

고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘 연구

이협건*, 김영운

Study of Improved CNN Algorithm for Object Classification Machine Learning of Simple High Resolution Image

Hyeopgeon Lee*, Young-Woon Kim

요약 CNN(Convolutional Neural Network) 알고리즘은 인공지능망 구현에 활용되는 대표적인 알고리즘으로 기존 FNN(Fully connected multi layered Neural Network)의 문제점인 연산의 급격한 증가와 낮은 객체 인식률을 개선하였다. 그러나 IT 기기들의 급격한 발달로 최근 출시된 스마트폰 및 태블릿의 카메라에 촬영되는 이미지들의 최대 해상도는 108MP로 약 1억 8백만 화소이다. 특히 CNN 알고리즘은 고해상도의 단순 이미지를 학습 및 처리에 많은 비용과 시간이 요구된다. 이에 본 논문에서는 고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘을 제안한다. 제안하는 알고리즘은 고해상도의 이미지들의 학습모델 생성 시간을 감소하기 위해 CNN 알고리즘의 풀링 계층의 Max Pooling 알고리즘 연산을 위한 인접 행렬 값을 변경한다. 변경한 행렬 값마다 4MP, 8MP, 12MP의 고해상도 이미지들의 처리할 수 있는 학습 모델들을 구현한다. 성능평가 결과, 제안하는 알고리즘의 학습 모델의 생성 시간은 12MP 기준 약 36.26%의 감소하고, 학습 모델의 객체 분류 정확도와 손실률은 기존 모델 대비 약 1% 이내로 오차 범위 안에 포함되어 크게 문제가 되지 않는다. 향후 본 연구에서 사용된 학습 데이터보다 다양한 이미지 종류 및 실제 사진으로 학습 모델을 구현한 실질적인 검증이 필요하다.

Abstract A convolutional neural network (CNN) is a representative algorithm for implementing artificial neural networks. CNNs have improved on the issues of rapid increase in calculation amount and low object classification rates, which are associated with a conventional multi-layered fully-connected neural network (FNN). However, because of the rapid development of IT devices, the maximum resolution of images captured by current smartphone and tablet cameras has reached 108 million pixels (MP). Specifically, a traditional CNN algorithm requires a significant cost and time to learn and process simple, high-resolution images. Therefore, this study proposes an improved CNN algorithm for implementing an object classification learning model for simple, high-resolution images. The proposed method alters the adjacency matrix value of the pooling layer's max pooling operation for the CNN algorithm to reduce the high-resolution image learning model's creation time. This study implemented a learning model capable of processing 4, 8, and 12 MP high-resolution images for each altered matrix value. The performance evaluation result showed that the creation time of the learning model implemented with the proposed algorithm decreased by 36.26% for 12 MP images. Compared to the conventional model, the proposed learning model's object recognition accuracy and loss rate were less than 1%, which is within the acceptable error range. Practical verification is necessary through future studies by implementing a learning model with more varied image types and a larger amount of image data than those used in this study.

Key Words : CNN, Pooling Algorithm, Object Classification, Machine Learning Model

*Dept. of Big Data, Seoul Gangseo Campus of Korea Polytechnics College

Corresponding Author : Dept. of Big Data, Seoul Gangseo Campus of Korea Polytechnics College (hglee67@kopo.ac.kr)

Received January 26, 2023

Revised January 31, 2023

Accepted February 14, 2023

1. 서론

IT기술을 발달로 인해 최근 스마트 폰 및 태블릿 등 스마트 기기의 카메라 성능은 매년 급격하게 발달하고 있으며, 최근 108MP로 약 1억 8백만 화소를 기반 높은 해상도의 이미지 파일들이 생성되고 있다.

객체 분류에 활용되는 다양한 인공지능 신경망 알고리즘들은 이미지들의 픽셀 정보를 수치화하여 학습 모델을 생성한다. 특히 인공 신경망에 많이 활용되는 CNN(Convolutional Neural Network) 알고리즘[1]은 기존 FNN(Fully connected multi layered Neural Network)[2]에 비해 연산을 감소시키고, 높은 객체 인식률을 보이는 알고리즘이다.

그러나 CNN 알고리즘은 최근 카메라 기술의 발달로 인해 높아진 고해상도의 이미지를 처리함에 많은 처리 시간이 요구된다. 또한 특히 이미지 데이터의 특징을 잡기 어려운 높은 수준의 객체 분류가 아닌 높은 해상도의 단순 이미지(문자 및 배경과 분류할 객체의 경계가 명확한 이미지)의 객체 분류는 기존 CNN 알고리즘을 통한 학습 모델을 생성함에 있어 생성 속도에 부적합하다. 따라서 고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘이 필요하다.

이에 본 논문에서는 고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘을 제안한다.

제안하는 기법은 고해상도의 이미지들의 학습모델 생성 시간을 감소하기 위해 CNN 알고리즘의 풀링 계층의 Max Pooling 알고리즘의 연산을 위한 인접 행렬 값을 변경한다. 변경한 행렬 값마다 4MP, 8MP, 12MP의 고해상도 이미지들의 처리할 수 있는 학습 모델들을 구현한다. 또한 구현한 학습 모델들은 적합성, 생성 시간, 정확도, 손실률을 평가한다.

성능평가 결과, 제안하는 알고리즘으로 구현된 학습 모델의 생성 시간은 12MP 기준 약 36.26%의 감소하고, 학습 모델의 객체 분류 정확도와 손실률은 기존 모델 대비 약 1% 이내로 오차 범위 안에 포함되어 크게 문제가 되지 않는다.

본 논문의 구성은 다음과 같다. 2장에서는 CNN 알

고리즘에 대해 살펴보고, 제안하는 알고리즘에 필요한 요구사항을 도출한다. 3장에서는 본 논문에서 제안하는 알고리즘을 기반으로 학습 모델 및 학습 모델의 적합성을 평가하며, 4장에서는 제안한 알고리즘으로 생성된 학습 모델의 생성 시간, 정확도, 손실률을 분석한다. 마지막 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

본 장에서는 제안하는 알고리즘에 활용되는 CNN 알고리즘 이해 및 구성, FNN 알고리즘에 대해 살펴보고, 단순 이미지의 객체 분류에 대한 요구사항을 도출한다.

2.1 CNN 알고리즘 이해 및 구성

인공지능 기술 중 객체 분류는 광범위하게 활용되는 대표적인 기술이다. 객체 분류의 대표적인 머신러닝 알고리즘은 CNN으로 이미지를 분석하기 위한 패턴을 발견하는데 적합한 알고리즘이다. <그림 1>은 CNN 알고리즘의 구조를 나타낸다.

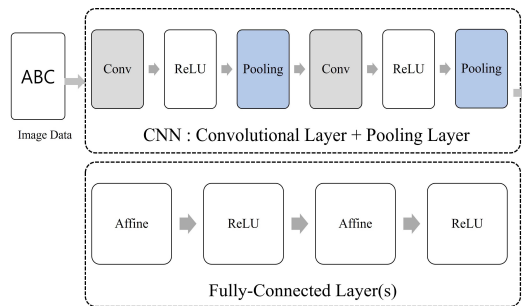


그림 1. CNN 알고리즘 구조
Fig. 1. Structure of the CNN algorithm

CNN 알고리즘[3][4]의 구성은 크게 이미지 데이터들의 특징을 추출하는 합성곱 계층(Convolution Layer)과 추출된 특징을 Sub-Sampling하는 풀링 계층이다. 합성곱 계층의 주요 역할은 이미지에 필터링 기법을 적용하여 이미지 분류를 보다 정확하게 할 수 있게 돕는다. 필터링 기법은 이미지의 테두리 부분을 추출,

이미지의 명암 비를 처리한다. 풀링 계층은 인접 데이터들을 대표적인 값으로 변환하고, 그 값을 기반으로 처리하도록 하여 데이터 연산의 크기를 감소시키는 등 다양한 기능을 수행한다.

일반적인 신경망 알고리즘의 구조[5]는 Affine 연산과 ReLU와 같은 비선형 활성화 함수(Nonlinear Activation Function)의 합성하여 여러 계층을 쌓은 구조이다. 즉, CNN 알고리즘은 앞서 설명한 합성곱계층과 풀링 계층을 일반 신경망에서 활용되는 Fully Connected Layer를 처리하기 전에 추가한 구조이다.

2.2 CNN과 FNN 알고리즘과 비교

CNN 알고리즘이 연구 개발된 전 주로 사용되던 FNN 알고리즘은 신경망으로 학습시키는 방법이다.

FNN 알고리즘[2][6]은 가장 큰 문제점은 인접 픽셀들의 상관관계를 무시한다. 일반적인 이미지 데이터들은 인접한 픽셀들의 상관관계가 매우 높지만, FNN 알고리즘 처리 방법은 입력받는 이미지 데이터들을 벡터 형태로 변환하여 입력받기 때문에 상관관계와 상관없이 벡터화(Vectorization) 과정에서 데이터의 손실이 발생한다.

FNN 알고리즘의 다른 문제점[7]은 막대한 양의 연산 요구이다. 만약 1,000*1,000 해상도의 이미지를 처리한다면, 벡터의 차원의 수는 약 300만개(1,000 * 1,000 * 3)이다. 300만개의 차원을 처리하기 위해서는 방대한 양의 뉴런이 요구된다. 따라서 인공신경망 전체의 연산의 수는 급격하게 증가하게 된다.

CNN 알고리즘은 앞서 설명한 FNN의 정보손실 정보 손실의 최소화하기 위해 입력받는 이미지 데이터를 행렬 형태로 입력받는다. 따라서 CNN 알고리즘은 FNN 알고리즘에서 발생하는 벡터화 과정에서 발생하는 데이터 손실을 방지한다.

급격하게 증가되는 연산 문제는 마지막에 실행되는 계층인 Fully-Connected Layer에서 Model Parameter 수가 많은 원본 이미지 데이터가 아닌 풀링 계층(Pooling Layer)에 의해 축소된 이미지 데이터를 처리한다. 풀링 계층은 Stride에 비해 정확도는 낮지만, Model Parameter 수를 증가시키지 않고, 연산 및 학습량을 감소시키는 장점을 갖는다.

따라서 CNN 알고리즘은 기존 FNN 알고리즘의 문제점을 개선하여, 방대한 양의 데이터 학습 및 높은 수준의 학습 등 다양한 영역에 활용된다.

2.3 요구사항 도출

본 절에서는 제안하는 개선된 CNN 알고리즘 구현을 위한 요구사항을 도출한다.

최근 스마트폰 및 태블릿 등 IT 기기들의 성능은 매년 급격히 발달한다. 최근 스마트폰 및 태블릿의 카메라에 촬영되는 이미지들의 최대 해상도는 108MP로 약 1억 8백만 화소이다. 전면 및 후면 카메라 평균 해상도는 12MP로 약 100만 화소이다.

앞서 설명한 CNN 알고리즘은 풀링 계층을 통해 처리해야할 연산 및 학습량을 감소시킨다. 풀링 계층에 적용되는 풀링 알고리즘의 기본 영역 크기는 2X2 행렬 값이다. 바로 인접한 픽셀들의 값에 최대, 최소, 평균 등 계산한다. CNN 알고리즘은 FNN 알고리즘에 비해 연산이 현저히 감소하지만, 최근 카메라 기술의 발달로 인해 높아진 해상도의 이미지를 처리함에 많은 처리 시간이 요구된다.

특히 이미지 데이터의 특징을 잡기 어려운 높은 수준의 객체 분류가 아닌 높은 해상도의 단순 이미지(문자 및 배경과 분류할 객체의 경계가 명확한 이미지)의 객체 분류는 기존 CNN 알고리즘을 통한 학습 모델을 생성함에 있어 생성 속도에 부적합하다. 따라서 고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘이 필요하다.

3. 제안하는 개선된 CNN 알고리즘

본 장에서는 앞서 요구사항 분석을 결과를 기반으로 고해상도 단순 이미지의 객체 분류 학습모델 구현을 위한 개선된 CNN 알고리즘을 제안한다.

3.1. 학습 데이터

제안하는 알고리즘을 구현하기 위한 학습 데이터는 고해상도 단순 이미지로 문자 이미지 데이터를 사용한다.

〈그림 2〉는 학습 모델 구현을 위한 학습 데이터 예를 나타낸다.

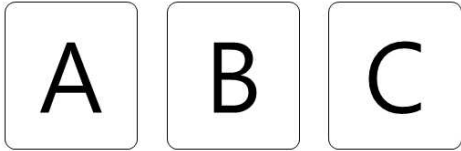


그림 2. 학습 모델 구현을 위한 학습 데이터 예
Fig. 2. Example of the learning data for machine learning model

학습 데이터는 높은 해상도의 이미지로 4MP(2,688×1,520), 8MP(3,840×2,160), 12MP(4,032×3,024) 3가지를 동일한 이미지로 1,000장씩 해상도만 변경하여 사용한다. 학습에 사용된 분류는 대문자 A부터 Z까지 총 10가지로 각 문자별 100장씩 학습 및 검증에 활용한다.

3.2. 제안하는 알고리즘

CNN 알고리즘은 합성곱 계층과 풀링 계층으로 구성되며, 풀링 계층은 인접 데이터들을 대표적인 값으로 변환하고, 그 값을 기반으로 학습 및 분석을 수행하여 연산 및 학습량을 감소한다.

풀링 계층에 사용되는 알고리즘은 Max Pooling, Min Pooling, Average Pooling 알고리즘으로 인접한 행렬의 값을 최대, 최소, 평균값으로 추출하며, 일반적으로 Max Pooling 알고리즘을 사용한다.

Max Pooling 알고리즘은 인접한 행렬 2×2의 값 중 최댓값을 반환하여 인공지능망 전체의 연산 및 학습량을 감소시킨다. 〈그림 3〉은 Max Pooling 알고리즘의 예를 나타낸다.

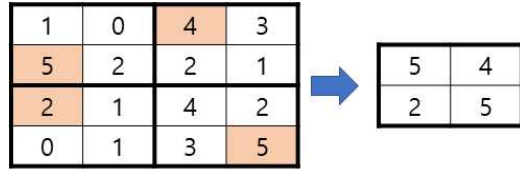


그림 3. Max Pooling 알고리즘의 예
Fig. 3. Example of the max pooling algorithm

기존 Max Pooling 알고리즘의 인접 행렬의 범위는 2×2로 고해상도의 단순 이미지의 학습에 많은 연산을 요구한다.

제안하는 알고리즘은 학습 모델 생성에 필요한 이미지들의 해상도에 따라 Max Pooling의 인접 행렬 범위를 유동적으로 변경이 필요하다. 제안하는 인접 행렬의 범위는 3×3과 4×4 범위를 추가하여 기존 2×2를 포함하여 총 3가지를 적용한다. 추가한 범위에 따른 높은 해상도별 학습 모델을 구현하고, 학습 모델 적합성과 성능 분석한다.

3.3. 제안하는 알고리즘의 학습모델 구현

학습 모델 구현은 인공지능 플랫폼인 TensorFlow를 이용한다. Max Pooling 알고리즘의 행렬 값을 변경은 TensorFlow의 layers에 정의된 Pooling 객체의 생성자의 pool_size 변수의 값을 개발 도구(PyCharm)를 통해 수정하여 구현한다. 〈그림 4〉는 3×3 설정된 Pooling 객체의 생성자 함수를 나타낸다.

```
def __init__(self,
              pool_size=(3, 3), # Pool Size =>3,3
              strides=None,
              padding='valid',
              data_format=None,
              **kwargs):
    super(MaxPooling2D, self).__init__(
```

그림 4. 3×3 설정된 Pooling 객체의 생성자 함수
Fig. 4. Constructor method of pooling object

Pooling 객체의 pool_size 변경은 기본 값인 2×2와 제안하는 3×3, 4×4를 수행한다. 3×3과 4×4와 같이 해상도에 나뉘지 않는 이미지는 인접 행렬 나머지 값을 무시하도록 설정한다. 〈그림 5〉는 제안하는

알고리즘을 구현한 학습 모델 구조의 의사 코드를 나타낸다.

```

1  model = Sequential([
2  // 이미지 회전, 플립, 확대하여 데이터 증가
3  layers.experimental.preprocessing.RandomFlip
4  RandomRotation, RandomRotation
5  // 이미지 전처리
6  layers.experimental.preprocessing.Rescaling
7
8  // Convolution Layer 정의
9  layers.Conv2D(16, 3, padding="same",
10                 activation="relu"),
11 // Max Pooling 정의
12 layers.MaxPooling2D()
13 // Convolution Layer와 Max Pooling 정의
14 layers.Conv2D(32, 3, padding="same",
15                 activation="relu"),
16 layers.MaxPooling2D(),
17 // Convolution Layer와 Max Pooling 정의
18 layers.Conv2D(64, 3, padding="same",
19                 activation="relu"),
20 layers.MaxPooling2D()
21 // 데이터 증가를 위한 Dropout
22 layers.Dropout(0.2),
23 // 신경망 알고리즘, 활성화 함수 ReLU
24 layers.Flatten(),
25 layers.Dense(128, activation="relu"),
26 layers.Dense(num_classes)
27 ])
    
```

그림 5. 제안하는 알고리즘을 구현한 학습 모델 구조의 의사 코드

Fig. 5. Pseudo code of the machine learning model layer for proposal algorithm

학습 데이터는 전체 데이터의 80%인 800장의 이미지만 가지고 학습 모델을 구현하면 데이터 부족으로 인한 과적합(Overfitting) 현상이 발생한다. 이로 인해 학습 모델을 구현할 때 과적합 현상을 해결하기 위해 이미지를 랜덤하게 회전, 확대, 플립을 수행하고, Dropout을 정의한다. 또한 학습 횟수는 15회 수행하도록

정의한다.

학습 모델에 활용된 활성화 함수는 ReLU를 사용하고, 학습 모델이 올바르게 생성되었는지 확인하기 위해 옵티마이저는 Adam, 평가 모델은 정확도로 정의한다.

3.4. 학습 모델 적합성 평가 결과

앞서 제시한 학습 모델의 구현이 올바른지 적합성을 평가하기 위해 학습 모델 생성 결과를 검증한다. 검증 방법은 학습 모델 평가 지표인 정확도와 손실률을 분석한다. 검증 데이터는 학습 데이터의 20%인 200장의 이미지로 정의하고, 학습 데이터는 나머지 80%인 800장의 이미지로 학습 모델을 구현한다.

학습 모델 검증 결과, 학습 모델의 검증용 데이터의 정확도와 학습용 데이터의 정확도의 그래프의 간격의 차이가 많이 발생되지 않고, 유사한 형태로 결과 값이 측정된다. 또한 학습 모델의 검증용 데이터의 손실률은 정확도와 유사하게 검증용 데이터의 정확도와 학습용 데이터의 그래프의 간격의 차이가 많이 발생되지 않고, 유사한 형태로 결과 값이 측정된다. 따라서 본 논문에서 제시한 학습 모델 구현은 올바르게 구현되었음을 나타낸다.

4. 성능평가

본 장에서는 제안하는 개선된 CNN 알고리즘의 성능평가를 위해 학습모델 생성 시간, 생성된 학습 모델의 정확도, 손실률을 분석한다.

4.1 실험 환경

실험환경은 개선된 CNN 알고리즘이 적용된 학습 모델 생성을 위해 GPU 기반 워크스테이션 1대를 활용한다. GPU 기반 워크스테이션의 성능은 CPU 4.0GHz 6Core, 64GB(DDR4 3200), NVidia RTX A4000 16GB 4DP Nvidia 그래픽카드, SSD 512GB이다.

학습모델 구현에 활용한 인공지능 플랫폼은 NVIDIA Ampere 아키텍처를 지원 및 Cuda 환경에서 동작이 가능한 TensorFlow 2.5를 사용한다. 학습 모델 생성을 위한 학습 횟수는 15회이며, 배치 크기는 128로

정의한다. 풀링 알고리즘은 인접한 값 중 최대값을 가져오는 Max Pooling을 사용하고, 크기는 기본 값인 2×2와 3×3, 4×4로 증가하며 분석한다.

학습데이터는 앞서 3장에서 제시한 높은 해상도의 이미지로 4MP, 8MP, 12MP 3가지를 동일한 이미지로 1,000장씩 해상도만 변경하여 사용한다

〈표 1〉은 제안하는 개선된 CNN 알고리즘의 성능 평가를 위한 실험 환경을 나타낸다.

표 1. 실험 환경

Table 1. Experiment environment

Item	Value
CPU	4.0Ghz 6Core
RAM	64GB(DDR4 3200)
GPU	NVidia RTX A4000 16GB 4DP
SSD	512G
AI Platform	TensorFlow 2.5
GPU Architecture	NVIDIA Ampere
Epochs	15
Batch Size	128
Pooling Algorithm	Max Pooling
Pooling Size	2×2, 3×3, 4×4
Image Resolution	4, 8, 12MP
Image Count	1,000

4.2 학습 모델 생성 시간 분석

학습모델 생성 방법은 앞서 3장에서 제시한 학습모델 구현에 맞춰 학습모델을 생성한다. 학습 횟수는 15회, 배치 크기는 128로 정의하여 학습모델을 생성한다. 학습데이터는 동일한 이미지 1,000장씩 4MP, 8MP, 12MP로 해상도만 다르게 생성된 단순 문자 이미지이다. 학습용 데이터와 검증용 데이터셋 비중은 80:20으로 800장이 학습용 데이터, 200장이 검증용 데이터이다.

학습모델 생성 시간 측정 방법은 학습모델 구현을 위한 파이썬 파일이 실행된 시간부터 학습모델이 생성

이 완료된 시간을 측정하여 생성시간을 구한다. [표 2]와 〈그림 6〉 이미지 해상도별 풀링 크기에 따른 학습 모델 생성 시간 분석 결과를 나타낸다.

표 2. 이미지 해상도별 풀링 크기에 따른 학습모델 생성 시간 분석 결과

Table 2. Result of the create time of machine learning model by pooling size

(단위 : 분)

Pooling Size	2×2	3×3	4×4
4MP	41.12	36.56	29.35
8MP	90.56	75.19	63.41
12MP	143.43	108.34	91.41

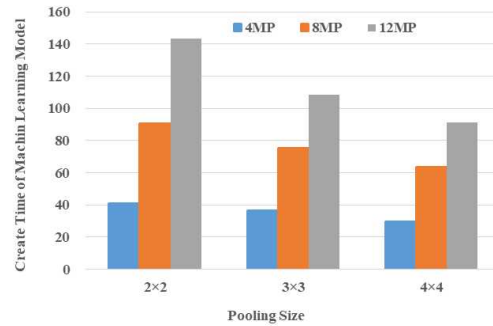


그림 6. CNN 알고리즘 구조

Fig. 6. The structure of the CNN algorithm

분석 결과, 해상도 가장 높은 12MP 이미지의 학습 모델 생성시간이 최대 143분 43초 가장 오래 걸렸지만, 풀링 크기를 4×4로 증가시키면 91분 41초로 약 36.26% 학습모델 생성시간이 감소한다.

또한 이미지 해상도가 8MP와 4MP인 학습 모델 생성 시간은 풀링 크기가 커질수록 감소한다. 8MP 이미지 학습 모델 생성시간은 풀링 크기 4×4에서 90분 56초, 풀링 크기 2×2에서 63분 41초로 약 29.98% 감소한다. 4MP 이미지 학습 모델 생성시간은 풀링 크기 4×4에서 41분 12초, 풀링 크기 2×2에서 29분 35초로 약 28.62% 감소한다.

이와 같은 결과가 측정된 이유는 학습 모델을 생성

할 때 필요한 연산 및 학습량이 풀링 크기에 따라 감소 되기 때문이다.

4.3 생성된 학습 모델의 정확도 분석

학습 모델의 정확도 분석은 생성된 학습 모델의 적합성을 평가하는 지표로 중요하다. 생성된 학습 모델의 낮은 정확도는 학습 모델로서 가치가 없기 때문이다. 본 절에서는 앞서 분석한 학습 모델의 정확도를 분석한다. 정확도 분석 방법은 학습모델의 생성(Model Compile)할 때 평가지표를 정확도(Accuracy)로 정의하고 학습 횟수별 결과를 생성하여 분석한다. [표 3]와 <그림 7>은 이미지 해상도별 풀링 크기에 따른 학습모델의 정확도 분석 결과를 나타낸다.

표 3. 이미지 해상도별 풀링 크기에 따른 학습모델의 정확도 분석 결과
Table 3. Result of accuracy of the machine learning model by pooling size

Pooling Size	2×2	3×3	4×4
4MP	99.28	96.51	98.21
8MP	98.94	98.18	98.92
12MP	97.68	97.27	96.22
AVG.	98.00	97.32	97.78

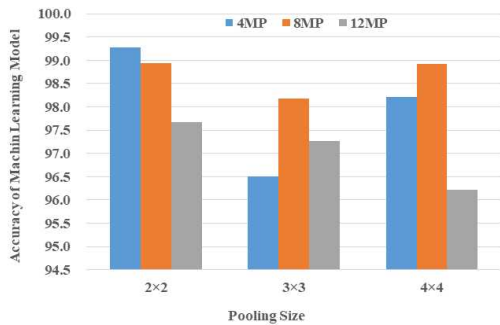


그림 7. 이미지 해상도별 풀링 크기에 따른 학습모델의 정확도 분석 결과

Fig. 7. Result of accuracy of the machine learning model by pooling size

분석 결과, CNN 알고리즘의 기본 값인 2×2 풀링 크기는 4MP에서 99.28%, 8MP에서 98.94%, 12MP에서 97.68%의 정확도로 분석되며, 평균 98%의 정확도를 가진다. 3×3 풀링 크기는 4MP에서 96.51%, 8MP에서 98.18%, 12MP에서 97.27%의 정확도로 분석되며, 평균 97.32%의 정확도를 가진다. 4×4 풀링 크기는 4MP에서 98.21%, 8MP에서 98.92%, 12MP에서 96.22%의 정확도로 분석되며, 평균 97.78%의 정확도를 가진다.

학습 데이터는 해상도가 다른 동일한 이미지이지만, 학습 모델의 정확도는 조금씩 차이가 발생한다. 이러한 이유는 학습 데이터 부족으로 발생하는 과적합(Overfitting) 문제를 해결하기 위해 학습 모델 구현에 Dropout를 적용하였기 때문이다. Dropout으로 적용에 따른 학습 모델의 정확도 오차 범위를 약 1% 이내로 가정하면, 이미지의 해상도 크기에 학습 모델의 정확도는 크게 문제되지 않는다.

또한 풀링 크기 변경에 따른 정확도는 2×2 풀링 크기에서 98%, 3×3 풀링 크기에서 97.32%, 4×4 풀링 크기에서 97.78%로 앞서 가정한 1%의 오차 범위에 포함되어 풀링 크기에 따른 학습 모델 정확도는 크게 문제되지 않는다.

4.4 생성된 학습 모델의 손실률 분석

학습 모델의 손실률 분석은 생성된 학습 모델의 적합성을 평가하는 지표로 중요하다. 손실률은 손실 함수(Loss Function)로부터 수치화된 값으로 오차가 클수록 값이 크고, 오차가 작을수록 값이 작아진다. 따라서 높은 손실률은 학습 모델로서 가치가 없다. 손실률 분석을 위해 사용한 손실 함수는 TensorFlow에서 제공되는 tf.keras.losses.categorical_crossentropy를 사용하여 분석한다.

[표 4]와 <그림 8>은 이미지 해상도별 풀링 크기에 따른 학습모델의 손실률 분석 결과를 나타낸다.

표 4. 이미지 해상도별 풀링 크기에 따른 학습모델의 손실률 분석 결과

Table 4. Result of the loss rate of machine learning model by pooling size

Pooling Size	2×2	3×3	4×4
4MP	0.013	0.035	0.019
8MP	0.011	0.017	0.011
12MP	0.024	0.026	0.037
AVG.	0.016	0.026	0.022

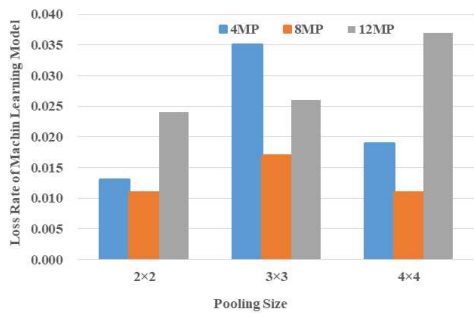


그림 8. 이미지 해상도별 풀링 크기에 따른 학습모델의 손실률 분석 결과
 Fig. 8. Result of the loss rate of machine learning model by pooling size

분석 결과, 2×2 풀링 크기는 4MP에서 0.013, 8MP에서 0.011, 12MP에서 0.024의 손실률을 측정되며, 평균 0.0216%의 손실률을 가진다. 3×3 풀링 크기는 4MP에서 0.035%, 8MP에서 0.017, 12MP에서 0.026의 손실률로 분석되며, 평균 0.026%의 손실률을 가진다. 4×4 풀링 크기는 4MP에서 0.019, 8MP에서 0.011, 12MP에서 0.037의 손실률로 분석되며, 평균 0.022의 손실률을 가진다.

측정된 전체 손실률 평균은 0.022로 2×2 풀링 크기에서 평균 이하의 손실률을 측정되었지만, 학습 모델 생성에 적용된 Dropout를 고려하면 풀링 크기 따른 손실률 차이는 거의 없다.

5. 결론

제안하는 기법은 고해상도의 이미지들의 학습모델 생성 시간을 감소하기 위해 CNN 알고리즘의 풀링 계층의 Max Pooling 알고리즘의 연산을 위한 인접 행

렬 값을 변경한다. 변경한 행렬 값마다 4MP, 8MP, 12MP의 고해상도 이미지들의 처리할 수 있는 학습 모델들을 구현한다. 또한 구현한 학습 모델들은 적합성, 생성 시간, 정확도, 손실률을 평가한다.

성능평가 결과, 제안하는 알고리즘으로 구현된 학습 모델의 생성 시간은 12MP 기준 약 36.26%의 감소하고, 학습 모델의 객체 분류 정확도와 손실률은 기존 모델 대비 약 1% 이내로 오차 범위 안에 포함되어 크게 문제가 되지 않는다.

향후 본 연구에서 사용된 학습 데이터보다 다양한 이미지 종류와 방대한 양의 이미지 데이터로 학습 모델을 구현하여 실질적인 검증이 필요하다.

REFERENCES

- [1] Kwang-Man Ko. "A DDoS Attack Detection Technique through CNN Model in Software Define Network" Journal of Korea Institute of Information, Electronics, and Communication Technology 13, 6 (2020) : 605-610
- [2] Jong-Ho Choi."Binary CNN Operation Algorithm using Bit-plane Image"Journal of Korea Institute of Information, Electronics, and Communication Technology 12,6(2019): 567-572.
- [3] Su-Rak Son, Byung-Kwan Lee, Son-Kweon Sim, Yi-Na Jeong."A Design of Passenger Detection and Sharing System(PDSS) to support the Driving (Decision) of an Autonomous Vehicles"Journal of Korea Institute of Information, Electronics, and Communication Technology13,2(2020): 138-144.
- [4] Jeonghwan Hwang, Ji Won Yoon."Random Noise Addition for Detecting Adversarially Generated Image Dataset"Journal of Korea Institute of Information, Electronics, and Communication Technology12,6(2019): 629-635.
- [5] Fang-Jing Shen, Jian-Hao Chen, Wei-Yen Wang, Dien-Lin Tsai, Lien-Chieh Shen, Ching-Tung Tseng."A CNN-Based Human Head Detection Algorithm Implemented on

Edge AI Chip" International Conference on System Science and Engineering (ICSSE), (2020)

[6] Zubaida Said Ameen, Abdullahi Umar Ibrahim, Auwalu Saleh Mubarak, Fadi Al-Turjman."An improved CNN-LSTM deep model for Classification of guideRNA in CRISPR-Cas12 System" International Conference on Artificial Intelligence in Everything (AIE), (2022)

[7] Hammou Djalal Rafik."Classification and detection of covid-19 in human respiratory lungs using convolutional neural network architectures" International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP), (2021)

저자약력

이 협 건 (Hyeopgeon Lee)

[중신회원]



- 2011.03 - 2015.08, 숭실대학교 일반대학원 컴퓨터학과 공학박사
- 2015.12 - 현재, 한국폴리텍대학 서울강서 캠퍼스 빅데이터과 교수

〈관심분야〉 빅데이터처리, 빅데이터, 클라우드컴퓨팅
MSA 프로그래밍 등

김 영 운 (Young-Woon Kim)

[중신회원]



- 2004.09 - 2018.08, 숭실대학교 일반대학원 컴퓨터학과 공학박사
- 2015.12 - 현재, 한국폴리텍대학 서울강서 캠퍼스 빅데이터과 교수

〈관심분야〉 빅데이터, 소프트웨어 로봇 프로그래밍,
클라우드컴퓨팅