

이미지 데이터 증강 기술을 활용한 기계학습용 데이터셋 제작

소유진*, 강규창*

군산대학교*

so.npng98@gmail.com, kc.kang@kunsan.ac.kr

Dataset for Machine Learning using Image Data Augmentation

Yu-Jin So*, Kyuchang Kang*

Kunsan National University*

요약

본 구현은 기계학습용 데이터셋 제작에 대한 사례연구이다. 구현한 데이터셋은 수 백장 규모의 소수의 이미지를 다양한 이미지 증강 기술을 활용하여 다양한 속성의 이미지로 양적으로 확장하여 구축하였다. 구현한 데이터셋의 입력은 4종의 클래스를 가지는 총 560장의 이미지이고 데이터 증강을 통해 최종 구축된 데이터는 4종의 총 54,880장이다. 데이터셋에 대한 사용자 인터페이스는 딥러닝 학습에서 이미지 데이터셋으로 많이 활용되는 MNIST 데이터셋이 제공하는 인터페이스와 유사한 구조로 구현되었다.

I. 서론

일반적으로 딥러닝 기반 알고리즘의 학습 성능과 분류 정확도를 높이기 위해서 필수적으로 필요한 것은 데이터셋이다. 그러나 단순히 알고리즘 개발을 위한 데이터셋과는 달리 실제 산업 현장에서 필요로 하는 데이터는 양적으로 부족하고 특수한 형태의 형상이라 학계에서 일반적으로 사용되고 있는 이미지 데이터셋을 활용한 방법은 적절하지 않은 경우가 많다. 본 구현은 상용 자동차 부품 데이터셋 구축의 전 단계 사례연구로 소수의 데이터를 활용하여 다양한 이미지 변환을 통한 데이터 증강을 거쳐 기계학습용 데이터셋을 일 사례로 구축하고자 한다.

그러나 본 구현에서는 실 부품 이미지 확보에 어려움이 있어 연구실에서 보유한 이미지로 데이터 증강 기술에 대한 이론적 연구를 실시하고 데이터셋 구축 프로세스를 선 적용하여 데이터셋을 구축해보았다.

II. 데이터셋 구축 및 테스트

본 데이터셋 구축은 진단키트 영상으로 4종의 라벨을 가지는 560개의 데이터를 54,880개의 데이터로 증강시키는 기술적 구현에 관한 것이다.

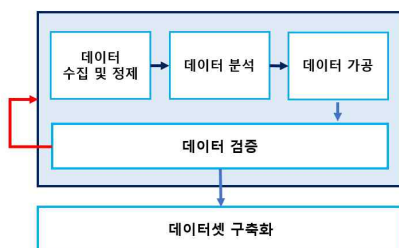


그림 1. 데이터셋 구축 과정 구성도

그림 1은 본 데이터셋 구축을 위한 구성도를 나타낸다. ‘데이터 수집 및 정제’ 단계에서는 데이터셋을 제작하기 위해 구축하고자 하는 데이터셋의 주제와 일치하는 데이터를 수집하고 데이터셋의 성능 향상을 고려하여 유의미한 데이터를 사용할 수 있도록 수집된 데이터들을 정제하는 역할을 수행한다. ‘데이터 분석’ 단계에서는 정제된 데이터를 가지고 학습데이터로 사용하기 위해 이미지를 분석하고, 이미지 증대 및 데이터 가공 방법

및 입력데이터 크기 및 형태를 정의하기 위해 분석하는 과정을 수행한다. ‘데이터 가공’ 단계에서는 분석된 결과를 바탕으로 이미지 증대를 포함한 전반적인 이미지 전처리를 한 후, 가공된 데이터를 데이터셋으로 사용하기 위해 CSV파일 형태로 저장하는 역할을 수행한다. ‘데이터 검증’ 단계에서는 데이터셋의 가공 여부를 판별하기 위해 데이터 성능평가를 진행하는 역할을 수행한다. ‘데이터셋 구축화’ 단계에서는 검증이 끝난 데이터셋을 구조화하여 최종적으로 데이터셋을 제작하는 역할을 수행한다.

본 구현에서는 미리 사전에 수집과 분석을 끝낸 준비한 데이터를 사용하였고, ‘데이터 수집 및 정제’와 ‘데이터 분석’ 단계를 제외한 다음 단계인 데이터 가공 및 데이터 검증 단계를 통해 이미지 증강 기술을 활용한 데이터셋을 구축하고자 한다.

본 구현에서 입력으로 사용되는 이미지 데이터는 중화항체반응을 검출해내는 진단키트 영상이다. 진단키트는 총 4종의 진단 반응을 표시하며 목표 검출량에 따라 Strong, Medium, Weak, Negative와 같은 4종의 진단 반응으로 구분된다. 진단키트 영상을 사용하기 전, 활용이 가능한 사진을 따로 분류하고 데이터셋 제작의 필수 요소에 따라 이미지를 분석한 결과에 따라 총 560장의 이미지를 선정하였다. 선정된 이미지는 이미지 증강 기술을 통해 각각 밝기, 대조, 노이즈, 회전 기술을 포함하여 각각을 조합한 총 8가지의 증대기술을 적용하였다[1].

이미지 밝기(Brightness)값의 기준은 기존 이미지의 화소 강도(Pixel Intensity)에서 -15%에서 15% 사이의 무작위 값으로 하여 화소 강도를 조절하였다. 이미지의 대조(Contrast)값 기준은 이미지의 흐릿한 부분을 선명하게 나타내기 위해 사용하였다. 화소 강도 값에 임의의 숫자를 곱한 후 화소 강도 값이 사전에 지정한 최소값보다 작을 경우 화소 강도 값을 최소값으로 설정하고, 화소 강도 값이 사전에 지정한 최대값보다 클 경우 화소 강도의 최대값으로 설정되게 하였다. 이미지 블러링(Blurring)은 이미지를 찍고 전송하는 과정에서 발생하는 정도를 유추하여 기준으로 하였다. Blur, GaussianBlur, MedianBlur 세가지 필터를 사용하였고, 흐림효과를 추가하는 Blur, GaussianBlur의 필터 값은 3x3, 노이즈를 제거하는 효과를 주는 MedianBlur는 필터 값을 3으로 설정하였다. 이미지 회전(Rotation)은 입력데이터 이미지가 단순하여 90도의 회전만 되도록 설정하였다.

표 1. 데이터셋 구축 결과

	Original	Brightness	Contrast	Blurring	Rotation	Brightness Contrast	Brightness Blurring	Contrast Blurring	Brightness Contrast Blurring	Result
Strong	138	138*5=690	138*5=690	138*3=414	1932	1380	690*3=2070	690*3=2070	1380*3=4140	13,524
Medium	138	138*5=690	138*5=690	138*3=414	1932	1380	690*3=2070	690*3=2070	1380*3=4140	13,524
Weak	138	138*5=690	138*5=690	138*3=414	1932	1380	690*3=2070	690*3=2070	1380*3=4140	13,524
Negative	146	146*5=730	146*5=730	146*3=438	2044	1460	730*3=2190	730*3=2190	1460*3=4380	14,308
total	560	2800	2800	1680	7840	5600	8400	8400	16800	54,880

본 구현을 통해 구축된 데이터셋은 총 54,880개이며 데이터 검증에 위해 마지막에 CSV 형태로 저장되고 데이터셋으로 구조화된다. 구조화한 데이터를 검증하기 위해 일반적인 컴퓨터 비전 라이브러리에서 제공하는 MNIST[2]와 같은 데이터셋의 활용법과 같은 인터페이스를 제공하고자 아래와 같은 구조와 같이 간단한 클래스를 정의하여 구현하였다.

```

1 class load_data():
2     def __init__(self, file, height, width):
3         self.file = file
4         self.height = height
5         self.width = width
6
7     def data(self):
8         image = []
9         labeling = []
10
11         data_file = open(self.file, 'r')
12         data_list = data_file.readlines()
13         data_file.close()
14
15         for record in data_list:
16             all_values = record.split(',')
17             labeling.append(all_values[0])
18             image_data = np.asarray(all_values[1:-1]).reshape
19                 ((self.height,self.width))
20             image.append(image_data)
21
22         labeling = np.asarray(labeling[:])
23         image = np.asarray(image[:])
24         return image, labeling

```

그림 2. 데이터셋 인터페이스 예시

구축한 데이터셋의 유효성을 확인하기 위해 MLP(Multi-Layer Perceptron) 및 CNN(Convolutional Neural Network) 기반으로 단순한 구조의 분류기를 구축하여 확인하였다. MLP 방식의 경우 95% 정도의 정확도를 나타내었고, CNN 방식의 경우 99%의 정확도를 나타내었다. 그림 2는 MLP 방식으로 실험하였을 때의 학습 곡선을 나타냈고, 표 2는 CNN 기반으로 실험하였을 때의 4종 클래스 각각에 대한 정확도 결과이다.

구축된 데이터는 컴퓨터 비전에서 많이 사용되는 MNIST보다 단순한 구조의 데이터라 기본적인 학습 네트워크에서도 높은 정확도의 결과가 나오는 것으로 판단된다.

III. 결과 및 고찰

표 1은 본 사례연구로 진행된 데이터셋의 구축 결과이다. 구현한 데이터셋의 입력은 4종의 클래스를 가지는 총 560장의 이미지이고 데이터 증강을 통해 최종 구축된 데이터는 4종의 총 54,880장이다.

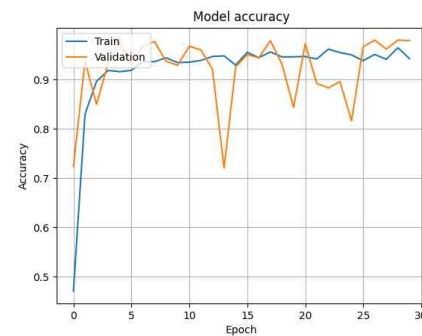


그림 2. MLP 테스트 결과

표 2. CNN 테스트 결과

Data	Accuracy
Strong	99.95%
Medium	99.95%
Weak	99.28%
Negative	100%

본 데이터셋 구축은 자동차 부품의 다양화로 이미지 인식을 통해 부품의 종류를 자동 검출하고 이를 기반으로 3D 스캐너와 같은 부품 검사 장비의 자동화에 활용할 목적으로 기획되었으나 실 부품 이미지 확보에 어려움이 있어 연구실에서 보유한 이미지로 이미지 데이터 증강에 대한 사례 연구를 먼저 실시하였다. 현재 실 부품 이미지 수집 장치에 대한 프로토타입 구현이 병행 진행중이므로 본 구현의 사례를 실 획득한 이미지로 변경하여 같은 절차로 데이터셋 구축이 진행될 예정이다.

ACKNOWLEDGMENT

본 논문은 한국산업단지공단 지원 두이엔지(주) 주관 연구사업(과제번호: IRJB2205)의 지원에 의해 이루어진 것임

참 고 문 헌

- [1] M Xu, S Yoon, A Fuentes, DS Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning, - arXiv preprint arXiv:2205.01491, 2022 - arxiv.org
- [2] Deng, L., "The mnist database of handwritten digit images for machine learning research". IEEE Signal Processing Magazine, 29(6), 141 - 142, 2012