 또한, 두 모델의 성능 차이를 더 자세히 분석하기 위해 Grad-CAM(Gradient-Weighted Class Activation Mapping)을 사용한다. 동일한 조건에서 학습된 두 모델의 히트맵을 구한 후 각 모델이 잘못 분류하였을 때의 히트맵과 오르게 분류하였을 때의 히트맵을 비교하고 이를 통해 두 모델이 이진 분류를 할 때 이미지의 어느 부분에 초점을 두는지 차이를 파악하고자 한다.

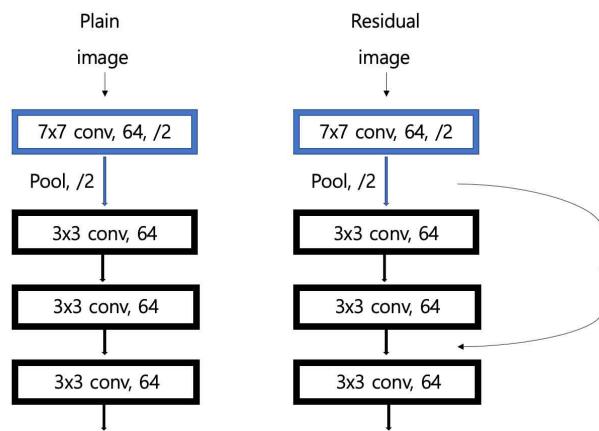
## 2. 이론적 배경

### 2.1. ResNet

ResNet (Residual Network)은 CNN(Convolutional Neural Network) 모델 중

하나이다. 기존 모형들은 레이어(Layer)가 깊을수록 Vanishing Gradient 문제가 발생하여 학습이 어렵고 성능도 떨어질 수 있다. ResNet은 [그림 2]에서 확인할 수 있는 Residual Block을 사용하여 이 문제를 해결하였다. Residual Block을 사용한 결과, ResNet은 VGGNet보다 8배 더 깊은 모형임에도 더 낮은 복잡성(Complexity)을 보이며 ImageNet 데이터셋에서 좋은 결과를 보여주었다(He et al., 2015).

[그림 1] ResNet의 Residual Block



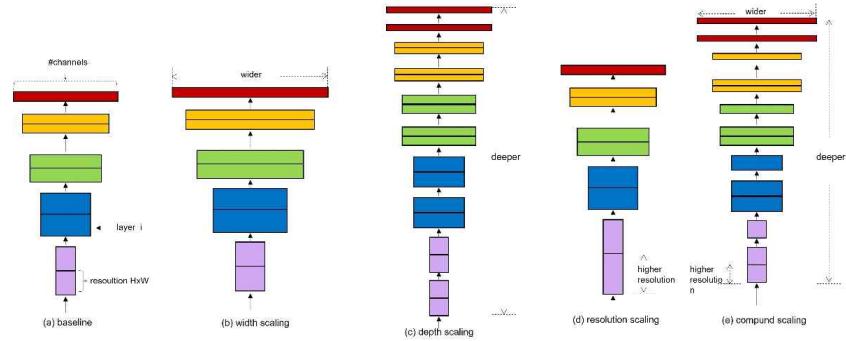
ResNet 모형은 사용한 레이어 수에 따라 ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152로 구성된다. 특히, ResNet은 18, 34개의 레이어를 사용했을 때보다 50, 101, 152개의 레이어를 사용했을 때 그 성능이 더 높아진다. 레이어가 깊어지면서 발생하는 Vanishing Gradient 문제없이 정확도가 크게 향상되었기 때문에 이미지 분류 문제에 ResNet을 많이 사용한다(He et al., 2015).

## 2.2. EfficientNet

CNN은 모형의 depth, width, resolution을 크게 만들수록 그 정확도가 한층 더 높아진다고 알려져 있다. 하지만 네트워크의 깊이(Depth)가 증가할수록 vanishing gradient 문제로 학습이 어려워지는 문제가 있고, 너비(Width)가 커지면 higher level feature를 포착하기 어렵고, 해상도(Resolution)가 매우 높아지면 오히려 정확도가 감소하는 문제가 있어 이 세 가지를 동시에 크게 증가시키기가 어려웠다. 하지만 EfficientNet은 Compound Scaling을 사용하여 위 문제들을 해결하면서도

높은 정확도를 낼 수 있었다(Tan & Le, 2019).

[그림 2] Conventional scaling과 Compound scaling의 차이 (Tan과 Le, 2019)



EfficientNet의 아키텍처는 MnasNet과 유사하나 FLOPS를 target으로 optimize하여 모형의 크기가 약간 더 커진 모습을 보인다. [표 1]에서 확인할 수 있듯이 EfficientNet은 mobile inverted bottleneck convolution(MB Conv)를 메인 블록으로 사용하고, squeeze-and-excitation(SE) 블록을 적용하였다. [표 1]의 baseline network를 시작으로, compound scaling을 적용하여 스케일을 키우면서 EfficientNet-B1부터 EfficientNet-B7까지 모형을 얻을 수 있다 (Tan & Le, 2019)

[표 1] EfficientNet-B0 baseline network (Tan과 Le, 2019)

Stage $i$	Operator $\hat{F}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $L_i$
1	Conv 3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv 1x1 & Pooling & FC	7×7	1280	1

EfficientNet은 이미지 분류 문제에서 높은 수준의 정확도를 보이면서도, 연산량을 크게 줄임으로써 매우 효율적인 모델이라는 평가를 받는다. 이에 따라, 우리는 대회 데이터에 EfficientNet을 적용하기로 결정하였으며, EfficientNet-B0부터 B7 중에서 EfficientNet-B0를 사용하기로 결정하였다.

### 2.3. Transfer learning

Transfer learning(전이학습)은 어떤 문제를 풀며 학습한 정보를 저장하여 연관된 다른 문제나 전혀 다른 문제를 해결하는 데 사용하는 학습 방법을 말한다(West 등, 2007). 과거에는 신경망의 학습에 수학적이고 기하적인 모형을 적용하여 전이 학습을 다루기 시작했으며(Stevo & Ante, 1976), 현대에 이르러서는 규모가 큰 이미지 데이터베이스로부터 학습한 CNN을 개량하여, 새롭지만 연관성 있는 다른 컴퓨터 비전 문제에 fine-tuning을 사용하여 전이학습을 적용하는 것이 효율적인 방식이 될 수 있음을 알게 되었다 (Yosinski et al., 2014; Simonyan & Zisserman, 2015).

전이학습에서 사용하는 Fine-tuning이란 pre-trained model(사전학습 모형)을 기반으로 네트워크의 아키텍처를 목적에 맞게 변형하고 사전학습 모형의 가중치를 미세하게 조정하는 방법을 말한다. Fine-tuning은 사전학습 모형을 사용한 이미지 분류 문제는 물론, 텍스트 분류에서도 사용될 정도로 널리 사용되는 방법이며, 전이 학습을 사용할 때 필수적으로 함께 사용되는 방법이기도 하다(Tajbakhsh et al., 2016; Howard & Ruder 2018).

ResNet과 EfficientNet 모두 전이학습을 위해 흔히 사용하는 ImageNet, CIFAR-100 등의 데이터셋으로 도출한 사전학습 모형이 존재한다. 두 모형 중 EfficientNet의 사전학습 모형 성능이 ResNet보다 더 뛰어나면서도, 그 연산량은 더 적었다 (Tan과 Le, 2019). [표 2]에서 전이학습에 사용한 데이터와 각 모형의 정확도, 모형 파라미터 개수를 확인할 수 있다. EfficientNet의 전이학습 모형이 다른 전이학습 모형에 비해 정확도가 높으면서도 모형에 사용된 파라미터의 개수는 매우 적음을 확인할 수 있다. 전이학습이 효율적인 성능을 보이므로 대회 데이터에 전이학습을 사용하기로 결정했다.

[표 2] EfficientNet의 전이학습 성능 결과 비교 (Tan & Le, 2019)

	<b>Model</b>	<b>Acc.</b>	<b>#Param</b>	<b>Our Model</b>	<b>Acc.</b>	<b>#Param(ratio)</b>
CIFAR-10	NASNet-A	98.0%	85M	EfficientNet-B0	98.1%	4M (21x)
CIFAR-100	NASNet-A	87.5%	85M	EfficientNet-B0	88.1%	4M (21x)
Birdsnap	Inception-v4	81.8%	41M	EfficientNet-B5	82.0%	28M (1.5x)
Stanford Cars	Inception-v4	93.4%	41M	EfficientNet-B3	93.6%	10M (4.1x)
Flowers	Inception-v4	98.5%	41M	EfficientNet-B5	98.5%	28M (1.5x)
FGVC Aircraft	Inception-v4	90.9%	41M	EfficientNet-B3	90.7%	10M (4.1x)
Oxford-IIIT Pets	ResNet-152	94.5%	58M	EfficientNet-B4	94.8%	17M (5.6x)
Food-101	Inception-v4	90.8%	41M	EfficientNet-B4	91.5%	17M (2.4x)

## 2.4. Grad-CAM

Grad-CAM(Gradient-weighted Class Activation Mapping)은 여러 CNN 모형에 Interpretability(해석가능성)와 Explainability(설명가능성)를 부여하는 기법이다. 보통, Interpretability와 Faithfulness는 서로 Trade-off 관계에 있다. 하지만 Grad-CAM은 이 두 부문에서 동시에 좋은 성과를 보이는데, 이는 CNN 모델이 Fully Connected Networks와 달리 Spatial Information을 잘 유지하기 때문이다 (Selvaraju et al., 2017). 그리고 Grad-CAM은 마지막 레이어에서 가장 많은 정보를 포함한다고 알려져 있는데, 이는 Grad-CAM의 핵심적인 아이디어라고 할 수 있다.

Grad-CAM의 구체적인 원리는 다음과 같다. CNN에서 Forward Propagate를 할 때 관심 있는 레이블을 1로, 나머지는 0으로 둔다. 그리고 Backpropagate를 통해 CNN의 Feature Map을 통합하면 Grad-CAM Localization 값을 구할 수 있다. 해당 값에 CNN이 이미지의 어떤 부분을 보고 레이블을 결정하는지에 대한 정보가 담겨 있다. 그 다음, Feature Map의 선형 결합(Linear Combination)에 ReLU 함수를 적용한다. 그 이유는 관심 레이블을 분류할 때 Positive한 방향으로 영향을 주는 Feature에만 관심이 있는데 이를 ReLU 함수가 해결해주기 때문이고, 또한 이로써 성능이 더 좋아지기 때문이다. Grad-CAM을 사용한다면, 원본 이미지에 히트맵(Heatmap)을 그림으로써 CNN이 어느 부분을 보고 레이블을 판단하였는지를 알 수 있으므로, Black Box라고 알려져 있던 기존 딥러닝 모델과는 달리 높은

Interpretability를 얻을 수 있다.

Zhang et al. (2021)에 따르면, Grad-CAM은 다른 CAM의 변형보다 성능이 더 좋고, CNN의 성능이 좋을수록 Grad-CAM의 성능도 함께 좋아진다고 한다. Zhang 등 (2021)은 뇌 이미지를 통해 경화증(Sclerosis) 여부를 분류하는 데에 grad-CAM을 사용하여 그 실용성을 선보였고, Umair 등 (2021)은 COVID-19 관련 흉부 X-ray 사진을 분류하는 과정에서 각 모델이 어느 부분에 초점을 두었는지 시각화하는 데에 grad-CAM을 활용하기도 하였다.

### 3. 연구 방법

#### 3.1. 데이터

대회에서 2,699개의 이미지 데이터를 제공하였고 이 중 1,619개를 train image dataset으로, 1,080개를 test image dataset으로 사용하였다. Train data와 test data의 구분은 대회에서 주어진 Train.csv와 Test.csv, 두 csv 파일에 기록된 ‘Image\_id’를 이미지의 파일명과 대조하여 구분하였다. Train.csv 파일은 ‘Image\_id’와 ‘Label’ 총 두 개의 칼럼으로 구성되어 있는데, ‘Image\_id’는 Train data에 속하는 이미지의 파일명을 나타내고, ‘Label’은 0 혹은 1 중 하나의 값을 나타낸다. Test.csv 파일은 ‘Image\_id’ 한 칼럼으로 구성되어 있고 Test data에 속하는 이미지의 파일명을 나타내고 있다.

Train data는 모델의 훈련에 사용하였으며, Test data는 훈련한 모델로 binary classification을 수행하여 0 혹은 1의 Label 중 어떤 값을 가지게 되는지 예측하는 데 사용하였다. Image의 width는 498부터 3200까지, height는 498부터 3200까지 다양한 크기의 이미지로 구성되어 있었고, channel은 모두 RGB 3채널로 균일했다

#### 3.2. 모형 설정

이미지 분류에 사용한 Model은 ResNet-50과 EfficientNet-B0의 사전학습 모형을 사용하였다. 학습에 앞서서 이미지 데이터 조정을 실시하였다. EfficientNet-B0를 적용하기 위해서는 이미지의 크기를  $224 \times 224$ 로 조정해 주어야 하여 이미지를 모두 resize하였고, ResNet-50에도 같은 이미지 크기를 사용하였다. 그리고 RGB 각 채널의 평균과 표준편차를 계산하여 image normalization에 사용하였다. 다음으로

[그림 3] 대회 이미지 데이터 샘플 (좌: Label 0, 우: Label 1)



훈련 이미지 데이터의 수가 1,619개로 비교적 적다고 생각하여 Data augmentation을 적용하였다. 훈련 데이터를 학습할 때 배치 크기는 32, epoch는 25로 설정하고, Validation dataset으로 train dataset의 20%를 설정하여 훈련을 진행했다. 대회에서는 모형 성능 평가에 AUC를 사용하였기 때문에 score function으로 roc\_auc\_score를 사용하였고, optimizer function으로는 Adam을, learning rate는 0.001, Loss function으로는 pytorch의 CrossEntropyLoss를 사용하였다.

### 3.3. Grad-CAM을 위한 모형 설정

본 연구에서는 이미지 분류 외에도 두 CNN, EfficientNet-B0와 ResNet-50이 이미지의 어떤 부분을 보고 ‘잎이 해충에 의한 손상을 입었는지’ 여부를 판단하는지, 그리고 이 두 모형 간에 어떠한 차이가 있는지를 보고자 하였다. 따라서 우리는 총 1,296개의 train image dataset을 학습시킨 두 모형으로, 323개의 validation image dataset을 Evaluate 한 후, 두 모형이 잘못 분류한 이미지와 잘 분류한 이미지에 대해 Grad-CAM으로 얻은 각각의 히트맵을 비교하였다.

Grad-CAM을 사용하기 위해선 모델의 마지막 Convolutional 레이어의 Activation을 사용하여야 하는데, EfficientNet-B0는 마지막 레이어가 “top\_conv”라는 레이어명을, ResNet-50은 “conv5\_block3\_3\_conv”레이어 명을 갖는다.

본 연구 방법에서는 Grad-CAM을 통해 구한 히트맵의 결과가 모형에 따라 서로 다른지에 대해서만 관심이 있으므로 두 모형을 동일한 조건에서 학습시켰다. Grad-CAM을 사용하기 위해서 두 모형 모두 Optimizer는 Adam, learning rate는

0.001, Epoch은 20으로 두고 학습하였고, 사전학습 모형을 사용했으며, 그 위에 Keras의 레이어 중 GlobalAveragePooling2D와 Dropout(0.4)을 덧붙였다.

## 4. 연구 결과

### 4.1. 이미지 분류 결과

ResNet-50과 EfficientNet-B0 두 모형에 대한 validation loss와 AUC score는 다음과 같았다.

[표 3] ResNet-50과 EfficientNet-B0의 Validation loss와 AUC

	Validation loss	Validation AUC
<b>ResNet-50</b>	0.00761	0.99846
<b>EfficientNet-B0</b>	0.00019	1.000

25번의 epoch 만에 loss가 아주 작아졌고, AUC도 매우 뛰어남을 알 수 있었다. 각 모형의 Best model을 따로 저장한 뒤, test image dataset에 적용한 결과는 다음과 같았다.

[표 4] ResNet-50과 EfficientNet-B0의 Test AUC

	AUC Score
<b>ResNet-50</b>	0.9843
<b>EfficientNet-B0</b>	0.9981

기대한 바와 같이 EfficientNet-B0의 AUC 점수가 ResNet-50보다 더 높게 나왔다. 하지만 두 수치는 크게 차이가 나지 않았는데 이는 주어진 이미지 분류 문제가

class가 두 개인 binary classification이었기 때문에 분류 난도가 낮았고, 대체로 모든 모형에서 분류를 잘 수행하는 것으로 나타나서 그 성능 차이가 크게 나타나지 않은 것으로 보인다.

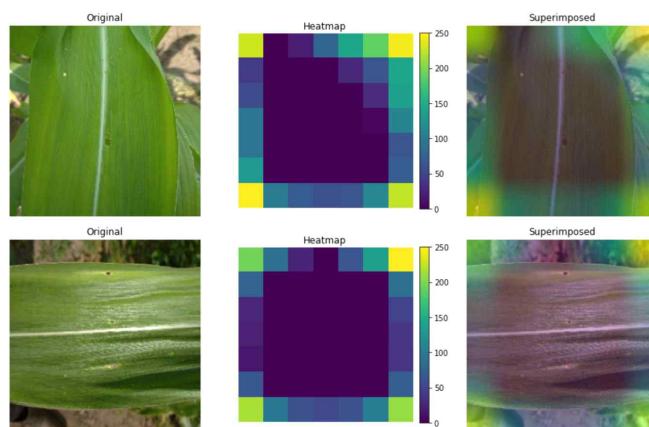
## 4.2. Grad-CAM 결과

EfficientNet-B0 모형은 총 2개의 이미지를 잘못 분류하였다. [그림 4]에 EfficientNet-B0가 잘못 분류한 이미지가, [그림 5]에는 그 히트맵이 있다. EfficientNet-B0가 레이블을 올바르게 분류한 경우의 히트맵은 [그림 6]에 나타나 있다.

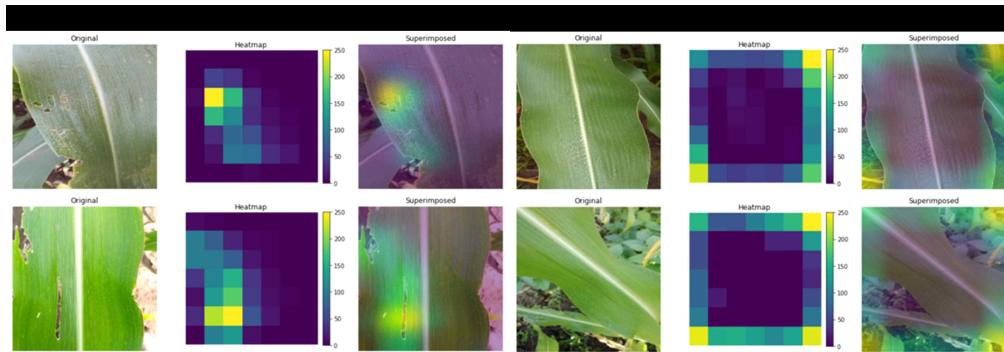
[그림 4] EfficientNet-B0가 잘못 분류한 이미지



[그림 5] EfficientNet-B0가 잘못 분류한 이미지 히트맵



[그림 6] EfficientNet-B0가 올바르게 분류한 히트맵 (좌: Label 1, 우: Label 0)



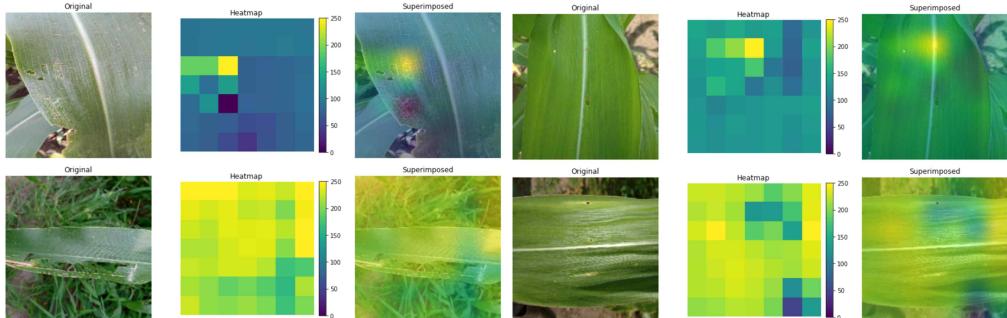
[그림 6]에서 EfficientNet-B0가 레이블 1을 분류할 때 잎의 구멍에 초점을 맞춘 것을 볼 수 있고, 레이블 0을 분류할 때 잎의 바깥 부분에 초점을 맞췄음을 알 수 있다. [그림 5]를 보면, 실제로는 레이블 1인데 0으로 잘못 분류한 이미지에 대해서도, 레이블 0을 분류했을 때처럼 잎의 바깥 부분에 초점을 맞췄음을 확인할 수 있다.

ResNet-50 모형은 총 4개의 이미지를 잘못 분류하였으며, 그 이미지는 [그림 7]에, 이미지의 히트맵은 [그림 8]에 있다. ResNet-50 모형이 레이블을 올바르게 분류한 히트맵은 [그림 9]에 나타나 있다.

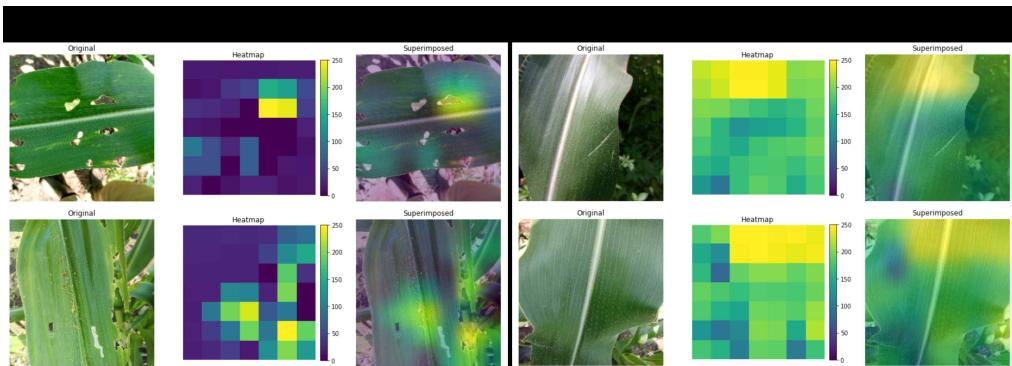
[그림 7] ResNet-50이 잘못 분류한 이미지



[그림 8] ResNet-50이 잘못 분류한 이미지 히트맵



[그림 9] ResNet-50이 올바르게 분류한 히트맵 (좌: Label 1, 우: Label 0)



[그림 9]에서 ResNet-50은, EfficientNet-B0와 동일하게, 레이블 1을 분류할 때 잎의 구멍에 초점을 맞춘 것을 볼 수 있다. 반면, 레이블 0을 분류할 때, ResNet-50은 잎의 바깥 부분에 초점을 두는 것이 아닌, 잎의 전체적인 표면에 초점을 두었음을 알 수 있다. [그림 8]에서 첫 번째 이미지를 보면 모형이 잎의 구멍 부분에 초점을 두었음에도 레이블 1로 올바르게 분류하지 못했다. [그림 8]의 두 번째, 네 번째 이미지는, 레이블 0을 분류했을 때처럼, 잎의 전체적인 표면에 초점을 두었음을 볼 수 있다. [그림 8]의 세 번째 이미지는 ResNet-50이 잎의 어떤 부분에는 초점을 두었으나, 잘못된 위치에 초점을 둔 것을 알 수 있다.

[그림 9]에서 ResNet-50은, EfficientNet-B0와 동일하게, 레이블 1을 분류할 때 잎의 구멍에 초점을 맞춘 것을 볼 수 있다. 레이블 0을 분류할 때, ResNet-50은 잎의 바깥 부분에 초점을 두는 것이 아닌, 잎의 전체적인 표면에 초점을 두었음을 알 수 있다. [그림 8]에서 첫 번째 이미지를 보면 모형이 잎의 구멍 부분에 초점을 두었음에도 레이블 1로 올바르게 분류하지 못했다. [그림 8]의 두 번째, 네 번째

이미지는, 레이블 0을 분류했을 때처럼, 잎의 전체적인 표면에 초점을 두었음을 볼 수 있다. [그림 8]의 세 번째 이미지는 ResNet-50이 잎의 어떤 부분에는 초점을 두었으나, 잘못된 위치에 초점을 둔 것을 알 수 있다.

정리하자면, EfficientNet-B0와 ResNet-50 모두 레이블 1을 분류할 때 잎의 구멍에 초점을 두었고, 레이블 0을 분류할 때는 EfficientNet-B0는 잎의 바깥을, ResNet-50은 잎의 전체적인 표면을 보면서 분류한 것을 알 수 있다. 반면, 레이블을 잘못 분류한 경우는 각각의 모형이 레이블 0을 분류할 때처럼 초점을 두었음을 알 수 있었다.

## 5. 결론

본 연구에서는 ZINDI에서 주최하는 대회에 참가하여 잎에 해충으로 인한 손상이 있는지를 분류하는 이미지 분류 연구를 진행하였다. 분석 결과 잎에 손상이 있는지를 분류하는 과제에서 EfficientNet-B0가 ResNet-50보다 좋은 성능을 보인다는 것을 확인할 수 있었다. 하지만 그 차이가 크게 나타나지 않았는데, 이는 주어진 과제가 분류 난도가 낮은 이진 분류 과제였기 때문에 이러한 결과가 나타난 것으로 생각된다. 추후 분류 난도가 더 높은 과제를 진행한다면 더 뚜렷한 차이를 알 수 있을 것으로 기대된다.

히트맵과 관련하여 얻은 결과와 제시하는 논의점은 다음과 같다. 우선, EfficientNet-B0의 히트맵이 ResNet-50의 히트맵보다 일관성 있고 이해하기 편하다는 특징을 가진다는 점을 알 수 있었다. 이는 Zhang et al. (2021)이 제시하였듯이 ResNet-50보다 EfficientNet-B0의 모델의 성능이 더 뛰어나서 히트맵의 결과도 더 좋다는 것으로 추론해 볼 수 있다. 두 번째로, 본 연구 결과에서 얻은 히트맵을 통해 모델이 잎의 바깥 부분도 본다는 것을 알게 되었다. 잎이 해충에 영향을 받았는지 판단하기 위해 잎만 보면 된다는 사실을 이용하여, 잎의 바깥 부분을 Crop하고 잎으로 분류하도록 하면 더 좋은 분류 성능을 얻을 수 있을 것으로 생각된다. 마지막으로, AUC 점수에서는 두 모델 간의 큰 차이를 볼 수 없었으나, 히트맵에서 두 모델 간의 차이를 발견했다는 것에 의의가 있다.

하지만 Grad-CAM으로 알게 된, 모형이 잎의 바깥 부분을 본다는 사실은, 연구에 사용한 EfficientNet-B0와 ResNet-50 두 모형에 한정된 결과일 수 있다는 한계점이 있다. 추후 연구에서는 더 다양한 모델을 사용하고 히트맵을 분석하여 본 연구 결과의 일관성을 확인해 볼 필요가 있다.

## 참고문헌

- [1] 백성범. (2011). RDA Interrobang 20호. 농촌진흥청.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- [3] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105–6114). PMLR.
- [4] West, J., Ventura, D., & Warnick, S. (2007). Spring research presentation: A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences*, 1(08).
- [5] Stevo, B., & Ante, F. (1976). The influence of pattern similarity and transfer learning upon the training of a base perceptron B2. In *Proceedings of Symposium Informatica* (pp. 3–121).
- [6] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *Advances in Neural Information Processing Systems*, 27.
- [7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [8] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning?. *IEEE Transactions on Medical Imaging*, 35(5), 1299–1312.
- [9] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- [10] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 618–626).
- [11] Zhang, Y., Hong, D., McClement, D., Oladosu, O., Pridham, G., & Slaney, G. (2021). Grad-CAM helps interpret the deep learning

- models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. *Journal of Neuroscience Methods*, 353, 109098.
- [12] Umair, M., Khan, M. S., Ahmed, F., Baothman, F., Alqahtani, F., Alian, M., & Ahmad, J. (2021). Detection of COVID-19 Using Transfer Learning and Grad-CAM Visualization on Indigenously Collected X-ray Dataset. *Sensors*, 21(17), 5813.

## Code Similarity Determination Using SBERT

김희민<sup>1)</sup>, 노지현<sup>1)</sup>, 황재원<sup>1)</sup>

### Abstract

To increase software productivity, it is essential to have an automated method of analysis, development, and maintenance. A first step toward this automation is to develop an AI solution to find programs that perform similar tasks. The challenge of this competition is to verify the similarity of codes for the same problem, thereby contributing to the development of automated software. We utilize BERT-based models and variant models such as RoBERTa and SBERT that are widely used in Natural Language Processing. In addition to using sample pair data provided by the competition, we build pair data processed by Meta data using BM25 algorithm. Above all, data preprocessing is the key as it includes factors like annotations in codes, which greatly affects performance. An experiment is conducted by combining types of data and 3 BERT based models. SBERT model with sample data achieves the highest accuracy of 90.7% among 5 performances.

---

1) All authors have equal contribution, Department of Artificial Intelligence,  
Chung-Ang University, Seoul 06974, Korea

## 1. 서론

현재 휴대폰부터 컴퓨터까지 광범위하게 사용되는 소프트웨어는 학습, 업무 및 엔터테인먼트 분야 뿐 아니라 타인과의 상호 작용에 이르기까지 우리 삶의 모든 층면에서 필요로 하는 곳이 확산되고 있다. 그러나 이러한 수요의 증가에 비해 전 세계적으로 양질의 소프트웨어를 공급할 개발자와 전문가가 심각하게 부족한 것이 현실이다. 일반적으로 프로그램은 개발자의 노하우와 비즈니스적 니즈가 결합된 창의적인 부분과 일반적인 기능을 수행하는 부분의 결합체인데, 제한된 개발자 공급 하에서 소프트웨어 생산성을 증대시키기 위해서는 자동화된 방식으로 분석, 개발 및 유지 관리하는 방법을 갖추는 것이 필수적이며 세계적으로 많은 연구가 진행되고 있다.

이러한 자동화를 위한 첫 단계로, 유사한 작업을 수행하는 프로그램을 찾기 위한 AI 솔루션을 개발하고자 한다. 본 대회의 과제는 동일 문제에 대한 코드들에 대한 유사성을 검증하는 것으로, 이를 통해 자동화된 소프트웨어 개발에 기여하고자 한다.

이를 위해 우리는 자연어 처리에 많이 활용되는 BERT 기반 모델과 변형 모델들을 활용하여 알고리즘을 개발하였다. 대회에서 제공해주는 sample pair 데이터와 Meta 데이터를 가공한 pair data 활용하여 알고리즘을 학습시켰으며, 두 결과를 비교하여 최종 결과에 사용할 데이터를 선정하였다. 코드 작성 시에는 코드 외에도 주석 등이 포함되어 데이터 전처리가 핵심이며, 성능에 큰 영향을 미친다. 알고리즘 성능 평가 지표는 accuracy가 사용되었다.

## 2. 데이터

### 2.1. 데이터 설명 및 전처리

본 대회는 Meta 데이터와 sample pair 데이터를 제공해주고 있다. Meta 데이터는 300개의 문제를 해결하는 여러개의 Python 코드이다. Sample pair 데이터는 Meta 데이터를 가공하여 17000여개의 pair를 생성한 것이다. 각 pair는 서로 다른 문제를 해결하는 코드일 경우 0, 서로 같은 문제를 해결하는 코드일 경우 1로

labeling되어 있는데, 해당 similarity를 binary classification하는 것이 대회의 과제이다. Sample train 데이터는 17,970개, test 데이터는 17,9700개이다.

Python 코드에는 주석이 포함되어 있는데, 이는 코드간의 유사성을 판단하기에는 불필요하고 성능에 부정적인 영향을 줄 것이라고 판단하여 제거하였다. 또한 코드 작성 시 들여쓰기, 띄어쓰기, 다중 행 등을 제거 등 전처리 과정을 수행하였다.

## 2.2. BM25를 활용한 Pair data 생성

Sample pair 데이터 외에, 우리는 pair data를 추가적으로 생성하여 성능 향상을 도모하고자 하였다. 이를 위해 키워드 기반의 랭킹 알고리즘인 BM25(Stephen Robertson et al., 2009)를 적용하여 코드의 유사성을 판단하여 라벨링하였다. 해당 알고리즘은 ranking 함수의 종류 중 하나로, 주어진 쿼리에 대해 문서와의 연관성을 평가할 수 있다. BM25의 수식은 다음과 같다.

$$score(D, Q) = \sum_{i=1}^n IDF(q_i) * \frac{f(q_i, D) * (k_1 + 1)}{f(q_i, D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

$q_i$ : 쿼리에서  $i$ 번째 토큰 (형태소 / Bi-gram/ BPE 등 사용 가능)

$IDF(q_i)$ : 쿼리의  $q_i$ 번째 토큰에 대한 inverse document frequency

$avgdl$  : 문서 집합의 평균 문서 길이

Bag-of words 개념을 사용하여, 쿼리에 있는 용어가 각각의 문서에 얼마나 자주 등장하는지를 평가하는데, 해당 모델을 사용하여 negative pair는 멀어지도록, positive pair는 가까워지도록 학습하였다. BM25를 각 문제의 첫번째 코드를 기준으로 적용하였으며, 코드와의 연관성을 검토하여 scoring한 후, 이를 기반으로 다른 코드들과의 pair 구성하고 유사성 여부를 labeling할 수 있다. 해당 알고리즘을 활용하여 5,133,767개의 pair 데이터를 생성하였다.

### 3. 방법론

해당 태스크에 적용한 핵심 모델은 자연어 처리 분야에 크게 기여한 BERT(Bidirectional Encoder Representation from Transformer)이며, BERT를 기반으로 발전한 여러 모델 중 RoBERTa와 SBERT을 추가 적용하여 실험을 진행하였다.

#### 3.1. BERT

BERT는 구글에서 개발한 자연어처리 사전 훈련 언어모델이며, 특정 분야에 국한된 기술이 아니라 모든 자연어 처리 분야에서 좋은 성능을 내는 범용 Language Model이다.

BERT의 핵심은 Transformer 모형에서 Encoder 부분만 사용하고 pre-training과 fine-tuning 시 구조를 조금 다르게 하여 Transfer Learning을 용이하게 만드는 것이다. 기존의 모형들은 문장의 길이가 길어질수록 첫 단어의 의미가 끝 단어의 의미까지 반영되기 어렵다는 한계점이 있었다. 이에 Attention을 추가한 RNN 구조를 사용하였지만 이는 연산 속도가 매우 느리다는 단점이 존재했다. 그 단점을 보완하고자 Attention만을 사용하는 신경망을 구성하는 방법이 제안되었고, 그 방법이 바로 Self-Attention을 사용하는 Transformer 모델이다. BERT는 Transformer 블록으로 이루어져 있고, unlabeled text에서 양방향 심층 표현을 pre-train하도록 설계함으로써 성공적인 performance를 이끌어내고 있다.

BERT의 내부 동작 과정은 다음과 같다.

##### 1. Input : Token Embedding + Segment Embedding + Position Embedding

BERT(Devlin et al., 2018)는 WordPiece embedding을 사용하여 문장을 token 단위로 분리한다. 단순히 띄어쓰기로 토큰을 나누는 것이 아닌 각 문자 단위로 임베딩을 하고, 자주 등장하면서 가장 긴 길이의 sub-word를 하나의 단위로 만든다. 자주 등장하지 않는 단어는 다시 sub-word로 만든다. 이는 이전에 자주 등장하지 않았던 단어를 모두 OOV(Out-Of-Vocabulary) 처리하여 모델의 성능을 높였다. 즉, 신조어나 오탈자가 있는 입력값도 딥러닝 모델이 학습 단계에서 보았을 만한

단어들로 나눠 입력되어 흔하지 않은 단어들에 대한 예측이 가능하게 된다. Segment Embedding은 토큰 시킨 단어들을 다시 하나의 문장으로 만드는 작업이다. BERT에서는 두개의 문장을 구분자([SEP])를 넣어 구분하고 그 두 문장을 하나의 Segment로 지정하여 입력한다. 이때 한 segment를 512 sub-word 길이로 제한한다.

Self Attention은 입력의 위치를 고려하지 않고 입력 토큰의 위치 정보를 고려하기 때문에, Sinusoid 함수를 이용한 Positional encoding을 사용한다.

위의 세 가지 embedding을 합친 결과에 layer normalization과 dropout을 적용하여 모델의 Input으로 사용한다.

[그림 1] BERT input representation (input embeddings)

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	# #ing	[SEP]
Token Embeddings	$E_{[CLS]}$	$E_{my}$	$E_{dog}$	$E_{is}$	$E_{cute}$	$E_{[SEP]}$	$E_{he}$	$E_{likes}$	$E_{play}$	$E_{#ing}$	$E_{[SEP]}$
Segment Embeddings	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_A$	$E_B$	$E_B$	$E_B$	$E_B$	$E_B$
Position Embeddings	$E_0$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$	$E_8$	$E_9$	$E_{10}$

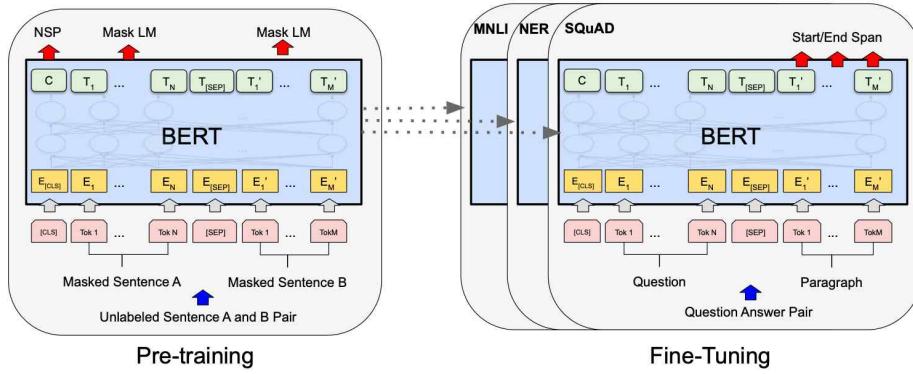
## 2. Pre-training

기존의 방법들은 보통 문장을 왼쪽에서 오른쪽으로 학습하여 다음 단어를 예측하는 방식이거나, 예측할 단어의 좌우 문맥을 고려하여 예측하는 방식을 사용한다. 하지만 BERT는 언어의 특성을 잘 학습하도록, MLM(Masked Language Model)과 NSP(Next Sentence Prediction) 두 가지 방식을 사용한다. MLM란 문장의 중간 단어를 마스킹한 후 전체 문장에서 해당 단어를 예측하는 방식으로 학습하는 것이고, NSP는 두 문장이 이어지는 관계인지 아닌지를 학습하는 것이다.

## 3. Fine-tuning

BERT는 각 task에 따라 간단한 레이어를 추가하고, 적은 데이터와 학습 시간으로 Fine-Tuning만 거쳐도 기존의 각 테스크별 SOTA 모델들을 압도하는 성능을 보여주었다.

[그림 2] Overall pre-training and fine-tuning procedures for BERT



### 3.2. RoBERTa (Robustly optimized BERT)

BERT는 구글에서 개발한 자연어처리 사전 훈련 언어모델이며, 특정 분야에 국한된 기술이 아니라 모든 자연어 처리 분야에서 좋은 성능을 내는 범용 Language Model이다.

RoBERTa(Robustly optimized BERT approach)는 BERT의 성능을 높인 기술 중 하나로 BERT의 파라미터 및 트레이닝 방법의 변화를 통해 성능을 향상시켰으며, 다음과 같은 tuning을 진행하였다(Liu et al., 2019).

1. 기존 BERT는 단 한번의 random mask를 적용(Static Masking)하여, 모든 epoch에서 동일한 mask가 반복되었다. 반면, RoBERTa는 epoch마다 다른 masking을 수행하는 Dynamic Masking을 적용하였다. 특히 이는 큰 데이터셋을 pre training 할 때 중요한 방법으로 기존 masking보다 나은 성능을 보였다.

2. NSP(Next Sentence Prediction)가 BERT 모델 사전 학습에 유용하지 않음을 밝히며 NSP를 objective를 제거하였다. 이 외로 과거 NMT 연구(Scaling Neural Machine Translation)에서 아주 큰 mini-batch로 학습하면 적절한 학습률을 사용할 때 optimization speed와 end-task performance를 모두 향상시킬 수 있다는 것이 밝혀졌다. (Myle Ott 등, 2017)
3. 기존의 BERT는 Wordpiece tokenizing 방법 중 하나인 30K의 문자 수준 BPE(Byte-Pair Encoding) vocabulary을 사용했으며, RoBERTa에서는 부가적인 입력의 전처리나 토큰화 없이, 50K 서브워드 유닛을 포함한 바이트 수준의 더 큰 BPE vocabulary로 BERT를 학습시켰다. BPE는 일부 테스크에서 최종 테스크 성능을 약간 하락시켰으나, 보편적인 인코딩 전략이 성능 하락보다 더 중요하다고 여기고 이러한 인코딩 방식을 적용시켰다.

추가로 생성한 pair 데이터는 5,133,767개로, batch size를 크게 설정하여 학습이 가능하다. 이는 성능 향상으로 이어질 수 있다. 또한 해당 논문에서 사용하는 Dynamic masking 방식은 데이터 크기가 클 수록 이점을 가진다. 따라서 우리는 RoBERTa model 활용에 따른 성능 향상을 기대하였다.

### 3.3. SBERT (Sentence Bert)

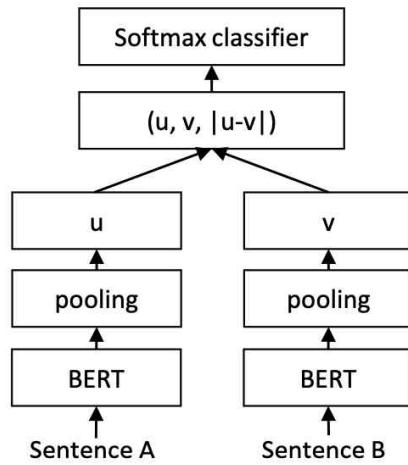
SBERT(Reimers et al., 2019)는 BERT의 문장 임베딩 성능을 우수하게 개선시킨 모델이다. BERT의 문장 임베딩을 응용하여 BERT를 Fine tuning한 모델이다. 2개의 downstream task는 다음과 같다.

#### 1. 문장 쌍 classification

문장 쌍 classification task 중에서도 NLI(Natural Language Inferencing) 문제를 푸는 것에 집중하여 fine tuning을 진행하였다. 두 개의 문장이 주어졌을 때, 수반 관계인지, 모순 관계인지, 중립 관계인지를 맞추는 multiclass classification task이다. 문장 A와 문장 B 각각을 BERT의 입력으로 넣고, BERT의 문장 임베딩을 얻기 위한 방식은 mean pooling 혹은 max pooling을 통해서 각각에 대한 문장 임베딩 벡터를 얻는다. 여기서는 이를 각각 u와 v라고 한 후 u벡터와 v벡터의 차이 벡터를 구한다. 이 벡터는 수식으로  $|u-v|$ 로 표현할 수 있으며, 이 세 가지 벡터를 concatenation한다. 그리고 이 벡터를 출력층으로 보내 classification 문제

를 풀도록 한다. softmax 함수까지 적용한 후 실제값에 해당하는 label로부터 오차를 줄이는 방식으로 학습시킨다. 구조를 도식화하면 다음과 같다.

[그림 3] SBERT의 문장 쌍 분류 과정



## 2. 문장 쌍 regression task

두 개의 문장으로부터 의미적 유사성을 구하는 STS(Semantic Textual Similarity) 테스크를 수행한다. 유사도의 범위값은 0~5이며, 유사도가 높을 수록 5에 가까운 label을 가진다. 문장 A와 문장 B 각각을 BERT의 입력으로 넣고, 각각에 대한 문장 임베딩 벡터를 얻는다. 여기까지는 1의 task와 동일하다. 그 후 이를 각각  $u$ 와  $v$ 라고 하였을 때 이 두 벡터의 코사인 유사도를 구한 후 해당 유사도와 레이블 유사도와의 평균 제곱 오차(Mean Squared Error, MSE)를 최소화하는 방식으로 학습한다. 코사인 유사도의 값의 범위는 -1과 1사이므로 위 데이터와 같이 레이블 스코어의 범위가 0~5점이라면 학습 전 해당 레이블들의 값들을 5로 나누어 값의 범위를 줄인 후 학습할 수 있다.

SBERT는 문장 간 관계를 파악하기에도 용이하기 때문에 다수의 문장이 있을 경우 (챗봇 구현, 글 창작 등)에도 많이 사용된다. 두 코드의 유사성을 확인하는 과정에서는 두 코드를 두 개의 문장이라고 간주한 후 둘 사이의 유사성을 계산하여 유

사도 여부를 labeling하는 방식으로 이해하여 적용할 수 있다.

## 4. 실험 구성 및 결과

### 4.1. 실험 설계

사용 데이터의 종류와 모델의 종류에 따라 조합하여 실험을 진행하였다. 우리는 자연어 처리 분야에 있어서 기본적이고 널리 알려져 있는 BERT와 이를 변형한 SBERT, RoBERTa의 3가지 모델을 선정하였다. 데이터는 Sample Data와 추가로 생성한 Pair Data을 사용하였는데 Pair Data는 개수가 너무 많아서 2가지 방식으로 나누어 적용하였다.

- $\frac{2}{3}$  만큼 random resampling
- Early stopping

3가지 모델과 3가지 데이터 처리 방식을 조합한 후, 실험을 진행하였다.

### 4.2. 실험 결과

Metric은 accuracy로 전체 샘플의 개수 중에서 얼마나 우리의 알고리즘이 정답이라고 예측한 샘플이 포함되었는지의 비율을 의미한다.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Number of samples correctly predicted}}{\text{Total number of samples}} \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned}$$

[표 1] Test accuracy for three adjusted datasets and three models

Model	Datasets	Accuracy
BERT	Sample data	0.7916
SBERT	Sample data	<b>0.9073</b>
	Early Stopping	0.6805
RoBERTa	Sample data	0.7801
$\frac{2}{3}$	Random data	0.8919

실험 설계에 따른 Accuracy 결과는 [표 1] 과 같다.

Sample 데이터를 활용하여 SBERT 모델에 적합시킨 결과의 accuracy가 0.9073으로 가장 높았고, 새로 생성한 pair 데이터를 resampling하여 RoBERTa 모델에 적합시킨 accuracy가 0.8919로 두 번째로 높았다. 대체로 우리가 생성한 Pair 데이터를 활용한 결과보다는 대회 측에서 제공해준 sample 데이터를 활용한 모형의 accuracy가 더 높은 것을 확인할 수 있었다. Early stopping을 적용한 결과는 다소 낮은 성적을 보였는데, 최적점에 도달하기 전 학습을 중지한 것으로 판단하였다. 모형의 결과들을 고려하여 SBERT와 sample data의 조합을 최종 모형으로 선정하였다.

## 5. 결론 및 시사점

딥러닝 자연어 처리 분야에 대한 여러 가지 모델을 직접 학습시키고 검증해보는 등 다양한 시도를 해볼 수 있었다. 이러한 시도 끝에 SBERT와 sample data의 조합의 성능이 90.073으로 우리가 설계한 모형들 중 가장 성능이 좋다는 것을 확인할 수가 있었다. SBERT는 두 문장의 유사도를 기반으로 학습시킨 모델이기에 대회 task에서 높은 성능을 달성했을 것이라고 판단하였다. 프로젝트를 종료 후 느낀

점 및 시사점은 다음과 같다.

첫째, sample pair 데이터 외에 알고리즘에 기반하여 추가 데이터를 생성하였으나 여러 가지 모형에서 성능이 나아지지 않았다. 데이터의 수를 늘렸더라도 전처리를 진행하지 않은 데이터일 경우 오히려 성능에 악영향을 줄 수 있음을 실감할 수 있었다. 전처리의 중요성을 실감할 수 있었으며, 다양한 tokenizer와 전처리 방식 시도를 통해 성능 향상을 이룰 수 있을 것이다(Peeters et al., 2020).

둘째, 모형 학습 과정에서 graphcodebert(Guo et al., 2021), CodeBERT(Zhangyin Feng 등, 2020), CodeBERTaPy등의 코드 기반 언어모델을 시도해 본다면 더 좋은 분류기가 형성될 것이다.

## 참고 문헌

- [1] Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4), 333–389.
- [2] Devlin, J., Chang, M. W., Lee, K. and Toutanova K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.
- [3] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks, Ubiquitous Knowledge Processing Lab (UKP-TUDA) Department of Computer Science, Technische Universitat Darmstadt.
- [4] Ott, M., Edunov, S., Grangier, D., & Auli, M. (2017). Scaling Neural Machine Translation, Facebook AI Research, Menlo Park & New York.
- [5] Liu. Y., Ott. M., Goyal. N., Du. J., Joshi. M., Chen. D., Levy. O., Lewis. M., Zettlemoyer. L., & Stoyanov. V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach, Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA.
- [6] Tracz, J., Wojcik, P., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R. & Gawlik, I. (2020). BERT-based similarity learning for product matching, *Proceedings of the Workshop on Natural Language Processing in E-Commerce (EComNLP)*.
- [7] Peeters, R., Bizer, C. & Glavaš, G. (2020). Intermediate Training of BERT for Product Matching, *Proceedings of the 2nd International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs co-located with 46th International Conference on Very Large Data Bases (DI2KG 2020)*.

## Comparison of Algorithm Performance for Handling Data Imbalance

김병찬<sup>1)</sup>, 김희원<sup>1)</sup>, 최지은<sup>1)</sup>, 차호영<sup>1)</sup>

### Abstract

With the development of smart factories, the production of defective products has decreased. However, the defective products inevitably occurs. In the field of Computer vision in smart factories, the classification of defective products and genuine products have continuously studied. However, the number of defective products is significantly lower than the number of genuine products. Therefore, data for learning defective products is insufficient, and data imbalance between genuine and defective products occurs. This study is to find out what algorithms are good for solving data imbalance. We tried various algorithms based on 'ResNet101'. Among them, Up-Sampling showed the best performance with F1\_Score of 0.732 and Cost-Sensitive Learning showed poor performance with 0.51. As a future research plan, we will overcome the limitations that dealing with imbalances by dividing defective products labeled with various 'State' into binary forms such as 'Good' and 'Bad'.

---

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

### 1.1 연구배경 및 목적

공정 제조는 공급품, 재료 또는 원자료에 기계적 혹은 화학적 작용을 통해 새로운 제품으로 전환시키는 작업 활동이다. 4차 산업의 발전에 따라 공정 제조 또한 발전하는 추세를 보여주고 있다. 2016년 스위스 Davos-Kloster에서 “제4차 산업 혁명 마스터하기”라는 주제로 세계 경제 포럼 연례회의가 개최되었다. 세계 경제 포럼 창립자 겸 회장인 Klaus Schwab는 그의 저서에서 제4차 산업혁명을 선언하면서 이 네번째 혁명이 기술 발전에 의한 혁명이라고 정의하였다. Klaus Schwab의 제4차 산업혁명 선언 이후 생산 제조 기술과 정보통신기술(ICT)이 융합된 스마트 팩토리가 큰 주목을 받으면서 주요 선진국은 제조업의 중요성을 강조하면서 자국의 제조 경쟁력을 높여나가고 있다. 2018년 산업연구원의 정책자료에 따르면 독일은 국가 차원에서 제조업과 관련한 다양한 논의 끝에 Industry 4.0를 추진하고 있으며 현재 Platform Industry 4.0으로 발전시키고 있다. 또한 미국은 세일 가스와 IT 중심의 ‘Reshoring’, 일본은 ‘소사이어티 5.0’, 중국은 ‘중국 제조 2025’ 등을 추진하면서 제조업 중심의 전략을 수립하여 제조업 혁신을 추진하고 있다.

제조업 혁신을 위한 각국의 노력에 더불어 사물인터넷(IoT) 기술, 빅데이터 저장 및 분석 기술, 산업용 로봇 기술이 발전하면서 생산 시스템의 자동화가 가능해졌다. 자동화된 생산 시스템에서는 제조 공정에 따라 여러 종류의 센서 데이터들이 실시간으로 생성된다. 공정이 진행되는 동안 실시간으로 생성 된 데이터들은 정상 데이터와 이상 원인에 의한 이상 데이터이다. 자동화된 생산 시스템이 고도화됨에 따라 그에 맞추어 이상 탐지 역시 고도화된 시스템을 통해 이루어질 필요가 있다. (Cho et al., 2021)

또한, 스마트 공장의 발전에 따라 이상 품목의 생산이 줄어 들며, computer vision에 있어 이상 품목을 학습하기에 내용이 부족하며 학습을 위해서는 불균형한 데이터를 해결할 방법을 찾는 연구가 중요해졌다. 각종 제조업 도메인 이미지 데이터에 대한 외관 검사, 외관 상에 발생하는 결함과, 장비의 고장 등의 비정상적인 sample이 굉장히 적은 수로 발생하지만 정확하게 예측하지 못하면 큰 손실이 유발된다. 이상이 발생되면 제조 공정이 중단되어 매우 큰 손실로 이어지기 때문에 센서 데이터를 통한 제조 설비의 결함 진단은 정확하고 신속해야 한다. 전문가들의 경험을 기반으로 제조 설비의 결함을 진단하거나 제품의 불량 유무를 알아내는 것에 한계가 있어 신속하고 정확한 이상 탐지를 수행하기 어렵다. 제조 공정 시스템

의 복잡도가 증가하면서 센서 데이터와 제조 공정 상태 및 제품의 불량 검출 여부를 연결 지어 관리하는 것이 단순하지 않은 작업이 되어 신속하고 정확한 이상 탐지를 위해 새로운 관리 기법이 필요해졌다. 이는 우리가 참가하게 된 Computer Vision 이상 탐지(Anomaly Detection) 알고리즘 경진대회가 나오게 된 배경이다.

본 대회 목표는 사물의 종류를 분류하고, 정상 샘플과 비정상(이상치) 샘플을 분류하고자 한다. 또한, 불균형 데이터 셋을 학습하여 사물의 상태를 잘 분류할 수 있는 알고리즘을 개발하고자 한다. 이를 위해 우리는 Data Augmentation(증강), Batch Sampling 등 다양한 방법을 진행해봤다. 알고리즘의 평가 산식은 분류에서 가장 인기 있는 Macro-F1score로 평가된다.

## 2. 기존 방법론

Computer Vision 분야에서 자주 사용되는 모델들의 기반은 Image classification이다. 먼저 image classification에 대해 이해하기 위해 CNN기반 Image classification의 기초 모델을 먼저 소개하고 이를 바탕으로 대회의 베이스 라인 모델인 EfficientNet에 대해서 소개하도록 한다.

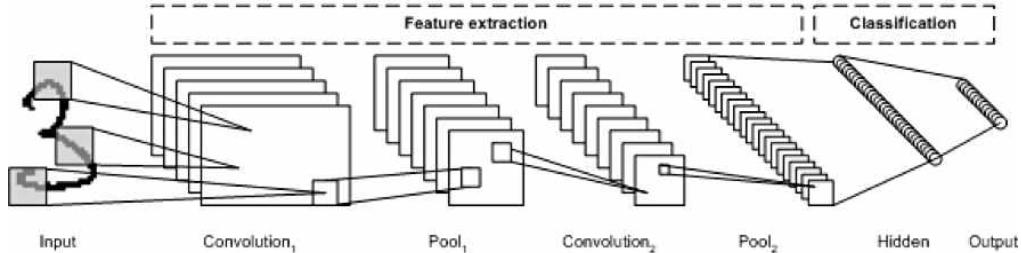
### 2.1 Image Classification Task

이미지 분류는 이미지 내 특정 사물을 분류하는 것이다. CNN은 인간의 시신경 구조를 모방한 Convolution과 Pooling을 반복해 특징(Feature)을 추출하고, 완전연결계층(Fully connected layers)을 통해 입력된 이미지를 분류하기 위한 변별적 학습을 수행한다. CNN은 학습해야 할 전체 파라미터 수를 감소시켜 빠른 학습 속도와 우수한 일반화 능력을 가질 수 있도록 도와준다. CNN구조를 이용한 이미지 분류의 대표모델로는 EfficientNet, VGG, ResNet 등이 있다.

### 2.2 EfficientNet

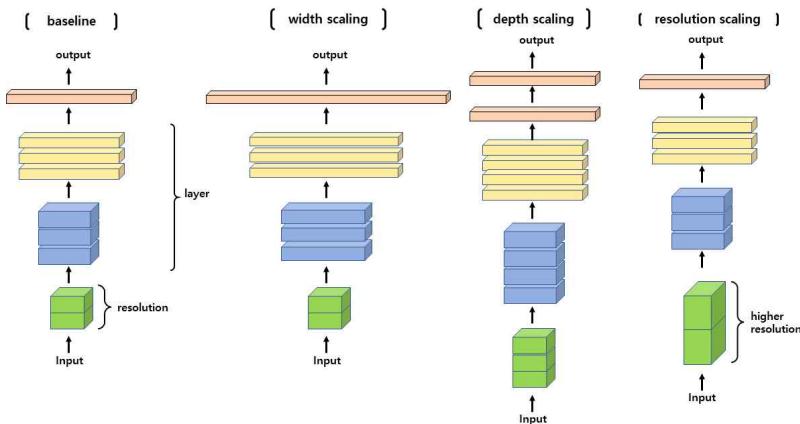
CNN은 더나은 성능을 위해 계속해서 발전되어왔다. CNN의 표현력에 있어서 광장히 중요한 요인은 [그림 2]에서 보이는 것처럼 크게 3가지가 있다. network의

[그림 1] CNN의 특징 추출(Feature extraction)과 분류(Classification) 예시



depth를 깊게 만드는 것, width가 넓을수록 미세한 정보도 많이 고려되기 때문에 channel width(filter개수)를 늘리는 것, input 이미지의 해상도를 올리는 방법이 있다.

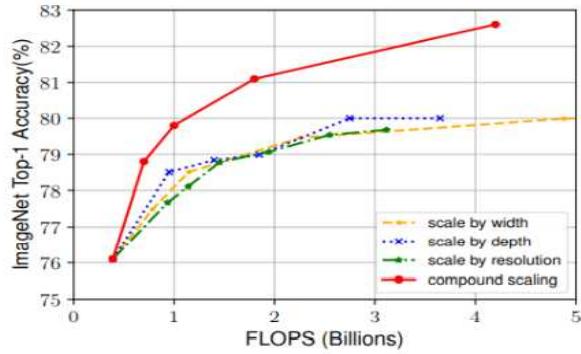
[그림 2] 모델을 개선시키는 3가지 요소



이전에는 이 세 차원 중 하나만 개선시키는 것이 일반적이었지만, EfficientNet에서는 세 차원의 최적의 조합을 효율적으로 만들 수 있도록 하는 compound scaling 방법을 제안했다. [그림 3]은 EfficientNet-B0 network에 다양한 scaling method를 적용한 ImageNet Top1 Accuracy를 보여준다. 모든 scaling method가 더 많은 FLOPS와 함께 성능을 향상시키지만, compound scaling method가 다른 single-dimension scaling method보다 2.5%가량 accuracy를 더 향상시킨 것을 확인할 수 있다(Tan et al., 2020).

[그림 3] EfficientNet scaling method 적용

성능



### 3. 연구 방법론

이미지 분류를 효율적으로 하기 위해 우리는 다양한 데이터로부터 가중치를 학습시킨 사전학습모델을 활용하여 이미지 분류에 있어 정확도를 향상시켰다. PyTorch에서 제공하는 다양한 사전학습모델들 중 우리가 분류해야 될 객체에 적합한 모형을 찾기 위해 10가지의 사전학습 모델을 비교하였고 가장 Validation dataset 기준으로 가장 성능이 좋았던 모델을 기준으로 데이터의 불균형한 문제 처리를 실시하였다. 데이터 불균형 처리는 크게 4가지 방법을 생각했는데 불량 데이터를 늘리는 오버샘플링 기법, 양품 데이터셋의 개수를 낮추는 다운샘플링 기법, 그리고 불양품의 개수를 다양한 Data Augmentation으로 데이터 수를 늘리는 방법, 마지막으로 데이터 레이블별로 가중치를 다르게 주는 cost sensitive method를 활용해서 데이터 불균형 문제를 다뤄 보았다.

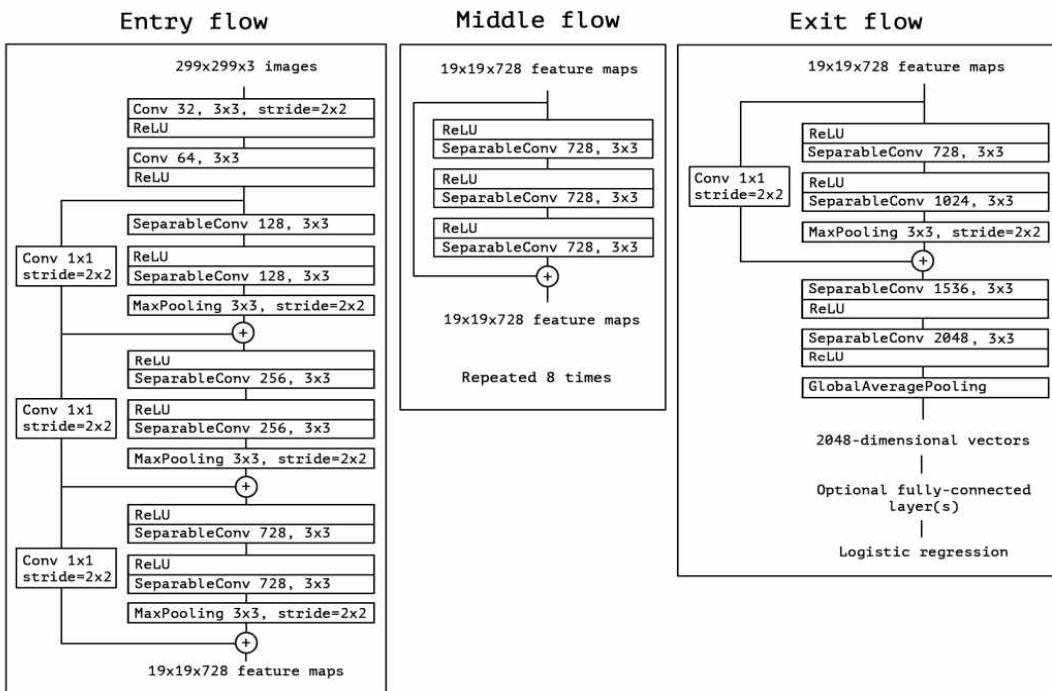
#### 3.1. Xception

eXtreme Inception의 약어로 그 전의 모델인 Inception을 depthwise separable convolution으로 성능을 향상시킨 모델이다(Chollet, 2017). Inception model은 convolution 레이어에서 2개의 width와 height의 2개의 차원과 채널 간 상관관계를 동시에 매핑하는 작업을 수행한다. 즉 cross channel 상관관계와 공간 상관관계를 독립적으로 살펴보는 작업으로서 이 프로세스를 더욱 쉽고 효율적으로

만든다. Xception은 이 기본적인 inception 모델의 발전된 모델로서 depthwise separable convolution layer를 기반으로 하는 신경망 아키텍처를 제안한다. 즉, convolution 신경망에서 피쳐맵 간 교차채널 상관 및 공간 상관의 매핑이 완전하게 분리될 수 있다는 가설을 세울 수 있다. 이 네트워크는 피쳐 추출을 위한 36개의 convolution 레이어를 가지고 있으며 실제 논문에서 실험했던 이미지 분류를 위해서 로지스틱 회귀 레이어를 따른다. Xception 구조는 residual connection을 가진 depthwise separable convolution의 선형 스택 구조이며 매우 쉽게 정의되고 수정될 수 있도록 Keras 또는 Pytorch의 timm 라이브러리에서 사용이 가능하였기 때문에 이 사전학습 모델을 사용하였다.

[그림 4]에 제시된 Xception 구조를 살펴보면, 데이터가 처음 entry flow에 들어가게 되면 middle flow에서 8번의 반복이 이루어진다. Middle flow를 거쳐 exit flow에 들어가게 되면서 실험에서는 분류 문제를 해결하기 위해 Logistic regression에 레이어를 거쳐 정답을 산출하게 된다. 모든 convolution layer와 separable convolution layer는 batch normalization 과정을 거쳐 오버피팅의 가능성을 줄여준다.

[그림 4] Xception Network의 구조



### 3.2 VGG

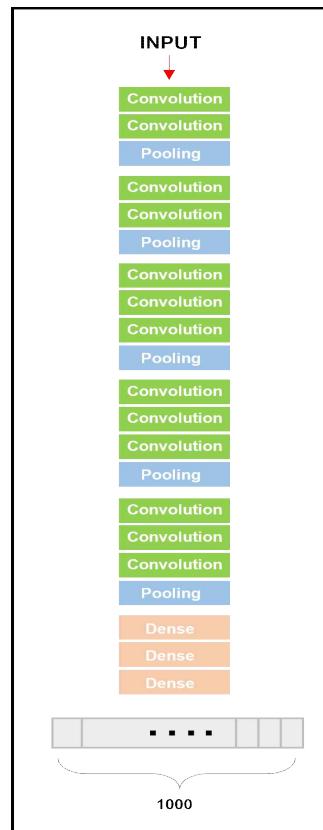
VGG는 매우 작은  $3 \times 3$  convolution 필터가 있는 아키텍처를 사용하여 깊이를 증가시키는 네트워크이다. 16개 또는 19의 웨이트 레이어를 넣음으로써 이미지 분류에 있어서 뛰어난 성능향상을 보인 모델이다(Simonyan et al., 2015). 이 모델의 실제 아키텍처는  $224 \times 224$ 의 RGB의 이미지파일을 사용한다. 특별한 데이터의 전처리가 필요없으며 정규화를 위해 각 픽셀의 평균 RGB 값을 빼는 식의 전처리만 진행된다. 이미지가 레이어를 거치면서 작은  $3 \times 3$  크기의 필터를 통해 convolution이 진행되는데 이  $3 \times 3$ 의 필터는 상하좌우를 캡쳐하기 위한 최소한의 사이즈이다. 즉 여러 층의 convolution layer를 거치는데 가장 작은 크기의 필터를 사용함을 특징으로 한다. 여러 층의 convolution layer를 거친 후 3개의 Dense layer를 거쳐서 태스크에 해당되게끔 1000개의 label로 나오게끔 소프트맥스 함수로 마지막 층이 구성되어 있다. 모든 convolution layer는 ReLU함수가 적용되어 있으며 단순히 convolution layer와 최소크기의 필터만 사용해서 이미지 분류에 뛰어난 성과를 보인 점에 의의가 있다.

[그림 5]는 VGG16의 기본구조를 보여주고 있다. 이 아키텍처는 convolution layer를 거친 후 max-pooling을 실행한다. 구조에서 알 수 있듯 weight 값을 업데이트하는 13개의 convolution layer와 3개의 Dense layer가 적용되어서 VGG16이라는 이름이 붙었다. convolution layer를 거치면서 필터의 크기는 2배씩 증가하게 되며 이미지의 크기는 작아지며 압축이 된다. Dense layer는 convolution layer를 거쳐서 압축된 이미지를 4096차원으로 출력되게끔 전결합층을 형성한 후 마지막에 태스크에 맞게끔 1000개의 차원으로 소프트 맥스함수를 적용하여 구조가 이루어져 있다. 이번 태스크에서 진행할 이미지 크기를 조절하여 VGG16 모델에 적용해 보았고 복잡하지 않은 구조이기에 이미지 셋이 많지 않아 잘 적합될 것이라고 예상하였다.

### 3.3 ResNet

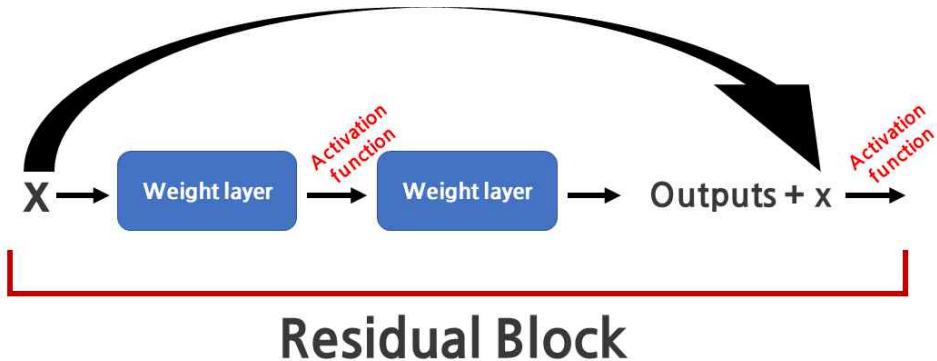
레이어가 깊어졌을 때의 딥러닝 학습을 효율적으로 이루어 질 수 있게끔 residual learning이라는 방법을 적용한 모델이다. 딥러닝은 모델의 깊이가 깊어질

[그림 5] VGG16의 구조



수록 학습이 효율적으로 되고 있는가라는 중요한 질문에 직면한다. vanishing gradients 문제는 학습 레이어가 깊어지면서 발생을 하며 이와 동시에 네트워크의 정확도가 떨어지는 문제가 생긴다(Hochreiter et al., 1998). 이를 해소하기 위해 resnet은 맵핑을 식별하는 layer를 추가한다.

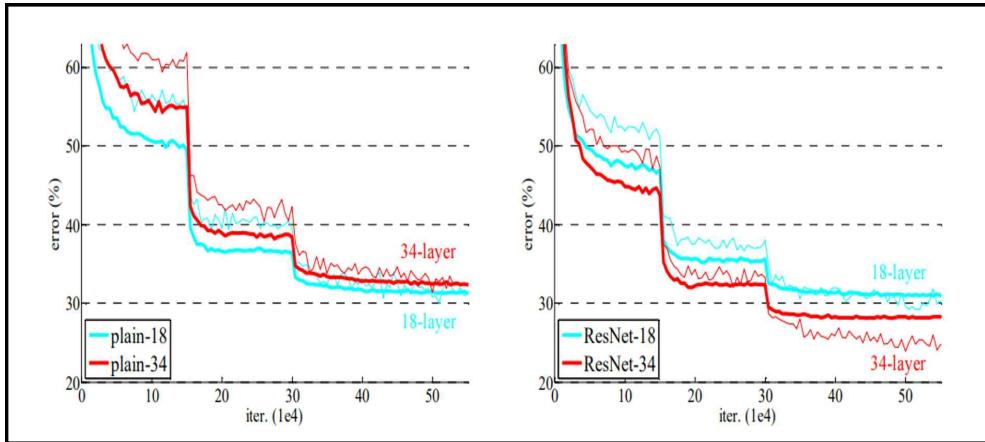
[그림 6] Residual Learning



[그림 6]은 이 모델에서 초기값을 맵핑하는 과정을 보여준다. weight layer에서 ReLU 연산을 끝낸 후 나온 값을 초기값과 더해줌으로써 정보의 손실을 막아준다. 이러한 구조는 쉽게 최적화가 되며 깊은 신경망에서 훈련 오류가 발생하는 것을 어느 정도 억제해주며 기존의 네트워크보다 네트워크의 수가 증가하더라도 높은 정확도를 얻을 수 있다(He et al., 2016).

즉, 핵심적인 이 모델의 주안점은 단순히 layer를 거쳐 층수를 늘리는 것이 아닌 초기의 정보를 저장하는 장치를 추가하는 것이다. 그래서 각각의 쌓여진(stacked) layer에 이 residual learning term을 넣어서 블록을 형성한다. 논문에서는 VGG16 모델과 33개의 convolution layer와 1개의 dense layer를 추가한 plain network, 그리고 plain network에 residual learning term을 추가한 resnet 모형을 가지고 실험을 실시하였고 [그림 7]을 보면 알 수 있듯 plain network는 layer가 깊어질수록 error가 커졌지만 resnet 모형은 오히려 layer가 깊어졌음에도 더 좋은 성능을 보임을 확인할 수 있었다.

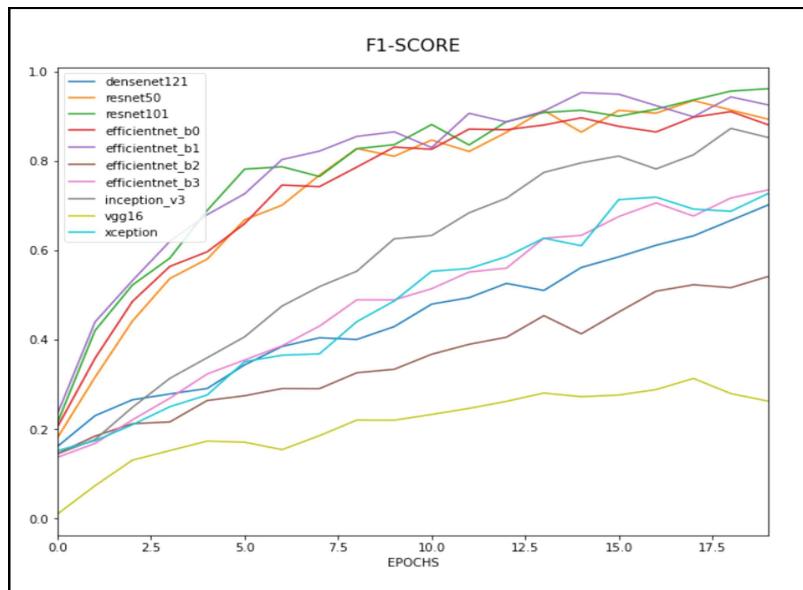
[그림 7] Training on ImageNet



### 3.4 Model Selection

위에서 소개한 3개의 사전학습 모델을 포함하여 다양한 이미지 분류 모델에 우리의 데이터셋을 적합시켜 validation set의 F1 Score를 비교해 본 결과는 [그림 8]과 같다. 약 20번의 학습을 시켰으며 resnet101의 성능이 가장 높았다. 아마도 resnet101의 모형이 깊은 층에서 오버피팅을 막는 효과가 있기 때문에 우리 분류 문제에 가장 적합했다고 판단이 된다. 물론, 다른 모형들의 F1 Score 값이 수렴하지 않았기 때문에 반복수를 더 높여서 실행했으면 다른 모형의 성능이 더 좋았을 수 있지만 사전학습 모델의 학습 이외에도 여러 시도를 해봐야 하기 때문에, 또한 20의 epoch만으로도 1에 가까운 훌륭한 성능을 보이기 때문에 resnet101을 우리의 사전학습 모델로 선정하고 불균형 문제를 해결하기 위해 다양한 시도를 해보았다.

[그림 8] 이미지 분류 모델의 성능 비교



#### 4. 데이터 불균형 처리를 위한 실험 설명

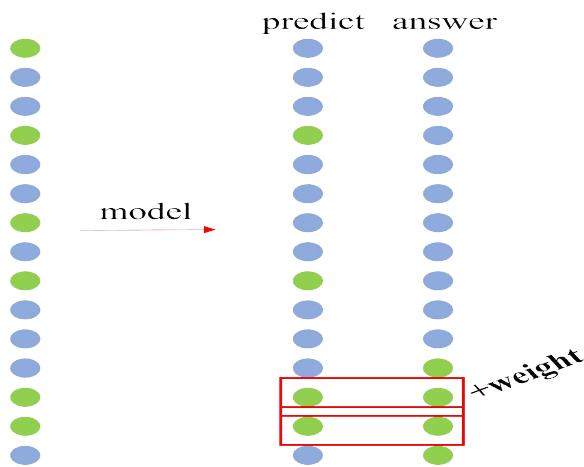
Classification 문제는 주어진 입력 데이터에 대해 해당 데이터의 클래스를 예측하는 문제이다. 분류하고자 하는 데이터 클래스 비율 차이가 너무 크면 (highly-imbalanced data), 단순히 우세한 클래스를 택하는 모형의 정확도가 높아지므로 모형의 성능판별이 어려워진다. 정확도(accuracy)가 높아도 데이터 개수가 적은 클래스의 재현율(recall-rate)이 급격히 작아지는 현상이 발생할 수 있기 때문이다.

이렇듯 분류 및 예측 문제에서 하나의 범주에 속하는 데이터 수가 다른 범주들에 속하는 표본 수에 비하여 현저하게 적을 경우 발생하는 문제를 ‘데이터 불균형 문제(imbalanced data problem)’이라고 한다. 본 대회의 데이터 분포 또한 심각한 불균형이 존재함을 확인하였고, 우리 팀이 이를 해결하기 위해 적용한 다양한 방법에 대해 소개하고자 한다.

##### 4.1 Cost-Sensitive Learning

Cost-Sensitive Learning은 각 레이블 별 가중치를 다르게 주어 학습할 때 영향을 조절하는 방식이다. 가령 수가 적은 레이블을 제대로 예측하였을 때 많은 수의 레이블을 예측하였을 때보다 점수를 더 주어서 학습이 균형적으로 이루어지게끔 하는 방법이다.

[그림 9] Cost-sensitive learning Algorithm



[그림 9]는 Cost-sensitive learning의 알고리즘을 시각적으로 보여준다. 다수의 레이블은 1미만의 가중치를, 그리고 상대적으로 적은 레이블 경우에는 2이상의 가중치를 부여하여 Back-Propagation 과정 진행 시 learning rate에 영향을 주게끔 설정하였다. 즉, 소수의 레이블을 올바르게 학습하여 맞췄을 경우 학습률을 대폭 향상시켜 더욱 학습이 잘되도록 하는 알고리즘 방법을 의미한다. (Elkan, C. 2001)

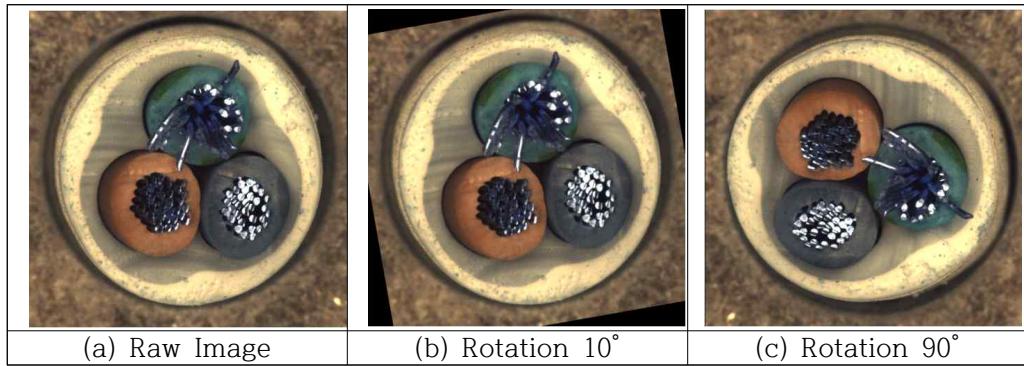
## 4.2 Data Augmentation

Data Augmentation은 이미지에 인위적인 조작을 하여 기존의 데이터의 양을 늘리거나 모델 훈련 시 여러 변수를 추가하여 모델의 학습할 수 있는 여건을 추가로 제공할 수 있는 개념으로 사용된다. Image Recognition 분야에서 주로 사용되는 Data Augmentation을 통해 이미지 불균형 처리를 진행해보았으며, 본 실험에서 Rotation, Affine transformation, Mix up 방법을 적용하였다.

### 4.2.1 Rotation

가장 단순한 Image Augmentation의 방법으로써, 이상 품목에 해당하는 라벨의 이미지를 회전하여 이미지의 개수를 늘리는 일련의 과정을 진행하였다. 각도를 여러 값으로 설정하여 진행하였으며, 이상치에 해당하는 데이터만을 가지고 하여 Train Set을 증가시켰다. 첫 번째의 경우,  $10^\circ$ ,  $350^\circ$ 로 회전시켜 훈련하였고 두 번째의 경우에는  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ 로 회전시켜 훈련을 진행하였다.

[그림 10] Image using Rotation



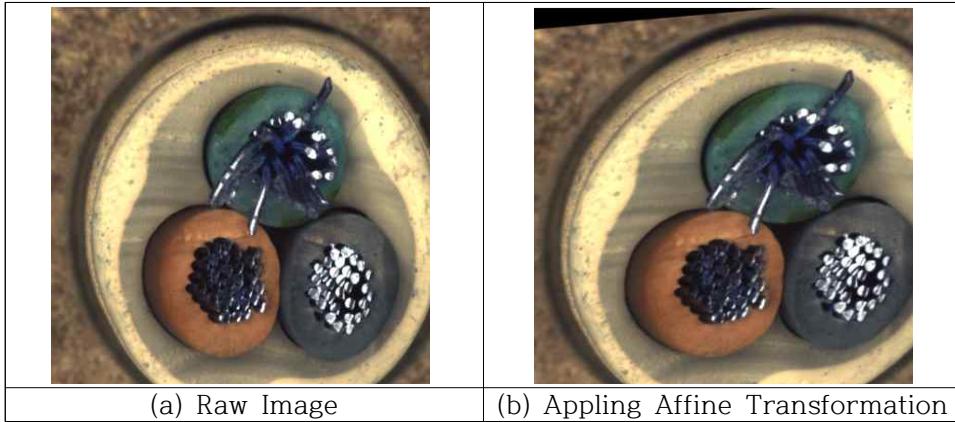
#### 4.2.2 Affine Transformation

다음, Affine Transformation은 적용하였다. Affine Transformation이란, 일차변환이며 이동, 전단, 확대 등을 조합하여 진행한다(Hataya et al., 2020).

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

점 세 개의 이동정보를 알면 행렬로 표시가 가능하며, 해당 연구에서 진행한 아핀 변환을 적용한 결과는 [표 2]와 같다.

[그림 11] Image using Affine Transformation



#### 4.2.3. Mix-up

마지막 Image augmentation으로써, Mix-up을 사용하였다(Cisse et al., 2018). 일반적으로 신경망의 특징은 2가지로 정리해 볼 수 있다. 첫 번째 특징은, 훈련 데이터셋에 대한 평균 예측을 최소함으로써 신경망을 최적화한다는 점이고, 두 번째는 이전에 나왔던 SOTA(State-of-the-art)의 신경망이 훈련 데이터셋의 크기에 선형적으로 비례함으로써 규모가 커진다는 점이다. 이때, 첫 번째 특징을 Empirical Risk Minimization(ERM) principle이라고도 칭한다. 딥러닝 기반의 학습에 ERM을 통해 risk(error)를 최소화를 하게 된다. 그러나, ERM 기반 학습에는 훈련데이터는 강한 규제(regularization) 방법을 사용하더라도 훈련데이터에 과적합되는 현상이 발생한다는 단점이 있다.

이에 대안점으로 나온 것이 Vicinal Risk Minimization(VRM) principle이다. 이는 ERM과 비슷하나, 데이터에서 다르다는 것이다. 우리가 앞서 언급해 온 Augmentation을 의미한다. 그 중 하나가 Mixup인 것이다.

수식은 다음과 같이 정의될 수 있다.

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \tilde{y} = \lambda y_i + (1 - \lambda)y_j$$

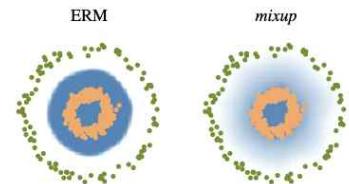
$x_i, y_i$ 는 training feature-target pair이고,  $\tilde{x}, \tilde{y}$ 는 virtual feature-target vector를 의미한다.  $\lambda$  값은 베타분포로부터 추출한 값이며, 가중치로 보면 된다.

[그림 12]는 해당 논문에서 제시한 Mixup의 구현 방식이며, 이를 토대로 Image Augmentation을 진행하였다.

[그림 12] Implement and Effectiveness of Mix-up

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

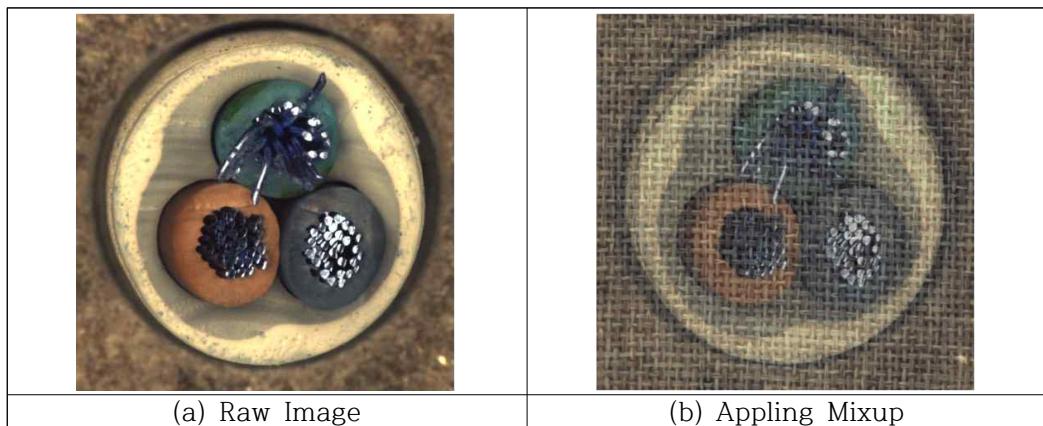
(a) One epoch of mixup training in PyTorch.



(b) Effect of *mixup* ( $\alpha = 1$ ) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates  $p(y = 1|x)$ .

[그림 13]은  $\lambda$ 값을 0.6일 경우, 보여지는 모습을 예시로 나타낸 것이다. 이를 통해 모델은 다양한 형태의 데이터로 학습을 하게 된다.

[그림 13] Image using Mix-up



### 4.3 Under-Sampling

Under-Sampling은 데이터의 수가 많은 클래스의 샘플 수를 축소시키는 방법이다. 다수의 클래스 데이터를 제거함으로써 총 데이터 수가 감소하기 때문에 학습시간이 감소하며, 과적합을 방지할 수 있다는 장점이 있다. 하지만, 많은 데이터가 제

거되면서 정보의 손실이 발생할 수 있으며 다른 방법에 비해 성능이 낮은 경향을 보인다.

Under-Sampling을 수행할 수 있는 방법으로는 Random Sampling, Tomek Links (Elhassan, AT. et al. 2017), CNN Rule(Hart et al., 1968), One Sided Selection(Gustavo et al., 2000) 등의 방법이 존재하며, 본 실험에서는 scikit-learn 라이브러리의 resample() 함수를 이용하여 Random Sampling 기법을 적용하였다.

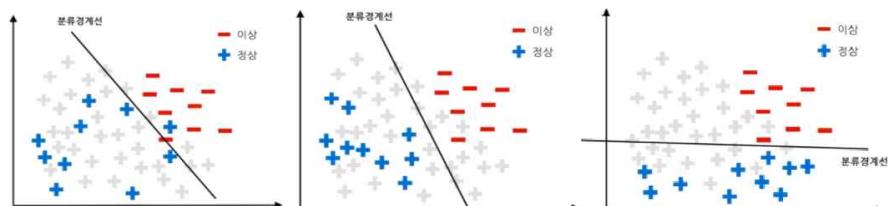
Random sampling은 다수 범주에서 무작위로 샘플링을 수행하며, [그림 14]은 Random Sampling을 3차례 수행한 결과를 시각적으로 보여준다. 무작위로 샘플링을 수행하기 때문에, 실행할 때마다 다른 결과가 얻는다는 단점이 존재한다.

이러한 Random Sampling 기법을 본 팀의 Train dataset에 적용한 결과, 총 3629 개의 정상 데이터가 648개로 감소한 것을 확인할 수 있었다.

#### 4.4 Over-sampling

Over-Sampling은 데이터의 수가 적은 범주의 데이터를 다수 범주의 데이터 수에 맞게 늘리는 샘플링 방식이다. 총 데이터 수가 증가하기 때문에 학습시간이 증가하

[그림 14] Random Sampling 3회 수행 결과

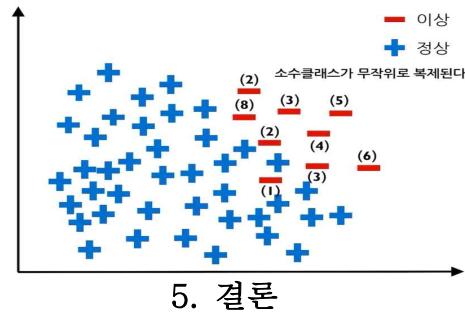


며, 과적합 문제가 발생할 수 있고, 노이즈 또는 이상치에 민감하다는 단점이 있다. 하지만, 제거되는 데이터가 없기 때문에 정보의 손실이 없으며, Under-sampling에 비해 분류 정확도가 높다. 따라서 Over-sampling은 불균형 데이터에 보편적으로 많이 사용하는 기법이다.

Resampling(Burnae, E. et al. 2015), SMOTE(Chawla et al., 2002), Borderline-SMOTE(Han et al., 2005), ADASYN(He et al., 2008) 등의 기법이 있으며, 본 실험에서는 Under-sampling과 동일하게 scikit-learn 라이브러리의 resample() 함수를 이용하여 Resampling 기법을 적용시켰다. Resampling은 [그림

12]와 같이 작동되며, 소수 클래스의 데이터가 무작위로 복제가 되는 방식으로 데이터 수가 증가한다.

[그림 12] Resampling 수행 결과



### 5.1. 실험 결과

실험 결과 성능은 소수의 레이블 또한 예측을 제대로 수행했는지 확인할 수 있는 Macro F-1 score로 평가하였다. Macro F-1 score는 참 값으로 예측한 값 중에서 실제로 참값인 비율을 나타내는 정밀도(Precision)와 실제 참 값 중에서 참 값으로 예측한 비율을 나타내는 재현율(Recall)의 조화평균 값으로 나타낸다.

이 지표는 단순 각각의 레이블의 정답을 정확히 맞췄을 때의 성능을 평가하는 정확도(Accuracy)보다 불균형 데이터에 더 적합하고 널리 사용이 되는 지표이다.

표에서 Up-sampling 기법이 다른 Down-sampling, Cost-sensitive Learning 기법 보다는 성능이 우수하다고 나오며 Augmentation 기법보다는 비교적 더 나은 결과를 나타냈음을 보여준다. 접수가 비교적 높게 나온 2개의 모델(Up-sampling기법, Augmentation 기법)이 다른 2개의 모델보다 더 좋게 나온 이유는 불량 데이터의 수를 더 늘렸기 때문이라고 판단이 된다.

Augmentation 역시 회전을 3개의 방향으로 하여 데이터를 추가하거나 3차원의 방향으로 이미지를 뒤트는 affine기법, 복수의 이미지를 섞는 mix-up 기법 등 다양한 시도를 해본 결과 이미지를 회전하고 affine을 적용했던 모형이 성능이 가장 좋았다. 결점은 찾는 이상치 탐지이다 보니 여러 방향에서 관찰할 수 있어서 성능이 좋았다고 판단하였다.

[표 1] 실험 결과

Method	Hyperparameter			BEST F1_Score
	Batch_size	Optimize_r	Learning_rate	
Up-Sampling	16,32,64,128	Adam, RMSprop Adagrad	1e-4, 1e-3, 5e-3, 7e-3	<b>0.732</b> (32/Adam/1e-3)
Down-Sampling	16,32,64,128	Adam, RMSprop Adagrad	1e-4, 1e-3, 5e-3, 7e-3	0.646 (32/Adam/1e-3)
Data augmentation	affine Rotation affine + Rotation Mix-up	16,32,64,128 32,64,128 16,32,64 32,64,128	Adam Adam Adam Adam	1e-3 1e-3 1e-3 1e-3 0.725 (affin+rotation/ 16/Adam/1e-3)
Cost-Sensitive Learning	16,32,64	Adam	7e-3	0.51 (16,Adam,7e-3)

## 5.2 한계점 및 향후 연구방향

먼저, 불균형 데이터를 처리할 때. State를 Good과 Bad인 Binary 형태로 나누어 불균형 처리를 했다는 점이 한계점이다. 실제 데이터의 클래스는 15개의 사물의 종류와 결합하여 총 30개의 클래스이기 때문에, Binary class가 아닌 30개의 클래스를 기준으로 불균형 데이터 처리 방법을 적용하였다면 보다 성능이 높은 결과를 얻을 수 있었을 것이다.

또한, 불균형 데이터를 처리하는 방법으로 제시했던 기법들을 더 다양하게 적용 시켰다면 더 좋았을 것 같다. Under-sampling에서 Tomek Links, CNN Rule, One Sided Selection 등의 방법과 Over-sampling에서 SMOTE, Borderline-SMOTE, ADYSYN 등의 기법을 좀 더 다양하게 적용시켰다면 보다 정밀한 결과를 얻을 수 있을 것이다.

데이터를 처리하는 방법 이외에 모델 측면에서, 성능을 향상시키는 방법으로 유명한 ‘앙상블’을 활용하여 모델을 업그레이드 시킨다면 업그레이드된 모델을 만들 수 것이다. 추후 연구에서는 앞서 언급한 개선 방법을 모두 활용하여 보다 높은 성능의 모델이 만들어질 것으로 기대한다.

## 참고문헌

- [1] Cho, J., Choi, E., Kim, J., Oh, M. & Roh, K. (2021) A Study on the Design of Supervised and Unsupervised Learning Models for Fault and Anomaly Detection in Manufacturing Facilities. *Korea Bigdata Society*, 6(1), pp. 23–35.
- [2] Tan, M & Le, Q. (2020) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning(ICML)*.
- [3] Chollet, F. (2017) Xception : Deep Learning with Depthwise Separable Convolutions. *Computer Vision and Pattern Recognition(CVPR)*. pp. 1800–1807.
- [4] Simonyan, K., & Zisserman, A. (2015) Very Deep Convolutional Networks For Large-Scale Image Recognition. *The International Conference on Learning Representations(ICLR)*.
- [9] Sepp H. (1998) The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 6(2), 107–116.
- [7] He, K., Ren, S., Sun, J. & Zhang, X. (2016) Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. pp. 770–778.
- [4] Elkan, C. (2001) The Foundations of Cost-Sensitive Learning. *International Joint Conference on Artificial Intelligence(IJCAT)*.
- [6] Hataya, R., Jan, Z., Nakayama, H. & Yoshizoe, K. (2020) Faster AutoAugment : Learning Augmentation Strategies Using Back-propagation. *European Conference on Computer Vision(ECCV)*. pp. 1–16.
- [9] Cisse, M., Dauphin, Y. N., Lopez-paz, D. and Zhang, H. (2018) mixup : Beyond Empirical Risk Minimization. *The International Conference on Learning Representations(ICLR)*.
- [10] Elhassan, AT. et al. (2017) Classification of Imbalance Data Using Tomek Link(T-Link) Combined with Random Under-sampling

(RUS) as a Data Reduction Method. Global Journal of Technology & Optimization.

- [11] Hart, P. et al. (1968) The condensed nearest neighbor rule. *Institute of Electrical and Electronics Engineers(IEEE)*, 14(3), 515–516.
- [12] Gustavo, E., B., Andre, D. C., & Maria-Carolina, M. (2000) Applying One-Sided Selection to Unbalanced Datasets. Mexican International Conference on Artificial Intelligence(MICAI).
- [13] Burnaev, E., Erofeev, P., Papanov, A. (2015) Influence of Resampling on Accuracy of Imbalanced Classification. Eighth International Conference on Machine Vision.
- [14] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002) SMOTE : Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 6, 321-357.
- [15] Han, H., Wang, W, Y. & Mao, B. H. (2005) Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning. *International Conference on Intelligent Computing(ICIC)*. pp. 878–887.
- [16] He, H., Bai, Y., Garcia, E, A. & Li, S. (2008) ADASYN : Adaptive synthetic sampling approach for imbalanced learning. *Institute of Electrical and Electronics Engineers(IEEE)*.

## Leaf area forecasting Using CNN

임성호<sup>1)</sup>, 류재욱<sup>2)</sup>, 이경서<sup>1)</sup>, 강진모<sup>1)</sup>

### Abstract

Recently, as climate change gets worse, various difficulties have arisen in growing crops. crop cultivation is very closely related to the food problem which is directly related to human life. And at the same time, it is a very sensitive and economically influential field such as food security in each country.

Therefore, it is obvious that it is an area where many governments and companies continue to expand and invest and challenge. various tools are being used to solve this problem, and AI is the most popular solution among them.

Our goal is to derive the optimal environment for efficient growth of crops based on AI. In this study, we compare the accuracy and prediction performance of deep learning algorithms for leaf area prediction.

---

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

2) All authors have equal contribution, the School of Mechanical Engineering, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

### 1.1. 연구배경 및 목적

최근 전 세계가 공통으로 고민하는 문제에는 기후변화가 빠질 수 없다. 무분별한 개발로 환경이 오염되면서 기후가 기존과 다르게 빠르게 변화함에 따라 농작물의 재배에도 많은 애로사항이 발생하고 있다.

농업 분야에서 발생하는 이러한 문제점을 해결하고자 스마트 팜 등 IT 기술을 사용하여 더욱 효율적으로 작물을 재배하고자 하는 노력이 계속되고 있다. 특히, AI를 기반으로 하여 작물 생육에 필요한 최적 환경을 도출하려는 시도들이 최근 큰 화두가 되고 있다. 농작물 재배는 인류에게는 없어서는 안 될 필수 소비재임과 동시에 각국의 식량안보 등 매우 민감하면서도 경제적 영향력이 매우 큰 분야이기에 많은 정부와 기업에서 계속해서 확대 투자하고 도전하는 영역임이 자명하다.

본 연구에서는 KIST에서 주최하고 데이콘(DACON)에서 운영하는 ‘생육 환경 최적화 경진대회’에 참여하여, 청경채 사진과 환경 데이터를 활용하여 다음 날 청경채의 일 면적을 예측하는 딥러닝 모형을 구축한다. 본 대회의 목표는 식물 생육 데이터와 환경 데이터를 활용하여 어떠한 상황에서 생육의 최적 환경이 도출되는지를 알기 위한 딥러닝 기반 알고리즘 개발이라고 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 대회에서 제공하는 데이터 구성과 그에 대한 전처리 과정에 대해 설명한다. 3장에서는 생육 환경을 최적화하기 위한 사용할 연구 모형을 제시한다. 4장에서는 모형의 성능평가 환경 및 기준에 대해 알아보고, 기본 모형과의 성능 차이를 비교하였다. 5장에서는 본 분석의 한계점과 개선 방안에 대해 논의한다.

## 2. 대회 소개 및 데이터 설명

### 2.1 데이터 구성

“생육 환경 최적화 경진대회”는 KIST에서 주최/주관하며 데이콘에서 운영하는 대회로, 주제는 청경채 사진과 환경 데이터를 활용한 일면적 예측 알고리즘을 개발하는 것이다. 본 대회는 학습용 데이터와 테스트용 데이터를 제공한다. 학습용 데

이터는 75개의 케이스로 구분되어 있으며 각 케이스에는 입력 데이터인 이미지 데이터, 환경 데이터와 출력 데이터인 레이블로 구성되어 있다. 이미지 데이터는 청경채를 찍은 사진이며, 환경 데이터는 촬영 된 시각으로부터 1일간 1분 간격으로 측정된 내/외부 온도, 습도, CO<sub>2</sub>, EC, LED 색깔별 동작 강도 등 총 18가지의 환경 정보가 저장된 데이터이다. 레이블은 해당 이미지가 촬영된 시점으로부터 1일 후의 일 면적 중량이다. 테스트용 데이터는 학습용 데이터와 동일하게 이미지 데이터와 환경 데이터로 구성되어 있다. 학습용 데이터는 총 1592개의 샘플, 테스트용 데이터는 460개의 샘플로 이루어져 있다.

따라서 본 대회는 1일 전 이미지 데이터와 1일간의 환경 데이터를 활용해 1일 후 청경채의 중량을 예측하는 모형을 만들어야 한다. 하지만 본 연구팀에서 이미지 데이터와 환경 데이터를 따로 간단한 모델을 통해 훈련해본 결과, 환경 데이터에 따른 예측 오차가 너무 커서 환경 데이터 없이 이미지 데이터만을 활용해 최종 예측 모형을 선정하였다.

## 2.2 데이터 전처리

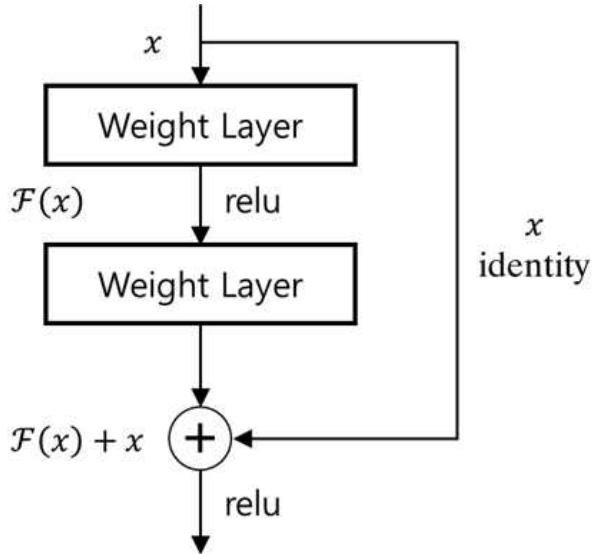
제공된 이미지 데이터의 크기가 크고 개수가 적기 때문에 크기를 줄이고 무작위로 변형하는 전처리 과정을 진행하였다. 먼저, 오류가 있는 사진을 훈련에서 제외시켰으며 이미지 파일을 RGB 값으로 저장하였다. 이후, 기존 이미지 크기인 2464×3280에서 224×224로 크기를 축소시키고 정규화를 했으며 무작위로 상하좌우 대칭, 회전, 밝기 및 대비를 변형하여 이미지를 랜덤하게 만들었다.

### 3. 딥러닝 모형

#### 3.1 SE(Squeeze and Excitation)-Resnet

SE-Resnet은 Resnet의 구조에 SENet의 SE block이 적용된 모형이다. Resnet(Residual neural network)은 [그림 1]과 같이 기본적인 CNN 구조에서 입력  $x$ 를 출력  $F(x)$ 에 바로 연결시켜 더해주는 Skip connection을 추가한 구조이다 (He et al., 2015). 본 연구에서는 Resnet을 사용하여 감정 예측의 정확도를 높이고자 하였다. Resnet 이전의 신경망은 20개의 층 이상으로 깊어지면 Gradient vanishing, exploding 문제나 Degradation 문제가 발생하여 오히려 성능이 감소하는 결과를 보였다. 하지만 Skip connection을 추가한 Resnet의 경우, 152개의 층 까지 성능 향상을 보였으며, 그 결과 2015년도 ILSVRC 대회와 그 외 다수의 대회에서 1등을 차지하였다.

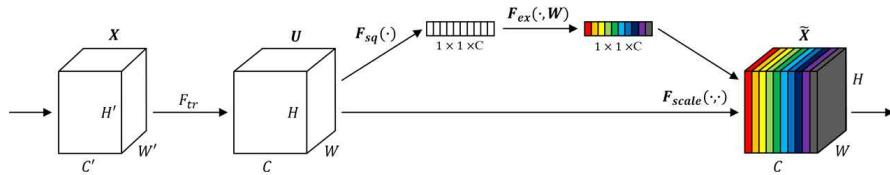
[그림 1] Resnet 구조



SENet은 [그림 2]와 같이 압축(Squeeze)과 자극(Excitation)으로 구성된 SE Block을 활용한 신경망으로, 중요한 특징은 추출하고 덜 중요한 특징은 억제하는 방식을 통해 좋은 성능을 만들어낸다(Hu et al., 2018). 먼저, 압축 과정에서는 각

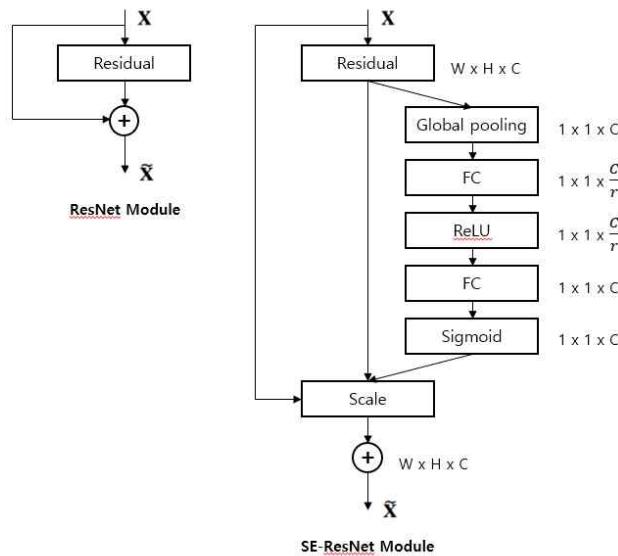
채널을 Global Average Pooling을 이용해 채널 전체 정보를 대표하는 1차원 스칼라 값으로 압축시킨다. 이후, 자극 과정에서는 채널 간 의존성을 고려하여 중요한 특징이 잘 추출되도록 채널 별 가중치를 재조정한다. SE Block은 다른 신경망에 적용하기가 간편하며, 매개변수의 증가량에 비해 성능 향상이 매우 뛰어나다는 장점이 있다. 그 결과 2017년도 ILSVRC 대회에서 1등을 차지하였다.

[그림 2] SENet 구조



Resnet과 SE Block을 결합한 SE-Resnet의 구조는 [그림 3]과 같다. Resnet50과 SE-Resnet50의 비교를 통해 SE-Resnet의 성능 향상을 확인할 수 있었다. [표 1]과 같이, SE-Resnet50은 기존의 Resnet50에 비해 연산량(GFLOPs)이 3.86에서 3.87로 0.26% 밖에 증가하지 않았지만 모형의 top-5 에러는 7.48에서 6.62로 11.5% 감소했으며, 심지어 더 깊은 모형인 Resnet101의 top-5 에러 6.52와 비슷한 것을 알 수 있다(Mazzeo et al., 2020).

[그림 3] SE-Resnet 구조



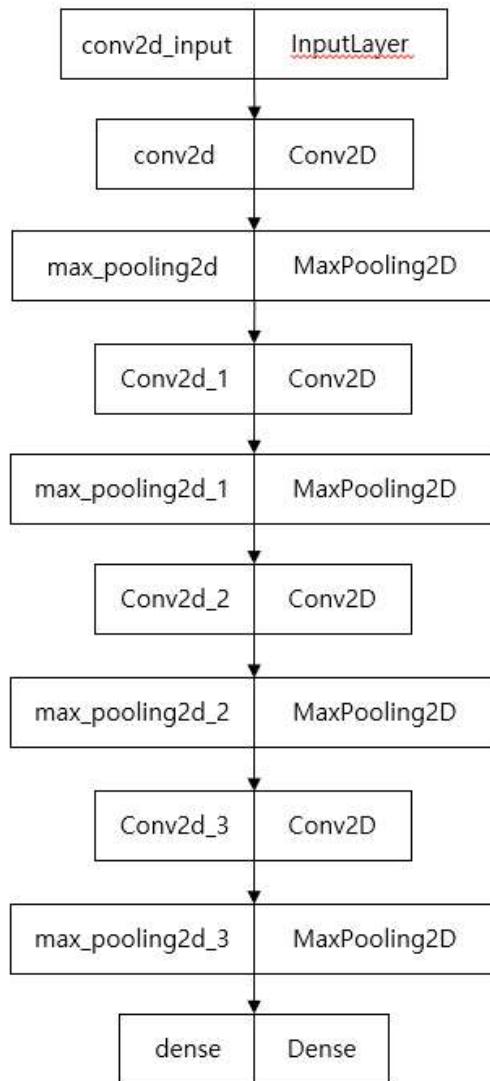
[표 1] Resnet50과 SE-Resnet50 비교

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [10]	24.7	7.8	24.80	7.48	3.86	23.29 <sub>(1.51)</sub>	6.62 <sub>(0.86)</sub>	3.87
ResNet-101 [10]	23.6	7.1	23.17	6.52	7.58	22.38 <sub>(0.79)</sub>	6.07 <sub>(0.45)</sub>	7.60
ResNet-152 [10]	23.0	6.7	22.42	6.34	11.30	21.57 <sub>(0.85)</sub>	5.73 <sub>(0.61)</sub>	11.32
ResNeXt-50 [47]	22.2	-	22.11	5.90	4.24	21.10 <sub>(1.01)</sub>	5.49 <sub>(0.41)</sub>	4.25
ResNeXt-101 [47]	21.2	5.6	21.18	5.57	7.99	20.70 <sub>(0.48)</sub>	5.01 <sub>(0.56)</sub>	8.00
VGG-16 [39]	-	-	27.02	8.81	15.47	25.22 <sub>(1.80)</sub>	7.70 <sub>(1.11)</sub>	15.48
BN-Inception [16]	25.2	7.82	25.38	7.89	2.03	24.23 <sub>(1.15)</sub>	7.14 <sub>(0.75)</sub>	2.04
Inception-ResNet-v2 [42]	19.9	4.9	20.37	5.21	11.75	19.80 <sub>(0.57)</sub>	4.79 <sub>(0.42)</sub>	11.76

### 3.2 SimpleCNN (Conv2D)

SimpleCNN은 Conv2D를 이용해 만든 기본적인 CNN 구조의 모형이며, 본 연구에서 사용한 SE-Resnet50과의 성능 비교를 위해 사용될 모형이다. CNN은 2012년 ImageNet 대회에서 AlexNet이 우승하면서부터 주목받기 시작했다(Krizhevsky et al., 2012). SimpleCNN은 [그림 4]와 같이 Conv2D와 Max Pooling을 4번 반복한 단순한 구조로 되어있다.

[그림 4] SimpleCNN 구조



## 4. 실험 및 결과

### 4.1 평가 기준 및 실험환경

본 대회의 평가 기준은 수식 (1)의 정규화된 평균절대오차(Normalized Mean Absolute Error)를 따른다.

$$NMAE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right| \quad (1)$$

이때  $n$ 은 테스트용 데이터 샘플 수,  $y_i$ 는 실제값,  $y'_i$ 는 예측값을 의미한다.

Optimizer는 Adam, Loss function은 MAE, epoch은 80, batch size는 32로 설정하고 1592개의 연습용 데이터 중 10%인 159개의 데이터를 검증용 데이터로 사용하여 훈련을 진행하였다.

코드를 검증한 환경은 NVIDIA GeForce RTX 3090 / Ubuntu 18.04.6 LTS (64bit)이며, 프로그램은 Visual Studio Code를 사용하였다.

### 4.2 실험 결과

SE-Resnet50 모형으로 테스트한 평가 결과 점수와 Conv2D 모형으로 테스트한 평가 결과 점수는 [표 2]와 같다. SE-Resnet50의 NMAE 값은 0.15843이고, SimpleCNN의 NMAE 값은 0.31105로, SE-Resnet50의 오차가 SimpleCNN에 비해 약 49 % 감소하였다.

[표 2] SE-Resnet50과 SimpleCNN 실험 결과

	Train_MAE	Val_MAE	NMAE
SE-Resnet50	1.1497	0.8229	0.15843
SimpleCNN	15.2445	9.5188	0.31105

## 5. 결론

우리 데이터 분석 결과는 SE-Resnet50의 성능이 Simple CNN보다 월등히 좋았음을 알 수 있다. 이는 레이어의 중간에 se block과 residual block을 추가했기 때문으로 보인다. SE-Resnet50 모델을 사용하여 train set과 validation set을 통해 학습 후 평가한 결과 [표 2]에서 확인 할 수 있듯이 test NMAE가 Simple CNN의 경우 0.31105, SE-Resnet50의 경우 0.15843으로 약 50% 정도로 loss가 감소했음을 알 수 있다. 처음에는 이미지의 크기를 바꾸지 않고  $2464 \times 3280$ 로 사용했으나 결과가 좋지 않아 여러 사이즈로 변경해가며 학습해보았고  $224 \times 224$  사이즈가 가장 좋은 결과를 냄을 확인할 수 있었다. 이 결과를 대회에 제출하였을 때 6월 16일 기준으로 25위를 기록하였다.

대회에서는 이미지 데이터와 환경 데이터 즉, 정형 데이터와 비정형 데이터를 모두 제공했음에도 불구하고 이미지 데이터만을 활용하여 예측을 했을 때 더 좋은 결과를 기록했다. 이는 데이터 자체에 결측치가 많은 것이 가장 큰 원인이다. 만약 전처리에 충분한 시간을 쏟고, 이를 통해 둘 모두를 아우르는 모형을 구축했다면 더 좋은 결과를 냈을 것으로 보인다.

## 참고문헌

- [1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- [2] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141.
- [3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105.
- [4] Mazzeo, P. L., Libetta, C., Spagnolo, P., & Distante, C. (2020). A Siamese Neural Network for Non-Invasive Baggage Re-Identification. *Journal of Imaging*, 6(11), 126.

데이터 과학 연구 제11권

2022년 pp.75-84

## US Patent Phrase to Phrase Matching Using DeBERTa V2 Model

최주원<sup>1)</sup>, 박지원<sup>2)</sup>, 지평진<sup>2)</sup>, 강태인<sup>2)</sup>

### Abstract

According to U.S patent and trademark office, the number of patents is increasing. It is hard to find similarities among old patents and new patents by hand. So we built a transformer model to find similarities by machine. For this, We used data from the U.S patent and trademark office. In this work, we compare the correlation of recent transformer models like BERT and DeBERTa V2 to find an effective model for patent similarity.

---

1) All authors have equal contribution, Department of Artificial Intelligence,  
Chung-Ang University, Seoul 06974, Korea

2) All authors have equal contribution, Department of Applied Statistics, Chung-Ang  
University, Seoul 06974, Korea

## 1. 서론

개인 혹은 기업의 지적 재산권의 가치는 점점 커지고 있다. 개인과 기업은 본인 이 소유하고 있는 지적 재산권의 보호를 위해 특허를 등록하고 있고, 그 숫자는 매년 늘어가고 있는 상황이다. U.S Patent and Trademark Office(USTPO. 2021)는 과학, 공학, 상업 정보에 대하여 세계에서 가장 큰 정보를 제공하는 곳이다. USTPO의 자료에 따르면, 지적 재산권 형태의 특허가 대략 11,000,000개가 저장되어 있고, 추가적으로 USTPO에 등록되는 1년 특허의 수는 1963년 90,982개에서 2020년 646,244개로 6배가 넘게 증가하였다.

특허 등록이 폭발적으로 증가한 상황 속에서, 기존의 특허와 새롭게 등록되는 특허의 유사성을 사람이 판단하는 것은 매우 힘든 일이다. 많은 시간과 인력을 쏟아부어야 하기 때문이다.

본 연구에서는 Kaggle에서 주최한 U.S Patent Phrase to Phrase Matching 대회의 데이터를 이용하여 기준 특허에 대한 신규 등록 특허의 유사도를 점검하는 모델을 제안하고자 한다. 주된 관심이 되는 모델은 DeBERTa V2로서, 자연어 처리 분야에서 State Of The Art를 기록한 BERT를 기반으로 발전시켜온 모델이다(He et al., 2020, Devlin et al., 2018).

본 논문의 구성은 다음과 같다. 2장에서는 연구에서 사용한 각 모델에 대한 상세한 설명을 기술하고 3장에서는 대회에서 사용한 데이터의 구조와 모델을 효율적으로 제작하기 위한 전략을 기술한 후, 4장에서 2장의 모델을 이용한 결과를 비교하고, 5장에서 본 분석에 대한 기술과 한계점에 대해 논의할 것이다.

## 2. 모델 설명

### 2.1. BERT

BERT는 Transformer의 구조를 거의 동일하게 사용하고 있지만(Devlin et al., 2018; Vaswani et al., 2017), Transformer에서 Decoder 부분을 제거하고 Encoder 부분만 사용하는 특징이 있다. Transformer처럼 encoder-decoder 구조를 사용해서 input이 있을 때, 특정한 output을 만들어내는 task를 하는 모델이라 기보다는, 대규모 말뭉치 학습을 통해 단어의 분산 표현을 보다 더 잘 담을 수 있

는 모델이다. 잘 학습된 단어 분산 표현을 통해 여러가지 새로운 task에 맞게 fine-tuning해서 사용할 수 있는 특징이 있다.

BERT의 input representation은 세가지 embedding인 Token Embedding, Position Embedding, Segment Embeddings 값의 합으로 구성이 된다. Token Embedding은 tokenized된 sequence A와 B가 입력으로 들어오면 input token들을 임베딩 시켜준다. Position Embedding에서는 position embedding이 없다면, 어느 단어가 앞이고 어느 단어가 뒤인지 transformer 구조에서는 알 수 없기 때문에 위치 정보를 주기 위해 넣는다. Segment Embeddings에서는 position embedding을 보면 숫자가 0-10까지 들어가는데 이것으로는 두 개의 문장이 어디서부터 어디까지인지 알 수 없다. 따라서 AAAA와 BBBB처럼 두 개의 문장을 구별하는 역할을 한다.

Bert는 Tokenizer로 BPE(Byte Pair Encoding) 알고리즘 기반 Wordpiece Tokenizer를 사용하며, 말뭉치에서 자주 등장한 문자열을 토큰으로 인식한다는 점에서 BPE와 본질적으로 유사하다(Wu et al., 2016; Sennrich et al., 2015). 다만 어휘 집합을 구축할 때 문자열을 병합하는 기준이 다르다. Wordpiece는 BPE처럼 단순히 빈도를 기준으로 병합하는 것이 아니라, 병합했을 때 말뭉치의 우도(likelihood)를 가장 높이는 쌍을 병합한다.

Bert는 크게 Pre-training, Fine-tuning 단계로 구분한다. Pre-training 단계에서는 labeling 하지 않은 데이터를 기반으로 학습을 진행한다. Fine-tuning 단계에서는 pre-trained된 파라미터로 초기화되고, 이 후 모델을 labeling된 데이터로 fine tuning을 한다. 실제 task에서의 모델은 초기에는 동일한 파라미터로 시작하지만, 최종적으로는 서로 다른 fine-tuned된 모델을 갖게 된다. Bert는 pre-trained 된 모델과 fine-tuned된 모델 사이의 구조적 차이가 거의 없게 된다.

## 2.2. DeBERTa

DeBERTa는 2가지 새로운 메커니즘인 Disentangled Attention와 Enhanced mask decoder을 활용하여 BERT와 RoBERTa 모델을 발전시킨 트랜스포머 기반 신경 언어 모델이다(Liu et al., 2019). 먼저, Disentangled Attention은 Input Layer의 각 단어가 Word(Content) Embedding과 Position Embedding의 합인 Vector를 사용하여 표현되는 BERT와 달리, DeBERTa의 각 단어는 각각 Content과 Position를 인코딩하는 두 개의 Vector를 사용하여 표현되며 단어 사이의 Attention Weight는 각 Content과 Relative Positions를 기반한 Disentangled Matrices를 사용하여 연산된다. 다음으로 Enhanced mask decoder는 BERT와 마-

찬가지로, DeBERTa는 Context Word의 Content과 Position Information를 활용한 Masked Language Modeling(MLM)을 사용하여 사전 학습된다. Disentangled Attention 메커니즘은 이미 Context Word의 Content과 Relative Position를 고려하지만, 단어의 Absolute Position은 고려하지 않는다. 한 문장에서 두 단어의 Local Context은 비슷하지만, 문장에서 다른 구문적 역할을 한다. 이러한 구문적 뉘앙스는 문장에서 단어의 Absolute Position에 크게 의존하므로 언어 모델링 과정에서 단어의 Absolute Position을 설명하는 것이 중요하다. DeBERTa는 Softmax Layer 직전에 Absolute Word Embedding을 통합하고, 단어 Content 및 Position의 Aggregated Context Embedding을 기반으로 마스크된 단어를 디코딩한다. 또한, DeBERTa는 사전학습 언어모델을 다운스트림 NLP 작업에 맞게 Fine-Tuning하기 위한 새로운 가상 적대적 훈련 방법을 제안한다. 이 방법은 모델의 일반화를 개선하는 데 효과적이다.

### 2.3. DeBERTa V2

DeBERTa V2 모델은 DeBERTa 모델을 발전시켜 만든 모델이다. 128K의 새로운 Vocabulary size를 training data로부터 구축했으며 기존 DeBERTa의 GPT2 Tokenizer 대신 Sentencepiece Tokenizer를 사용했다. Input token들의 지역적 의존성을 더 잘 학습시키기 위해서 첫 번째 transformer layer에 추가적인 convolution layer를 추가했다. 또한 position projection matrix와 attention layer 안의 content projection matrix를 공유하여 성능은 유지하되, parameter 수를 줄였다. 마지막으로 T5 모델과 유사하게 relative position을 인코딩하기 위해 log bucket을 사용했다.

## 3. 데이터 설명과 모델 전략

### 3.1. 대회 소개

본 대회는 Kaggle의 U.S. Patent Phrase to Phrase Matching 대회이며 특히 문서의 핵심 Phrase들을 일치시켜 관련 정보를 추출하기 위해 새로운 의미론적 유사성 데이터 세트에 대한 모델을 개발하는 것이 목적이다. Phrases 사이의 의미론적 유사성을 찾아내는 것은 특허가 이전에 개발되었는지 여부를 알아보기 위해 특

허 검색 및 심사에서 매우 중요한 과정이다. 가령, 한 특허 심사 문서에 "television set"가 핵심 Phrase일 때, 이전 특허 간행물 중, "TV set"가 기술되어 있는 경우, 모델은 이들이 동일한 것임을 인식하고 변리사 혹은 심사위원이 관련 문서를 검색하고 참고하는 데에 도움을 줄 것이다.

한 특허 신청 문서에 "strong material"이 핵심 Phrase이고 기존 문서에 "steel"이 핵심 Phrase로 사용되는 경우 동일한 것으로 인식할 수 있지만, "strong material"이 다른 도메인에서는 "steel"과는 전혀 다른 것으로 생각 될 수 있기 때문에 이러한 상황을 명확히 하기 위해서 데이터에 도메인 분류와 외부 데이터를 적절히 활용해 모델에 추가 정보를 제공했다.

### 3.2 데이터 구조

U.S. Patent Phrase to Phrase Matching 대회에서 제공하는 데이터는 train, test 데이터이며 train 데이터는 [표 1]와 같고, test 데이터는 train과 배타적인 데이터에 score column만 없는 구조이다. train 데이터로 학습 후에 test 데이터의 score를 채워 submission으로 제출한다.

[표 1] train data

	<b>id</b>	<b>anchor</b>	<b>target</b>	<b>context</b>	<b>score</b>
1	37d61fd22 72659b1	abatement	abatement of pollution	A47	0.5
2	7b9652b1 7b68b7a4	abatement	act of abating	A47	0.75
...	...	...	...	...	...
36471	756ec035 e694722b	wood article	wooden material	B44	0.75
36472	8d135da0 b55b8c88	wood article	wooden substrate	B44	0.75

참가자는 anchor phrase와 target phrase간의 유사도인 score를 예측하는 모델을 만들어야하고 phrase의 도메인 분류인 context를 활용할 수 있다. 도메인 분류를 활용하기 위해 Cooperative Patent Classification Codes Meaning 데이터, 일명 CPC Code Meaning 데이터를 추가적인 데이터로 활용했다. train 데이터의 context와 CPC Code Meaning 데이터의 Code와 결합하여 도메인 분류 키워드를 생성하고, anchor, target, keyword를 구분자와 함께 결합하여 언어 모델의 raw input으로 하였다. 최종 train 데이터는 [표 2]와 같다.

[표 2] 최종 train data

<b>id</b>	<b>anchor</b>	<b>target</b>	<b>context</b>	<b>score</b>	<b>text</b>
37d61fd22		abatement			abatement [SEP]abat
72659b1	abatement	of	A47	0.5	ement of pollution[ SEP]huma
		pollution			n...
7b9652b1		act of			abatement [SEP]act
7b68b7a4	abatement	abating	A47	0.75	of abating[S EP]human
					necessi...
...	...	...	...	...	...
756ec035	wood	wooden			wood
e694722b	article	material	B44	0.75	article[SE P]wooden
					material[S EP]perfor
					min...
8d135da0	wood	wooden			wood
b55b8c88	article	substrate	B44	0.75	article[SE P]wooden
					substrate[ SEP]perfo
					rmi...

### 3.3 모델 전략

본 대회에 우리가 사용한 사전 학습 언어 모델은 BERT와 DeBERTa V2 이다. 먼저, 우리는 본 대회에 최적화된 모델인 Bert For Patents을 활용하였다(Lee et al., 2019). Bert For Patents 모델은 Hidden Layers는 24, Hidden size는 1024, Attention Head는 16, Activation Function은 GELU로 설정하였다(Hendrycks et al., 2016). Tokenizer는 Wordpiece를 사용하였다. Bert For Patents 모델을 기반으로 매개변수를 조정하여 비교실험을 진행하였다(Lee et al., 2019). 모든 실험에서는 Batch Size 64, Epoch 5로 동일하게 설정하였다. 실험 1에서는 Learning Rate와 Weight Decay를 각각  $1e-5$ , 0.1로 설정하였으며, 실험 2에서는  $3e-5$ , 0.1로 설정하였으며, 실험 3에서는  $5e-5$ , 0.1로 설정하였다. Bert For Patents 모델 실험 결과는 [표 3]과 같다.

[표 3] Bert For Patents 모델 실험 결과

	Learning Rate	Weight Decay	Batch Size	Epoch	Result
1	$1e-5$	0.1	64	5	0.7877
2	$3e-5$	0.1	64	5	0.7801
3	$5e-5$	0.1	64	5	0.7762

다음으로 우리는 DeBERTa V2 Large 모델 기반으로 매개변수를 조정하여 비교실험을 진행하였다. Hidden Layers는 24, Hidden size는 1024, Attention Head는 16, Activation Function은 GELU로 설정하였다. Tokenizer는 Sentencepiece를 사용하였다. 또한 보다 일반화된 모델을 생성하기 위해 전체 데이터를 K번 나누어 교차검증하는 방식인 K-Fold Cross Validation을 적용하였다. 이에 데이터의 전 범위를 학습하고 검증 데이터로 성능을 평가하므로써 학습 데이터에만 과적합이 되는 것을 방지하였다. 모든 실험에서는 Bert For Patents 모델과 같이 Batch Size 64, Epoch 5로 동일하게 설정하였다. 실험 1에서는 Learning Rate, Weight

Decay와 K-Fold Cross Validation를 각각 2e-5, 0.1, 5로 설정하였으며, 실험 2에서는 2.5e-5, 0.1, 5로 설정하였으며, 실험 3에서는 3e-5, 0.1, 4로 설정하였다. DeBERTa V2 모델 실험 결과는 [표 4]과 같다.

[표 4] DeBERTa V2 모델 실험 결과

	Learning Rate	Weight Decay	K-Fold	Batch Size	Epoch
1	2e-5	0.1	4	64	5
2	2.5e-5	0.1	5	64	5
3	3e-5	0.1	4	64	5

### 3.4. 평가 지표

본 대회의 평가 지표로는 Pearson Correlation Coefficient을 사용하였다 (Ahlgren et al., 2003). Pearson Correlation Coefficient의 식은 (1)과 같다.

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

여기서  $r$ 은  $x$ ,  $y$ 의 Correlation Coefficient이다.  $n$ 은 Sample size이고,  $x_i$ 는  $x$ 의 변수의  $i$ 번째 Value를 나타내고,  $y_i$ 는  $y$ 의 변수의  $i$ 번째 Value를 나타낸다. 또한  $\bar{x}$ 는  $x$  변수의 평균이며,  $\bar{y}$ 는  $y$  변수의 평균이다.

## 4. 실험 결과

### 4.1. 실험 결과

Bert For Patent 모델을 사용한 결과보다 DeBERTa V2 모델을 사용한 결과가 더 높게 나타났다. [표3]과 같이, Bert For Patent에서 Learning Rate와 Weight Decay를 각각  $1e-5$ , 0.1로 설정한 실험 1 모델이 가장 높은 성능 0.7877을 기록하였다. 이어서 실험 2의 결과는 0.7801, 실험 3의 결과는 0.7762를 기록하였다. 또한 [표4]에서 보듯이, DeBERTa V2를 사용한 모델은 Learning Rate, Weight Decay, K-Fold Cross Validation를 각각  $3e-5$ , 0.1, 4로 설정한 실험 3의 결과 0.8393을 기록하여 최고 성능을 나타냈다. 실험 1의 결과는 0.8392, 실험 2의 결과는 0.8376를 기록하였다. 본 대회의 실험 결과 DeBERTa V2를 사용한 모델이 Bert For Patent를 사용한 모델보다 0.0516만큼 더 높은 성능을 기록했다.

## 5. 결론

우리는 기존 특허에 대한 신규 등록 특허의 유사도를 점검하는 모델로 DeBERTa에서 발전시켜 만든 DeBERTa V2을 사용하였다. Sentencepiece Tokenizer를 사용하는 DeBERTa V2 모델은 BERT보다는 더 좋은 성능을 나타냄을 실험과정을 통해 알 수 있다.

우리 팀의 대회 스코어는 0.8473으로 대회 마감일 기준 1등 팀의 점수와 약 3%의 성능 차이가 났음을 확인할 수 있었다.

분석을 하면서 느낀 한계점 및 아쉬운 점은, 시간적인 여유가 없어 시도하지 못했던 다양한 양상의 기법들을 사용하였다면 성능을 높이는데 더 좋은 결과를 낼 수 있었을 것이라고 생각한다.

또한, 학습과정에서 learning rate, k-fold, batch size, epoch 등 hyperparameter 설정을 우리가 임의로 값을 몇 개 바꿔가며 학습을 진행하였다. Kaggle 내에서 제공해주는 GPU를 사용하여 학습을 돌렸는데, GPU 자원의 부족으로 인해 시도하지 못했던 Random Search나 Grid Search을 하여 Hyperparameter Tuning을 하게 된다면 우리 모델의 성능을 조금 더 올릴 수 있을 것이라 생각한다.

## 참고문헌

- [1] USTPO. (2021). [https://www.uspto.gov/web/offices/ac/ido/oeip/taf/us\\_stat.htm](https://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm)
- [2] He, P., Liu, X., Gao, J. and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- [3] Jacob, D., Chang, M. W., Lee, K. & Toutanova, K. N. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [5] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- [6] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- [7] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- [8] Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.
- [9] Lee, J. S., & Hsiang, J. (2019). Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*.

## Imbalanced Anomaly Detection Using MVtec Data

구지윤<sup>1)</sup>, 김민경<sup>1)</sup>, 이수미<sup>1)</sup>, 이호준<sup>2)</sup>

### Abstract

This research is mainly about detecting anomaly data and construct an algorithm for the classification using MVTec data. Moreover, states of anomaly data should be labeled. Selected models are Resnet and EfficientNet, and the authors compared each models to evaluate their performance. The authors applied two of data augmentation to relieve class imbalance which is the main problem of image classification. One of the data augmentation is an over-sampling data which did not eliminate raw data. The other is an under-sampling data which maximizes the balance between normal and abnormal data. The authors compared classification models using two different augmentation to go over the performance of the model.

---

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

### 1.1. 연구배경 및 목적

이상 탐지(Anomaly Detection)란, 정상(Normal) sample과 비정상(Abnormal) sample을 구별해 내는 문제를 의미하고 불량 검출, 이상 감지 등 정상 상태가 아닌 모든 상태, 물체 등을 감지하는 기술을 말한다. 이 기술은 여러 분야에서 활용이 되는데 제조 환경에서 정상 품질의 제품 이미지 데이터를 학습한 뒤, 불량 이미지를 찾는 제조 품질 검사 분야, 평상시의 로그 데이터 등을 정상 데이터로 학습한 뒤 비정상적인 해킹이나 침입 시도 등을 감지하는 사이버 보안 분야, 신용카드에서 일정한 사용 패턴이 아닌 다른 과도한 결제가 생겼을 때 이를 감지하는 결제 시스템 사기 감지 분야, 평소의 화면을 정상 데이터로 학습한 뒤 이상 징후 등을 찾는 CCTV 이상 감지 분야, 주식 시장 이상 감지, 의료 진단, 지구 과학 이상 감지와 같이 다양한 곳에서 쓰이고 있다.

하지만 이렇게 높은 활용도를 보이는 이상치 탐지 분야에는 몇 가지 어려움이 존재한다. 하나는 적은 이상치 데이터에 있다. 예를 들어 제조 현장에서 제품 품질 검사 장비로 사용하기 위한 Anomaly Detection 알고리즘을 연구한다고 가정해 보자. 대부분의 제품은 정상 품질이므로 정상 데이터를 수집하기는 비교적 수월하다. 반면 불량 제품은 거의 발생하지 않는다. 또한 불량 제품은 무한 가지의 특성을 가지고 있어 학습하기가 더욱 어렵다. 결국 Anomaly Detection 알고리즘은 Normal 데이터에 비해 심히 적은 Abnormal 데이터를 지닌 Class Imbalance 문제를 포함하게 된다. 이러한 Class Imbalance를 고려하여 알고리즘을 구성하는 것이 Anomaly Detection에서 매우 중요하다. Dacon에서 실시한 ‘Computer Vision 이상치 탐지 알고리즘 경진대회’는 사물의 종류를 분류하고, 정상 sample과 비정상 sample을 분류하는 것을 목적으로 한다. 우리는 이번 연구에서 Class Imbalance 문제를 해결하기 위한 노력을 하였고 Resnet, EfficientNet과 같은 다양한 모델을 적용해 보았다.

## 2. 데이터 소개

### 2.1. 데이터 수집

Dacon에서 제공한 mvTec Dataset은 4,277개의 Train Data와 2,154개의 Test Data를 포함하고 있고 이를 분류 모형을 만드는데 활용했다<sup>1)</sup>. Data set은 크게 사진에 대한 정보를 담고 있는 메타정보를 의미하는 train\_df.csv 파일과 train image 파일로 구성되어 있다. train.csv 파일은 index와 label 2개의 column으로 구성이 되어있다.

[표 1] 사용한 변수

변수명	설명
index	이미지의 번호
이미지 정보	
label	이미지 상태
이미지	이미지 파일

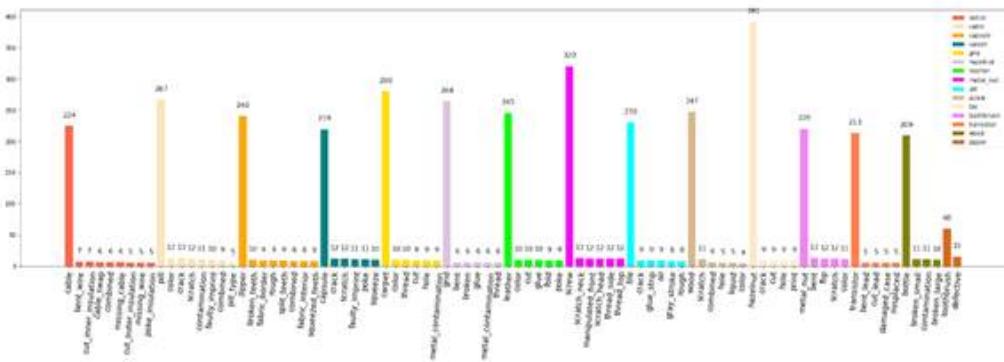
mvTec 데이터는 총 15개의 클래스로 이루어져 있다. Texture 클래스(Carpet, Grid, Leather, Tile, Wood)는 5개이며, Object 클래스(Bottle, Cable, Capsule, Hazelnut, Metal nut, Pill, Screw, Toothbrush, Transistor, Zipper)는 10개로 구분할 수 있다. 앞서 언급한 Class Imbalance 문제를 해결하기 위해 Data augmentation을 두 가지 방식으로 진행하였다. 이 방법에 대해서는 2.2절에서 설명하고자 한다.

## 2.1. 데이터 수집

클래스 불균형을 해소하기 위한 전처리로 data augmentation을 제안하였다. Data augmentation 이란 기준에 가지고 있던 학습 데이터의 고유 정보를 해치지 않으면서 기계가 다양하게 학습할 수 있도록 데이터를 변환하는 것이다.

1) <https://dacon.io/competitions/official/235894/data>

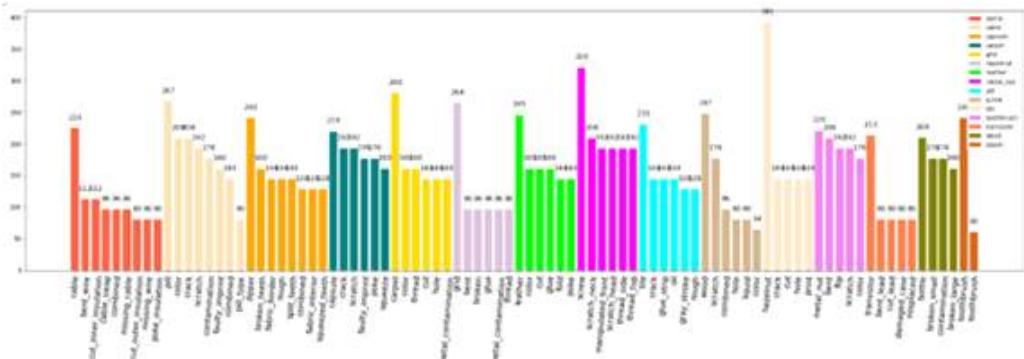
[그림 1] MVTec 학습데이터 시각화



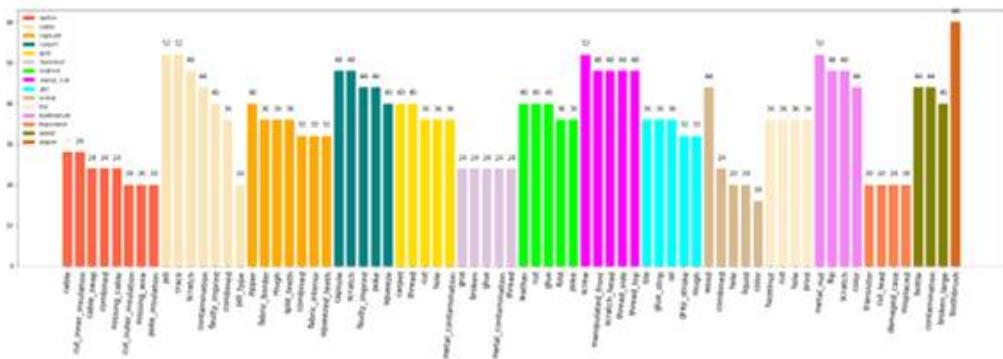
[그림 1]는 MVTec의 Training data를 시각화한 것이다. raw data는 학습 데이터가 4,277개이고, 정상치가 3,629개, 이상치가 648개 있다. 클래스 별 불균형이 심하기 때문에 data augmentation을 통해 불균형을 해소시킬 것이다. 데이터 증강을 over sampling과 under sampling 두 가지로 시도했다. 첫 번째로 시도한 under sampling한 데이터는 Inverted, Converted, Noise Added 총 3개의 옵션을 사용하여 데이터를 증강한 샘플이다. [그림 2]에서 보이는 것처럼, 증강된 데이터를 포함한 후에 보이는 불균형은 raw data를 삭제하여 데이터의 비율을 맞추도록 조정했다. 이를 통해 증강된 데이터는 총 2,563개로, 정상치가 619개, 이상치가 1,944개다.

Over sampling 데이터의 경우는 Random Crop, Rotation, Color Change, Flip 총 4개를 적용하였으며, random하게 4번을 반복하여 데이터를 증가했다. 이를 통해 증강된 학습데이터는 총 13,997개이며, 이 중 정상치가 3,629개이며 이상치가 10,368개다. [그림 3]에서 보여지는 것처럼 over sampling 데이터는 raw data를 보존했으며, 이는 down sampling 데이터와의 가장 큰 차이점이다.

[그림 2] Down sampling 학습데이터 시각화



[그림 3] Over sampling 학습데이터 시각화



### 3. 딥러닝 모델 및 실험

### 3.1. EfficientNet

Image classification 분야는 정확도를 초점으로 한 모델과 효율성의 측면을 고

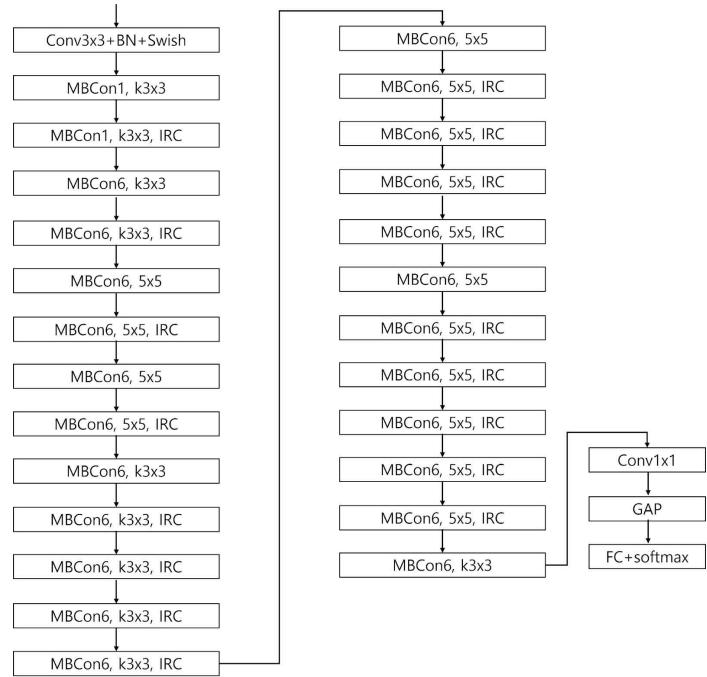
려하는 모델 두 가지의 방향을 가진 모델들이 제안되어왔다. 정확도를 초점으로 한 방향의 연구에서 깊이가 깊고 크기가 큰 모델이 높은 성능을 내는 논문들이 발표되면서, 모델을 크기를 키움으로써 성능을 높이는 연구들이 많이 이루어졌다. 그러나 크기가 큰 모델은 정확도를 높이지만, 사용하는 자원의 크기 또한 크게 늘리기 때문에 종종 over-parameterized 된다. 이에 효율을 위하여 크기가 큰 모델을 압축시키는 기법들이 제안되었고, 효율성을 측면을 고려하면서 동시에 정확도를 올리는 모델로 EfficientNet이 제안되었다.

일반적으로 (1) 모델의 깊이, (2) channel의 너비(filter 개수), (3) 입력 이미지의 크기(해상도) 이 세 가지 요소를 통해 모델의 크기를 조절한다. 모델의 깊이를 깊게 만들수록 채널 너비를 늘릴수록, 해상도를 올릴수록 모델이 커진다.

기준의 연구에서는 3가지의 Scaling의 Coefficient를 임의로 증가시켜 진행하였지만 이는 정확도 향상으로 이어지지 않는 경우가 많아 수동적인 조절의 필요성이 인정되었다. EfficientNet은 깊이, 너비, 이미지 해상도의 최적을 조합을 Grid Search를 통해 찾는다. Grid Search를 통해 Alpha, Beta, Gamma를 찾고, 균형적으로 확장하는 compound scale coefficient에 따라 깊이, 너비, 해상도의 크기가 증가하고 정확도가 향상되는 compound scaling 방법을 제안한다. 즉 깊이, 너비, 입력 이미지의 크기가 일정한 관계가 있다는 것을 실험적으로 찾아 수식으로 만들고 밀접하게 연관된 변수를 함께 조절하여 최고의 효율을 찾아내는 것이 Compound Scailing이다(Tan & Le, 2019).

EfficientNet 모델은 EfficientNet-B0부터 EfficientNet-B7까지의 모델 구조를 가지는데 이는 서로 다른 compound coefficient를 사용하여 기준인 EfficientNet-B0에서 확장된 형태이다(Tan & Le, 2019). 모든 모델에 대해 Train 해본 결과, EfficientNet-B3가 가장 좋은 성능을 기록하여 이후의 실험에는 EfficientNet-B3만을 사용한다. 아래는 실험에 사용한 Efficient-B3의 모델 구조이다.

[그림 4] EfficientNet-B3 모델 구조

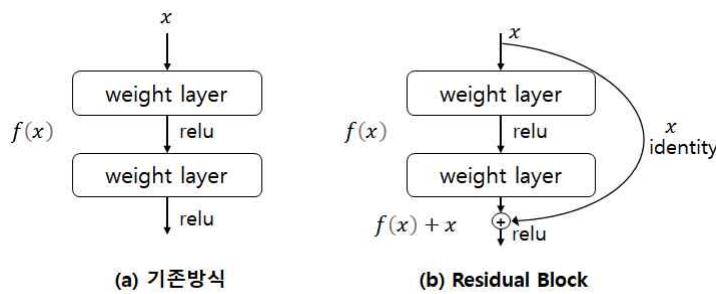


### 3.2. ResNet (Residual Network)

이론상 Deep Networks는 깊은 구조를 가질수록 복잡한 특징들을 학습하기 때문에 성능이 증가해야 하지만 오히려 성능이 저하되는 현상이 발생한다. 이는 단순히 Overfitting에 의한 현상이 아니며 Degradation problem에 의한 현상임이 파악되었다. Degradation problem은 모델의 깊이가 깊어질수록 output이 0인 층들이 생겨 Back-propagation 과정에서 이 층들의 가중치가 변경되지 않는 현상을 말한다(He et al., 2016). 즉, 모델이 깊은 구조를 가질수록 오히려 input이 원래 가지고 있던 특징들이 퇴화될 수 있음을 의미한다. 그러나 너무 얕은 네트워크는 모델의 학습 과정에서 input으로부터 특징을 추출하는데 한계가 있다. ResNet은 이러한 문제를 Residual Block의 도입을 통해 해결하였다. 일반 네트워크가 [그림 7]

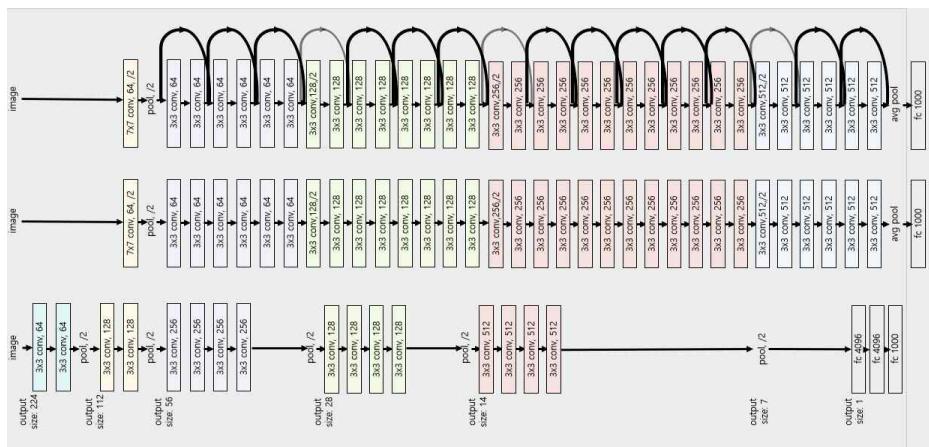
(a)와 같이 input 값들로부터 output 값들로의 매팅이 직접 적합되는 것과 달리 ResNet에서는 [그림 7] (b)와 같이 네트워크가 input 값들로부터 output 값들로의 매팅이 잔차를 이용한 Residual Mapping에 적합되도록 모델링 한다. 따라서 전체를 학습하는 것보다 학습이 용이해지며 수렴 난이도 역시 완화될 수 있다.

[그림 5] Residual Block 구조



ResNet 전체 아키텍처는 기존의 VGG-19 네트워크에 Residual Block을 연속적으로 삽입하는 구현이 비교적 간단한 구조이다. 기존의 VGG 네트워크보다 더 깊지만 Residual Block을 활용해 복잡도와 성능은 더 개선되었다.

[그림 6] ResNet 모델 구조



쌓은 층의 개수에 따라 모델 이름이 ResNet34, ResNet50, ResNet101 등으로

구분할 수 있으며 깊이가 깊어질수록 높은 정확도 향상을 보인다. ResNet34, ResNet50, ResNet101 모델에 대해 실험한 결과, ResNet50이 가장 좋은 성능을 기록하여 이후의 실험에서는 ResNet50만을 사용한다.

### 3.3. 실험 설명

Resnet50 그리고 Efficient B3 모델을 Pre-trained model로 사용하였으며. 데이터를 Raw data, Over sampling data, Under sampling data로 구분하여 각 모델에 대한 실험을 진행하였다. 그리고 샘플이 5개밖에 없는 class가 존재하였으므로 5개의 fold로 train data를 나눈 후 5개 예측값들의 평균을 취하여 test data에 대한 최종 예측값을 계산하는 5-fold Ensemble 기법을 적용하였다. 학습 환경은 Colab의 GPU, GPU TITAN XP으로 진행하였다. 마지막으로 일반화 성능을 높이기 위해 test 단계에서도 Data Augmentation을 하는 Test Time Augmentation(TTA)을 사용하였다. TTA는 Augmented data를 여러 번 보여준 다음 각각의 단계에 대해서 예측값을 평균하고 이 결과를 최종값으로 사용하는 방법이다. 실험 진행 시 Epoch은 25, Batch Size는 32, Learning Rate는 0.001, Optimizer는 Adam Optimizer, Loss는 Cross Entropy Loss를 사용하였다.

## 4. 결론

### 4.1. 실험 결과

Classification model에 쓰이는 평가 산식 중, 개별 F1-score의 평균을 내는 Macro-F1을 평가의 기준으로 설정했다. [표 1]에 따르면, Resnet50, Efficient B3 중에서는 Efficient B3가 성능이 좋은 것을 확인할 수 있다. Efficient B3은 raw data, over sampling data, under sampling data를 적용했을 때 원데이터를 적용한 것이 성능이 0.68로 가장 좋았고, 5-fold 양상블을 추가했을 때 성능이 0.81로 증가한 것을 확인할 수 있다. 데이터 전처리에서 문제가 있다는 것을 확인 할 수 있었다. 대회에서는 80등을 기록했다.

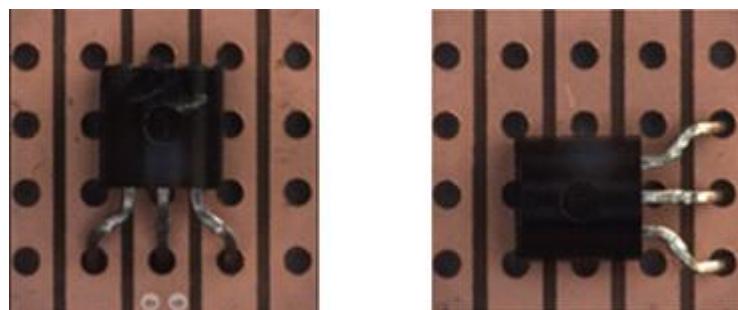
[표 2] 테스트 결과와 F1 Score

	Data	Ensemble	F1 Score
Efficient B3	Raw	yes	0.8149
Efficient B3	Raw	no	0.6808
Resnet50	Over sampling	no	0.5502
Efficient B3	Under sampling	yes	0.2884
Efficient B3	Over sampling	yes	0.2884

## 4.2 시사점과 후속연구 제안

첫 번째로 Imbalance class 문제를 해결하기 위해 적용한 데이터 증강이 오히려 모델의 성능을 떨어트린 문제점에서, 이상치가 포함된 데이터는 데이터 증강에 민감한 것으로 추정했다.

[그림 7] 정상치와 이상치 트랜지스터



[그림 1]의 좌측 트랜지스터의 라벨은 transistor-good이고, 우측은 데이터 증강을 통해 좌측의 그림을 90도 회전시킨 데이터다. 트랜지스터는 위치에 민감하기 때문에 우측의 데이터는 transistor-misplace로 분류되는 불량품이 되어야 한다. 이처럼 부품들 각각 이상치가 되는 요인이 다른데, 사전 지식이 부족하여 이를 모두 고려한 Augmentation을 하지 못한 것에서 한계점이 있었다. -30~30도로만 회전시키는 등 도메인 지식을 최대한 반영을 해본 것 또한 raw data의 성능을 뛰어넘지는 못하였다. 따라서 앞으로는 사전 지식을 정확히 알고 그에 맞춰 Augmentation을 시도하면 더 좋은 성능을 기대할 수 있을 것이라 사료된다.

두 번째, 이상치 분류 모델은 Domain 지식이 갖춰져 있거나, 충분한 자료수집으로 Domain을 파악해야 한다. 앞서 말한 대로 Cable의 전선 색이나 트랜지스터의 위치와 같은 데이터는 Domain이 없다면 이상치임을 판단할 수 없기 때문이다. 이는 앞서 말했던 Domain 지식이 모델 학습에 필요하다는 것을 시사한다. 또한, Domain 지식 없이 증강한 데이터는 오히려 raw data를 적용했을 때보다 성능이 낮아질 수 있다는 것을 확인했다.

세 번째로 이 실험에서 Colab의 computing power의 한계로 모델 학습에 상당한 시간이 걸렸다. 그로 인해 다양한 모델을 학습시킬 수 없었다. 특히 augmented data과 raw data의 score 차이를 찾아보기 위해 데이터에 유효한 증강만을 찾는 과정에서 GPU가 멈추기도 했다. 후속 연구에서는 환경을 개선하여 모델링에 제약이 없도록 하여 더 나은 결과를 낳도록 할 것이다.

Raw data와 augmented data를 적용한 모델을 비교하면서, 오히려 적은 데이터로도 좋은 성능을 내는 few-shot learning (FSL)을 시도해 보는 것을 제안한다. FSL은 규모가 큰 supervised data를 모아야 하는 문제점을 해소시킬 수 있다. 또한 데이터중심의 모델인 이미지 분류에도 FSL을 사용할 수 있기 때문에 후속 연구로 진행하면 좋은 성능을 기대할 수 있을 것이다(Wang, 2020).

## 참고문헌

- [1] Tan, M. & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning (ICML)*.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- [3] Wang, Y., Yao, Q., Kwok, J. T. & Ni, L. (2020). Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Computing Surveys*, 53(3), 1–34.

데이터 과학연구 제11권

2022년 pp.97-113

## Factors affecting social media using propensity score and survival analysis

강진모<sup>1)</sup>, 김희원<sup>1)</sup>

### Abstract

In this study, we analyzed the factors that affect social media usage using the Korea Media Penal Survey from 2001 to 2021. Propensity score matching was used to reduce selection bias due to residential area, and a Cox proportional regression model was employed to assess the effect of factors on use of social media. People who earned Bachelor's degree or higher degrees and were employed have higher social media usage. Old age, rural area, low income were identified to reduce social media usage. There were differences in significance of hazard ratios before and after propensity score matching. This findings suggest that residential area, age, education, employment status, and income affect social media use, which helps policyholders to develop a policy to support people who are exposed to digital alienation.

---

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

스마트폰의 보급 이후로 현대인들이 하루 중 스마트폰을 이용하면서 보내는 시간을 갈수록 늘어나고 있다. 스마트폰을 통해 인터넷 쇼핑을 하거나, 뉴스를 읽거나, 게임을 하는 등 다방면으로 활용하고 있지만, 소셜 네트워크 서비스(Social Network Service, SNS)는 스마트폰의 가장 주된 서비스 중 하나라고 할 수 있다.

특히, 코로나19가 빠르게 확산한 이후부터 SNS는 오프라인의 관계를 온라인으로 연결·확대하며 자기 생각과 감정을 표현하는데 도움을 주었다(김미정, 2021). 현대사회에서 SNS는 사람들을 연결해주는 단순함을 넘어 전문분야에 대한 정보의 생산 및 확산 등을 기제로도 영향력이 늘어나고 있다(김경숙, 이성엽, 2012).

SNS는 메시지 전달 도구로서의 기능이 점차 발전하고 있으며, 앞으로도 SNS의 중요성은 계속해서 높아질 것으로 예상할 수 있다(박현길, 2010). 하지만 이러한 급격한 변화에 적응하지 못하는 SNS 소외계층은 이전의 PC 및 인터넷 환경에서 발생하던 정보격차와는 또 다른 양상의 어려움에 노출되고 있다.

정보격차의 문제가 심화하는 빠른 변화 물결 속에서 SNS 이용에 어떠한 개인적 요인과 사회적 요인이 영향을 주는지에 대해 구체적으로 확인하고자 하는 노력이 필요하다. 하지만 그동안 SNS와 관련되어 진행된 선행연구는 주로 SNS의 기술 특성(최경석, 2021)이나 사용자의 행위(김성태, 2021) 등에 집중하거나 SNS 중독(조소연, 정주원, 2017) 등 문제점에 주목하는 경우가 많았으며, 설문 또는 면담을 통한 SNS 사용 경험의 탐색(박선미, 2022) 등이 주를 이루었다.

개인적·사회적 요인에 따라 SNS 이용률에 어떠한 변화가 나타나는지 파악하기 위해서는 장기간에 걸쳐 같은 대상자를 종단적으로 관찰해야 비교적 정확한 추정이 가능하지만, 자료 확보 등 현실적인 문제로 이를 만족하는 설계의 연구는 부족한 실정이다. 따라서 본 연구에서는 같은 연구대상에 대하여 다년간에 걸쳐 조사한 패널 데이터를 바탕으로 생존분석을 활용하여 SNS 이용률과 개인의 특성 간의 관계를 분석하고, SNS 이용률에 영향을 끼치는 요인을 탐색하는 것을 목적으로 한다.

## 2. 이론적 배경

### 2.1. 소셜 네트워크 서비스(Social Network Service; SNS)

SNS는 개인의 온라인 프로필을 바탕으로 형성된 관계와 연결에 따른 상호작용을 지원하는 웹 기반의 서비스를 의미한다(Boyd & Ellison, 2007). 또한, 이용자 간의 자유로운 의사소통과 정보 공유, 인적 네트워크 확대를 통해 사회적 관계를 유지해주는 온라인 플랫폼을 의미하기도 한다(이제홍, 황규영, 2018).

소셜 플랫폼에서 사용자들은 소소한 일상에서부터 정보와 지식, 노하우를 공유하며 사회정치 참여에 이르기까지 자유롭게 콘텐츠를 생성하고 유통하며 소비한다(박경자, 고준, 박승봉, 2015). 스마트폰의 확산과 함께 SNS 이용의 확산속도 또한 활발해지고 있다. 특히 한국 사회는 인간관계를 중요시하는 까닭에 SNS는 사회 구성원들로부터 큰 관심을 얻으면서 급속도로 확산하였다(최향섭, 2011). 예전에는 텍스트 중심이었던 트위터를 주로 사용했다면 최근에는 사진이나 동영상을 찍어, 온라인상에서 관계를 맺고 타인에게 공유하는 형태의 시각적 소셜 미디어인 인스타그램(전소정, 성용준, 양은주, 2018)이 SNS 유행을 이끌고 있으며 그리고 짧은 동영상으로 시간 부담 없이 쉽게 즐길 수 있는 쇼트 클립이 강력하고 새로운 형태의 미디어로 자리를 잡고 있다(이유진, 유세경, 2018).

SNS를 통해 이용자들은 같은 관심사를 가진 사람들과 손쉽게 소통하며 자신의 감정, 지식, 정보 등을 공유할 수 있다. 또한, 상호 간에 정보 교환이 매우 쉬워지면서 같은 제품 및 서비스의 소비자 사이에서도 많은 교류가 이루어질 수 있게 되었고, 소비자들은 기업, 정부에 대해서도 과거보다 능동적인 자세와 대응을 취할 수 있게 되었다. 이제 SNS는 사람들에게 가장 중요한 소통의 공간 중 하나로 자리 잡았으며, 사회에서의 중요성과 관심도는 계속해서 증가하고 있다. 따라서 많은 연구자가 SNS 이용 여부, 이용 빈도에 영향을 미치는 요인들을 규명하는 데에 관심을 기울이고 있으며, 특히 개인적 요인과 사회적 요인에는 각각 무엇이 있는지 연구할 필요성이 있다.

## 2.2. 디지털 정보격차

정보 소외계층이란 경제적, 사회적 신체적 여건 때문에 정보통신기기를 활용하는데 여러 가지 어려움이 존재하는 계층을 말한다(최선경, 2020). 지식의 획득이 곧 개인의 능력, 직업, 소득 등에 큰 영향을 끼치는 작금의 현실은 디지털 정보에 대한 접근성과 기술의 활용 가능성이 곧 삶의 질을 좌우하는 결정적 요소로 작용할 수 있음을 보여준다.

과거 디지털 정보격차의 개념은 주로 컴퓨터나 인터넷 등에 접근할 수 있는 사람과 그렇지 못한 정보 소외계층 간의 차이를 중심으로 이해되었고, 정부의 정책 또한 컴퓨터의 보급이나 통신망의 구축 중심으로 진행되었다. 그러나 교외 지역까지 인

터넷 사용에 어려움이 없어진 현재에는 디지털 정보의 활용 역량과 수준에서의 차이 등으로 인한 디지털 정보격차에 주목하고 있다. 특히 주요 선진국에서 인터넷 이용률과 스마트폰 보급률이 90%에 가까워지면서, 연구자들은 다양한 차원의 디지털 정보격차에 정책적 노력을 기울여야 함을 강조하고 있다(Ragnedda, 2017). 디지털 정보격차 실태를 파악하려는 노력은 과거부터 꾸준히 활발하게 이루어지고 있다. 연령, 성별에 따른 차이뿐만 아니라 저소득, 장애인, 농어민 등 사회적 약자의 정보격차 문제가 강조되고 있으며(이정연, 김현애, 2021), 기본적인 요인 외에도 가구의 유형, 거주 지역, 인종 등의 다양한 원인으로 인해 발생하는 디지털 정보격차에 관한 연구가 진행되고 있다. 이렇듯 중요한 문제로 다루어지는 디지털 정보격차 현상은 우리 사회가 직면해 있는 가장 큰 문제 중 하나라고 할 수 있다. 이 같은 우려를 염두에 두고 거주 지역을 포함한 여러 요인과 SNS 이용률 간의 관련 가능성을 살펴보고자 한다.

### 3. 연구 설계

#### 3.1. 분석 자료

본 연구에서는 정보통신정책연구원(Korea Information Society Development Institute: KISDI)에서 제공하는 ‘한국미디어패널’을 활용하였다. 한국미디어패널조사는 매년 실시되는 동일표본 추적 조사로, 동일가구와 개인의 미디어 환경 및 이용 행태 변화를 추적하기 위해 구축되었다. 첫 조사는 2010년이지만, 2011년부터 조사 대상을 전국 16개 광역시로 확대하여 총 5,109가구 및 만 6세 가구원 12,000명의 대상자로 패널을 구축하여 매년 6월에 추적 조사하였으며, 지역단위 충화추출 기법을 사용하여 특정 지역에 조사 대상자가 집중되지 않도록 표본을 선정하였다. 이러한 특성으로 기존의 다른 미디어 관련 통계자료들보다 추정치의 신뢰성과 효율성을 확보할 수 있다는 것이 장점이다.

본 연구에서는 수도권 거주 여부에 따라 SNS 이용률에 유의한 차이를 보이는지 살펴보기 위해 한국미디어패널의 2011년부터 2021년까지의 11개 연도 데이터를 활용하여 분석하였다. 첫 조사연도인 2011년에 SNS를 사용하는 1,633명을 제외한 10,367명을 최종 분석 대상으로 선정하였다. 해당 데이터는 조사 대상자의 인구통계학적 특성, 뉴미디어 이용 현황, 미디어 이용 행태에 관련된 정보 등을 포함하고 있다.

### 3.2. 변수 구성

본 연구는 거주 지역이 SNS 이용에 주는 효과를 분석하기 위하여 종속변수로는 ‘SNS 이용 여부’ 문항을 활용하였다. 연구 대상이 되는 표본을 대상으로 총 11년에 걸쳐 SNS 이용 여부를 확인하였다. 관심변수는 거주 지역으로, 설문 문항 중 ‘지역’ 변수를 활용하였고, 수도권과 비수도권 두 범주로 재구성하였다. 즉, 서울, 인천, 경기와 같은 지역에 거주한다면 수도권으로, 그 이외 지역에 거주한다면 비수도권으로 더미 변수를 구성해 관심변수로 활용하였다.

또한, SNS 이용 여부에 영향을 미칠 수 있는 개인의 특성을 나타내는 성별, 연령, 최종학력, 직업 유무, 소득 수준을 공변수로서 모형에 투입하였다. 연령 변수는 분포와 연령대 특성을 고려하여 ‘30세 미만’, ‘30세 이상 50세 미만’, ‘50세 이상’으로 3개의 범주로 구성하였다. 최종학력 변수는 무학·미취학 및 초등학교, 중학교, 고등학교 졸업을 ‘고졸 이하’로, 대학교, 대학원 졸업을 ‘대졸 이상’으로 분류하였다. 소득 변수의 범주는 ‘무소득’ ‘200만 원 미만’, ‘200만 원 이상’으로 구성하였다. 각 변수의 정의 및 구성은 [표 1]과 같다.

### 3.3. 연구 방법

#### 3.3.1 성향점수 매칭(Propensity Score Matching: PSM)

본 연구에서는 수도권과 비수도권 응답자의 SNS 이용까지 걸리는 기간을 비교함으로써, 거주 지역이 SNS 이용도에 미치는 영향을 살펴보았다. 이러한 매칭을 시행하는 이유는 수도권에 거주하는 집단과 비수도권에 거주하는 집단 간에 사회적 및 경제적 특성의 차이가 존재하여 거주 지역에 따른 차이를 정확히 파악하기 어렵기 때문이다. 이를 고려하지 않고 그대로 분석한다면, 거주 지역과 무관하게 두 집단은 이미 서로 다른 집단일 가능성이 높으며, 결과의 추정에서도 편의가 발생할 수 있다. 이러한 교란변수(confounder)로 인해 원인과 결과의 관계가 편향되는 현상을 선택 편의(selection bias)라고 한다. 본 연구에서는 교란변수들을 활용해 응답자가 어느 지역에 거주할지에 대한 확률을 의미하는 성향점수(propensity Score)를 추정하고 이를 모형화하여 선택 편의를 줄임으로써 거주 지역에 따른 차이의 추정을 어렵게 하는 공변수의 영향을 줄이고자 하였다.

[표 1] 변수 정의 및 구성

항목	변수명	설명
종속변수	SNS 사용	아니오:0, 예:1
관심변수	거주 지역	수도권:0, 비수도권:1
공변량	성별	남자:0, 여자:1
	연령	30세 미만:0, 30세 이상 50세 미만:1, 50세 이상:2
	최종학력	고졸 이하:0, 대학 이상:1
	직업 유무	없음:0, 있음:1
	소득	소득 없음:0, 월 소득 200만 원 미만:1, 월 소득 200만 원 이상:2

성향점수는 쳐치 혹은 통제집단에 속한 참여자들의 관찰 가능한 특성들( $X$ )이 주어졌을 때, 이들이 쳐치 집단에 참여할 조건부 확률( $P(T=1 | X)$ )을 나타낸다 (Rosenbaum & Rubin, 1983). 성향점수가 같거나 유사한 쳐치집단과 통제집단은 공변수의 균형성을 만족하여 유사한 분포를 가지게 된다. 따라서 두 집단 간의 종속변수 값을 비교했을 때 무작위 실험 평가(randomized experiment)에서 수행한 것과 같이 선택 편의가 없는 인과효과를 추정할 수 있다.

### 3.3.2 생존 분석

본 연구에서는 거주 지역이 SNS 이용에 미치는 영향을 검증하기 위해 생존분석 (survival analysis)을 활용하였다. 생존분석은 관찰 대상이 특정 사건을 경험할 때

까지 걸리는 시간을 모형화하여 시간에 따른 사건의 발생 확률을 추정하고, 사건 발생에 영향을 주는 요인을 밝히는 분석 기법이다(Kalbfleisch & Prentice, 1980). 본 연구에서는 SNS 이용까지 걸리는 기간이 생존시간이라고 할 수 있으며, SNS 이용 여부 문항에 ‘예’로 응답하는 경우가 사건, 각 응답자가 관찰 대상으로 대응된다.

생존분석의 가장 큰 특징은 미완결된 자료를 반영할 수 있다는 점일 것이다. 이를 중도절단(censoring)이라 하며, 사망, 탈퇴 등 여러 이유로 인하여 관찰 대상에서 특정 시점에 제외되는 경우를 일컫는다. 생존시간을  $T$ , 사건이 발생하는 시점을  $t$ 라고 한다면 생존함수(survival function)  $S(t)$ 는  $t$ 까지 생존할 확률을 의미하며, 위험함수(hazard function)  $h(t)$ 는  $t$  시점 직후 사건이 발생할 확률을 의미한다. 생존 함수와 위험함수는 각각 다음과 같이 정의할 수 있다(Lawless, 1982; Harris and Albert, 1991).

$$S(t) = \Pr(T > t) = 1 - \int_0^t f(t)dt \quad (1)$$

$$h(t) = \lim_{\text{TRIANGLE}t \rightarrow 0} \frac{\Pr[t + \text{TRIANGLE}t > T > t | T > t]}{\Delta t} \quad (2)$$

### (1) Kaplan-Meier 방법

생존율을 추정하는 대표적인 방법으로 Kaplan-Meier 방법을 들 수 있다. 이는 생존함수에 어떠한 분포를 가정하지 않은 채로 생존확률을 추정하는 비모수적 방법이다. 이 방법은  $t$  시점까지의 추정 생존함수인  $\hat{S}(t)$ 와 각 시점을 생존비인  $p_i$ 를 통해 도출할 수 있으며, 이는 시점  $i$ 에서 생존자 수  $n_i$ 와 사망자 수  $d_i$ 를 이용하여 아래와 같이 구할 수 있다.

$$p_i = \frac{n_i - d_i}{n_i}, \quad i = 1, 2, \dots, t \quad (3)$$

$$\hat{S}(t) = \prod_{i \leq t} \left[ 1 - \frac{d_i}{n_i} \right] \quad (4)$$

Kaplan-Meier 방법은 관찰 대상을 나누는 범주에 따라 생존율이 같은지에 대한 검정을 수행할 수 있는 여러 가지 방법이 있다. 그중 대표적인 방법으로 log-rank test가 있다. log-rank test는 각 사건 발생 시점에서 관찰된 사건 수와 예측된 사건 수의 차이를 통하여 비교하는 방법이다. 귀무가설은 ‘각 시점에서 집단 간 생존

율의 차이가 없다'이다.

## (2) Cox 비례위험 모형

본 연구는 연체 발생까지의 기간, 즉 생존기간을 Cox 비례위험 모형을 통해 추정하였다. Cox의 비례위험 모형은 특정 질환 발생에 영향을 주는 위험 인자를 규명하고, 생존에 미치는 영향을 통계적으로 검정할 수 있는 분석 방법이다. 이 모형은 Cox(1972)에 의해 제안되었으며, 모형의 형태는 다음과 같다.

$$h(t) = h_o(t)e^{\beta_1x_1 + \beta_2x_2 + \cdots + \beta_px_p} \quad (5)$$

Cox 비례위험 모형은 분포를 가정하지는 않지만, 회귀계수를 추정하므로 semiparametric 모형이라고도 한다. 이를 바탕으로 두 개체 또는 집단 간 위험비(hazard ratio)를 표현할 수 있는데, 수식으로 나타내면 다음과 같다.

$$HR = \frac{h_i(t)}{h_j(t)} = \lambda \quad (6)$$

위험비는 두 집단 간 상대적으로 사건이 일어날 확률의 차이를 나타낸다. Cox 비례위험 모형은 이러한 위험비를 시간과 무관하게 일정하다는 가정을 하는데, 이를 비례위험(proportional hazard) 가정이라고 한다. 만일 이 가정을 충족하지 않는 경우, 과대 추정으로 인해 정확한 추정량을 얻기 어렵다. 가정이 위배되지 않는지에 대한 확인 과정이 필요하다. Kaplan-Meier 생존율 S(t)를 구한 후 이를 이용하여  $\ln[-\ln S(t)]$ 를 그리는(LLS 그래프) 방법, 적합도 검정 등의 통계적 검정을 이용하여 확인하는 방법 등이 있다.

## 4. 분석 결과

### 4.1. 기술 분석 결과

성향점수 매칭 전 거주 지역에 따른 대상자의 특성은 [표 2]와 같으며, 각 변수에 대하여 카이제곱(chi-squared) 검정을 시행하였다. 본 연구의 분석 대상에 대한 일반적 특성을 살펴보면 다음과 같다. 먼저 거주 지역에 따라 수도권이 3,844명

(37.1%), 비수도권이 6,523명(62.9%)으로 비수도권 거주자가 수도권 거주자보다 많은 것을 확인할 수 있다.

[표 2] 성향점수 매칭 전 거주 지역에 따른 대상자 특성

변수	거주 지역				유의확률	
	수도권		비수도권			
	관측 수	비율(%)	관측 수	비율(%)		
총 인원	3,844	37.1	6,523	62.9		
성별 남자	1,750	45.5	2,999	46.0	0.672	
여자	2,094	54.5	3,524	54.0		
연령 30세 미만	1,081	28.1	1,414	21.7	<0.001	
30~49세	1,657	43.1	2,049	31.4		
50세 이상	1,106	28.8	3,060	46.9		
학력 고졸 이하	1,331	34.6	3,134	48.0	<0.001	
대졸 이상	2,513	65.4	3,389	52.0		
직업 없음	2,222	57.8	3,494	53.6	<0.001	
있음	1,622	42.2	3,029	46.4		
소득 없음	1,958	50.9	3,232	49.5	<0.001	
200만 원 미만	862	22.4	2,080	31.9		
200만 원 이상	1,024	26.6	1,211	18.6		

또한, 카이제곱 검정 결과에 따르면 거주 지역에 따라 연령, 최종학력, 직업 유무, 소득의 분포에 차이가 있는 것으로 나타났다.

구체적으로, 수도권은 연령이 30~49세에 해당하는 경우가 1,657명(43.1%)으로 가장 큰 비중을 차지했지만, 비수도권은 연령이 50세 이상에 해당하는 대상자가 3,060명(46.9%)으로 비수도권 대상자의 절반가량을 차지했다. 최종학력은 대졸 이상의 비율 수도권은 65.4%, 비수도권은 52.0%로 통계적으로 유의한 차이를 보였다. 직업이 있는 것으로 응답한 대상자는 수도권이 42.2%, 비수도권이 46.4%로 나타났으며, 소득의 경우 ‘월 200만 원 이상’으로 응답한 대상자가 수도권이 26.6%, 비수도권이 18.6%로, 거주 지역에 따른 소득 금액의 분포 차이를 확인할 수 있다.

성향점수 매칭 후 거주 지역에 따른 대상자의 특성은 [표 3]과 같으며, 매칭 전과 마찬가지로 각 변수에 대하여 카이제곱 검정을 시행하였다. 성향점수를 이용하여 1:1 최적(optimal) 매칭을 수행하여 수도권과 비수도권 대상자가 각각 3,844명으로 같게 할당된 것을 확인할 수 있다.

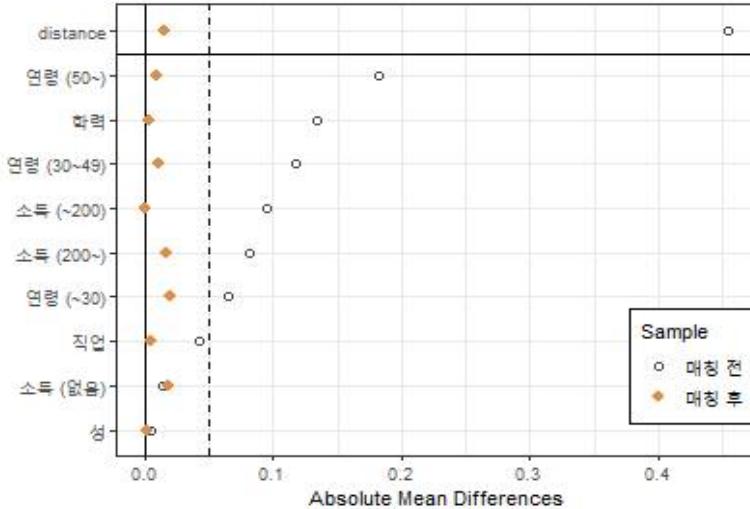
[표 3] 성향점수 매칭 후 거주 지역에 따른 대상자 특성

변수	거주 지역				유의확률	
	수도권		비수도권			
	관측 수	비율(%)	관측 수	비율(%)		
총 인원	3,844	50.0	3,844	50.0		
성별 남자	1,750	45.5	1,757	45.7	0.891	
여자	2,094	54.5	2,087	54.3		
연령 30세 미만	1,081	28.1	1,156	30.1	0.169	
30~49세	1,657	43.1	1,616	42.0		
50세 이상	1,106	28.8	1,072	27.9		
학력 고졸 이하	1,331	34.6	1,340	34.9	0.848	
대졸 이상	2,513	65.4	2,504	65.1		
직업 없음	2,222	57.8	2,238	58.2	0.729	
있음	1,622	42.2	1,606	41.8		
소득 없음	1,958	50.9	2,025	52.7	0.196	
200만 원 미만	862	22.4	860	22.4		
200만 원 이상	1,024	26.6	959	24.9		

성향점수 매칭을 하기 전과 후에 거주 지역에 따른 각 변수의 분포 변화를 확인할 수 있었다. 매칭을 하기 전에는 수도권과 비수도권 사이에 연령, 최종학력, 직업 유무, 소득에서 통계적으로 유의한 차이가 있었지만, 성향점수 매칭을 한 후에는 두 집단 간 모든 변수에서 통계적으로 유의하다고 볼 수 있는 차이가 없음을 확인할 수 있다. 이를 통해 매칭이 비교적 잘 이루어졌다고 판단할 수 있다.

성향점수 매칭 전후의 공변량 균형성 변화를 살펴보기 위해 러브플롯(Love plot)을 이용하였다. 러브 플롯은 토마스 러브(Thomas E. Love)가 제안한 공변량 균형성 점검 시각화 방법이다(Ahmed et al., 2006). 러브 플롯으로 시각화한 성향점수 및 공변량의 평균 차이 결과는 [그림 1]과 같다. 결과를 종합해 보았을 때, 성향점수 (distance)를 포함한 모든 공변량의 평균차이가 0.05 이내로 줄어든 것을 확인할 수 있다. 이러한 결과를 보았을 때, 성향점수 매칭을 통해 두 집단 간 균형성을 충분히 달성했다고 평가할 수 있다.

[그림 1] 매칭 전후 균형성 점검 러브 플롯



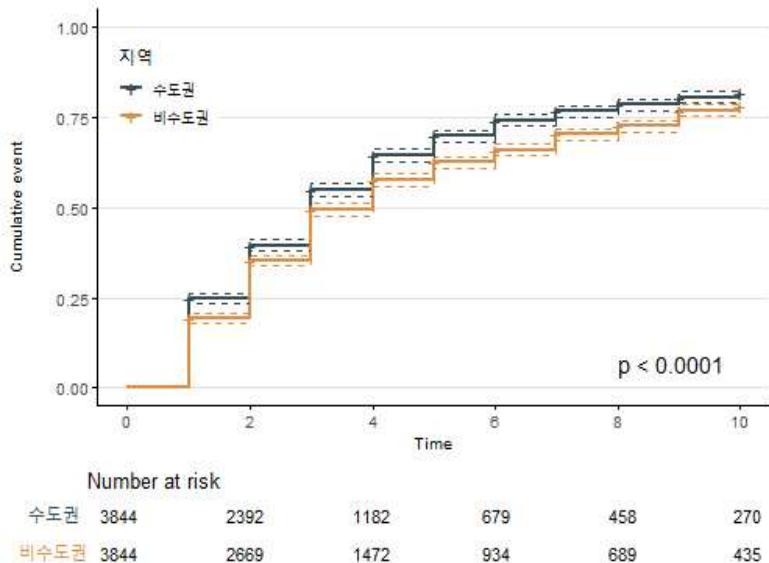
다음으로 Kaplan-Meier 방법으로 생존률을 추정하여 거주 지역에 따른 SNS 이용에 차이가 있는지 분석하였다. [그림 2]의 y축은 Kaplan-Meier 방법을 통해 추정한 각 시점의 누적 사건 발생률, 즉 SNS 이용률을 나타내었다. 수도권이 비수도권의 경우보다 SNS 이용률이 높은 것을 가시적으로 확인할 수 있다. 또한 두 곡선이 교차하는 경우가 없으며, log-rank test 결과 거주 지역에 따라 생존함수가 통계적으로 유의미한 차이가 있다고 해석할 수 있다.

#### 4.2. 생존분석 결과

본 연구는 성향점수 매칭을 활용하여 수도권과 비수도권 간 공변수의 분포를 유사하게 조정한 후, Cox 비례위험 모형을 통해 SNS 이용에 대한 거주 지역의 인과적인 영향력을 추정하고자 하였다.

Cox 비례위험 모형의 적합에 앞서, LLS 그래프를 바탕으로 비례성 가정의 만족 여부를 검토하였다. LLS 그래프가 교차하지 않고 평행한 패턴을 보이게 되면 그 변수는 비례성 가정을 만족한다고 판단할 수 있다. 성향점수 매칭 전후의 변수별 LLS 그래프는 [부록 1]과 [부록 2]에 제시하였다. 검토 결과, 모든 변수에서 비례성 가정을 위배하는 경우가 없는 것으로 판단할 수 있었으며, 모든 변수를 모형에

[그림 2] 매칭 후 거주 지역별 Kaplan-Meier



투입하여 분석을 진행하였다.

[표 4]와 [표 5]는 각각 성향점수 매칭 전과 후에 Cox 비례위험 모형을 적합한 결과이다. 분석 결과, 성향점수 매칭 전후의 차이점을 발견할 수 있었다.

우선, SNS 이용률에 통계적으로 유의하게 영향을 미치는 요인으로 거주 지역, 연령, 최종학력, 직업 유무, 소득(200만 원 미만)이 매칭 전후에 모두 동일하게 나타났다. 하지만 그 위험비에서는 다소 차이를 확인할 수 있는데, 거주 지역의 경우 0.77에서 0.81로 증가하였으며, 연령은 30~49세의 경우 0.34에서 0.37, 50세 이상의 경우 0.09에서 0.11로 모두 소폭 증가하였다. 이러한 결과는 성향점수 매칭을 통해 두 집단의 균형성을 맞추지 않고 Cox 비례위험 모형을 적합한다면 해당 변수들이 SNS 이용률에 미치는 영향력이 과소 추정될 수 있음을 보여준다.

이와는 달리 최종학력의 경우 위험비가 매칭 전 2.64에서 2.34로 오히려 감소하였으며, 이는 같은 맥락으로 영향력의 과대 추정의 문제가 발생할 수 있었음을 나타낸다. 직업 유무는 위험비가 1.18에서 1.20으로 소폭 증가하였으며, 소득은 월 200

[표 4] 성향점수 매칭 전 Cox 비례위험 모형 분석 결과

변수	위험비	95% 신뢰구간		유의확률
		하한	상한	
지역 수도권	1.00			
비수도권	0.77	0.73	0.81	<0.001
성 남자	1.00			
여자	1.05	0.99	1.11	0.121
연령 30세 미만	1.00			
30~49세	0.34	0.31	0.37	<0.001
50세 이상	0.09	0.08	0.10	<0.001
학력 고졸 이하	1.00			
대졸 이상	2.64	2.46	2.84	<0.001
직업 없음	1.00			
있음	1.18	1.06	1.33	0.004
소득 없음	1.00			
200만 원 미만	0.84	0.75	0.95	0.004
200만 원 이상	1.17	1.03	1.33	0.014

만 원 미만의 경우 0.02만큼 감소하였다.

소득 200만 원 이상의 경우는 성향점수 매칭 이전에 SNS 이용률을 통제적으로 유의하게 예측하는 것으로 확인되었지만, 매칭 후에는 SNS 이용률을 예측하는 요인이 아닌 것으로 나타났다. 이처럼 서로 상이한 두 가지 분석 결과는 수도권과 비수도권 집단 간의 특성으로 인해 소득 분포 등이 같지 않은 상황에서 그대로 두 집단을 비교하는 것이 다른 결과를 도출할 수 있음을 보여준다.

위 요인 중 SNS 이용률에 미치는 영향력 면에서는 최종학력이 대졸 이상일 때 2.34로 가장 크게 나타났으며, 다음으로 직업이 있는 경우로 확인되었다. 반대로 SNS 이용률을 감소시키는 영향력에서는 50세 이상 연령의 경우가 0.11로 가장 컸으며, 다음으로 30~49세 연령, 비수도권 거주, 월 소득 200만 원 미만 순으로 나타났다.

[표 5] 성향점수 매칭 후 Cox 비례위험 모형 분석 결과

변수	위험비	95% 신뢰구간		유의확률
		하한	상한	
지역	수도권	1.00		
	비수도권	0.81	0.76	0.85 <0.001
성	남자	1.00		
	여자	1.06	0.99	1.13 0.105
연령	30세 미만	1.00		
	30~49세	0.37	0.34	0.40 <0.001
	50세 이상	0.11	0.10	0.13 <0.001
학력	고졸 이하	1.00		
	대졸 이상	2.34	2.16	2.54 <0.001
직업	없음	1.00		
	있음	1.20	1.06	1.37 0.005
소득	없음	1.00		
	200만 원 미만	0.82	0.72	0.94 0.004
	200만 원 이상	1.10	0.95	1.26 0.203

## 5. 결론

본 연구의 주된 목적은 거주 지역이 수도권인 집단과 비수도권인 집단의 특성을 통제하여 집단 간 균형성을 달성한 후 거주 지역이 SNS 이용률에 미치는 영향을 살피는 한편, 개인적 특성 및 사회적 특성들과 SNS 이용률 간의 관계를 추정하는 데 있다. 그리고 이러한 분석 결과를 토대로 정보 소외계층의 SNS 이용 제고를 위한 정책적 합의와 실증 근거를 찾고자 하였다.

본 연구의 분석 결과로부터 다음과 같은 점들을 확인할 수 있다. 첫째, 성향점수 매칭을 적용하지 않고 분석한 결과에 따르면 월 소득 200만 원 이상이 SNS 이용률을 높이는 요인인 것으로 나타났다. 그러나 성향점수 매칭을 통해 거주 지역에 대한 선택 편의를 제거한 결과, 월 소득 200만 원 이상인 경우와 SNS 이용 간에 통계적인 관련성이 없는 것으로 나타나 앞선 분석 결과와 다른 결과를 나타내었다. 둘째, SNS 이용률을 낮추는 요인으로는 50세 이상인 경우, 30~49세인 경우, 비수도권 거주인 경우, 월 소득 200만 원 미만인 경우가 해당하는 것으로 확인되었다. 반면, SNS 이용률을 높이는 요인은 최종학력이 대졸 이상인 경우, 직업이 있는 경

우로 나타났다.

셋째, 성향점수 매칭 전과 후에 변수별로 SNS 이용률에 미치는 영향력의 차이가 나타나는 것으로 확인되었다. 성향점수 매칭을 적용한 뒤에 영향력이 감소한 경우는 최종학력, 월 소득 200만 원 미만이 해당하였으며, 영향력이 증가한 경우는 거주 지역, 연령, 직업 유무가 해당하는 것으로 나타났다.

이와 같은 결과는 거주 지역을 포함한 개인별 특성에 따른 SNS 접근성 및 활성화 방안에 대한 다각적인 제도적 지원을 모색할 필요성을 시사한다.

본 연구는 약 11년간 SNS 이용 여부를 조사한 패널 데이터를 통해 영향요인을 종단적으로 분석하고 추정해 보았다는 데에서 그 의의를 확인할 수 있다. 또한, 성향점수 매칭을 통해 처치변수인 거주 지역에 따른 집단의 균형성을 만족한 후 처치의 효과를 검증한 점이 본 연구의 강점이라고 할 수 있다.

그럼에도 본 연구는 몇 가지 한계점을 가지는 까닭에 후속 연구에서 이에 대한 개선 및 보완이 필요하다. 우선 본 연구는 이용하고 있는 SNS의 종류나 이용 시간, 빈도 등에 대한 세부적인 분석을 시행하지 못하였다. 따라서 본 연구는 SNS 이용 여부를 통해서만 그 효과를 조명하고 추정함에 따라 다각적인 효과를 구체적으로 평가하지는 못하였다는 한계를 가진다. 또한, 관찰 시점에 따라 변할 수 있는 직업 유무, 소득 금액 등의 변수에 대하여 기준시점의 값만을 활용하였기에, 이러한 시변(time-varying) 변수를 반영할 수 있는 정교한 모형을 고려한다면 더욱 유의미한 연구결과를 이끌어낼 수 있을 것으로 판단된다.

## 참고문헌

- [1] 김경숙, 이성엽 (2012). SNS를 활용한 직장인의 무형식학습 사례 연구: Facebook 활용을 중심으로, *HRD연구*, 13(4), 31-61.
- [2] 김미정 (2021). 대학생용 SNS정서표현성 척도 개발 및 타당화. *충북대학교 대학원 박사학위논문*.
- [3] 김성태 (2021). 관광 SNS 정보서비스특성, 관계 질, 지속적 이용의도 간의 구조적 관계: 관계 질의 매개효과를 중심으로. *Tourism Research*, 46(1), 93-114.
- [4] 박경자, 고준, 박승봉 (2015). 소셜네트워크서비스 사용중단은 왜 발생하는가?: 지각된 비용과 커플링효과를 중심으로. *인터넷전자상거래연구*, 15(1), 1-16.
- [5] 박선미 (2022). 성인학습자의 SNS 사용 경험 탐색: 코로나19로 인한 비대면 상황을 중심으로. *학습자중심교과교육연구*, 22(9), 443-462.
- [6] 박현길(2021). 스마트폰의 동반자-소셜네트워크서비스(SNS). *마케팅*, 44(9), 57-65.
- [7] 이유진, 유세경 (2018). 짧은 동영상 이용 동기가 동영상 유형별이용 정도에 미치는 영향에 관한 연구. *한국방송학보*, 32(4): 65-102.
- [8] 이정현, 김현애 (2021). 비대면 시대의 지식정보취약계층 일상생활 정보요구와 도서관 이용 경험에 관한 연구. *한국비블리아학회지*, 32(1), 223-246.
- [9] 이제홍, 황규영 (2018). 소셜네트워크서비스(SNS) 품질요인이 이용자만족 및 구전 의도에 미치는 영향. *e-비즈니스연구*, 19(1), 123-134.
- [10] 전소정, 성용준, 양은주(2018), 소셜미디어 이용행동과 여성의 신체상의 관계: 자기대상화 이론을 중심으로. *한국심리학회지: 여성*, 23(1), 69-89.
- [11] 조소연, 정주원(2017). 중학생의 SNS중독 경향성에 있어 내현적 자기애와 소외감, 자아존중감의 관계. *한국가정교육학회지*, 29(3), 125-140.
- [12] 최경석 (2021). SNS의 특성을 이용한 광고 매체로의 활용에 대한 연구, *조형미디어학*, 24(4), 12-18.
- [13] 최선경 (2020). 성인지적 관점의 지역사회 여성장애인 디지털정보격차 현황과 역량강화기반 정보화교육 지원 방안. *한국정보통신학회논문지*, 24(5), 655-661.
- [14] 최항섭 (2011). 소셜네트워크서비스(SNS) 이용의 부상과 확산. *한국의 사회동향 2011 – 문화 및 여가*.
- [15] A. Ahmed et al. (2006), Heart failure, chronic diuretic use, and

- increase in mortality and hospitalization: an observational study using propensity score methods. *European Heart Journal*, 27, 1431–1439.
- [16] Boyd, D. M. and Ellison, N. B. (2007). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1), 210–230.
  - [17] Cox, D. R. (1972). Regression models and life tables. *Journal of Royal Statistical Society*, 34, 187–220.
  - [18] Kalbfleisch, J. D. and Prentice, R. L. (1990). The Statistical Analysis of Failure Time Data. *New York : John Wiley & Sons, Inc.*
  - [19] Lawless, J. E. (1982). *Statistical Models and Methods for Lifetime Data*. New York : John Wiley & Sons, Inc.
  - [20] Paul R. R. and Donald B. R. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1), 41–55.
  - [21] Ragnedda, M. (2017). *The Third Digital Divide: A Weberian Approach to Digital Inequalities*. New York: Routledge.
  - [22] Harris, E. K. and Albert, A. (1991). *Survivorship Analysis for Clinical Studies*. CRC Press.

## **논문투고 안내**

“데이터과학연구”에 게재할 논문을 연중 접수하고 있습니다. 게재를 원하시는 분은 다음 사항들을 참조하기 바랍니다.

1. 논문분야: 경영, 사회, 교육, 공학, 의·약학 등의 학문 분야에서 데이터 조사 및 분석을 활용한 논문을 폭넓게 게재한다.
2. 논문부수 및 접수방법: 20매 내외의 분량을 연구소 E-mail을 통하여 접수한다. (E-mail : data@cau.ac.kr)
3. 심사 및 수정: 본 논문집에 투고한 논문은 편집위원회의 심사를 거쳐 수정 및 보완과 게재 여부를 결정한다.
4. 발간: 본 논문집은 연간 2회(6월, 12월) 발행을 원칙으로 한다.
5. 논문심사 진행사항 문의: 연구소 조교 (전화 02-820-6352)

---

## **데이터과학연구 제11권 2022**

---

2022년 9월 29일 인쇄

2022년 9월 30일 발행

발행인 과 일 엽

편집인 이 주 영

발행처 **중앙대학교 데이터과학연구소**

서울특별시 동작구 흑석동 221

중앙대학교 경영경제대학 응용통계학과

전화 (02) 820-5499 팩스 (02) 814-5498

E-mail : data@cau.ac.kr

Homepage : <http://rcds.cau.ac.kr>

인쇄 주식회사 다컴애드

---