The Transactions of the Korea Information Processing Society
TKIPS Vol.13, No.9, pp.444~452

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.9.444

A Study on the Optimization of Fire Awareness Model Based on Convolutional Neural Network: Layer Importance Evaluation-Based Approach

Won Jin† · Mi-Hwa Song††

ABSTRACT

This study proposes a deep learning architecture optimized for fire detection derived through Layer Importance Evaluation. In order to solve the problem of unnecessary complexity and operation of the existing Convolutional Neural Network (CNN)-based fire detection system, the operation of the inner layer of the model based on the weight and activation values was analyzed through the Layer Importance Evaluation technique, the layer with a high contribution to fire detection was identified, and the model was reconstructed only with the identified layer, and the performance indicators were compared and analyzed with the existing model. After learning the fire data using four transfer learning models: Xception, VGG19, ResNet, and EfficientNetB5, the Layer Importance Evaluation technique was applied to analyze the weight and activation value of each layer, and then a new model was constructed by selecting the top rank layers with the highest contribution. As a result of the study, it was confirmed that the implemented architecture maintains the same performance with parameters that are about 80% lighter than the existing model, and can contribute to increasing the efficiency of fire monitoring equipment by outputting the same performance in accuracy, loss, and confusion matrix indicators compared to conventional complex transfer learning models while having a learning speed of about 3 to 5 times faster.

Keywords : Layer Importance Evaluation, Transfer Learning Model, CNN Optimization, Real-Time Fire Detection, Contribution

합성곱 신경망 기반 화재 인식 모델 최적화 연구: Layer Importance Evaluation 기반 접근법

진 원† · 송 미 화††

요 약

본 연구는 Layer Importance Evaluation을 통해 도출된 화재 감지에 최적화된 딥러닝 아키텍처를 제안한다. 기존의 합성곱 신경망(Convolutional Neural Network, CNN) 기반 화재 감지 시스템의 불필요한 복잡성과 연산을 초래하는 문제점을 해결하기 위해, Layer Importance Evaluation 기법을 통해 가중치 및 활성화 값에 근거한 모델의 내부 레이어의 동작을 분석하고, 화재 감지에 기여도가 높은 레이어를 식별한 뒤, 식별한 레이어만으로 모델을 재구성하여, 기존 모델과의 성능 지표를 비교 분석하였다. Xception, VGG19, ResNet, EfficientNetB5 등 네 가지 전이 학습 모델을 사용하여 화재 데이터를 학습시킨 후, Layer Importance Evaluation기법을 적용하여 각 레이어의 가중치와 활성화 값을 분석한 뒤 기여도가 가장 높은 상위 랭크 레이어들을 선별하여 새로운 모델을 구축하였다. 연구 결과, 구현된 아키텍처는 기존 모델 대비 약 80% 가량 경량화 된 파라미터로도 동등한 성능을 유지하며, 약 3~5배가량 신속한 학습 속도를 가지면서도 기존의 복잡한 전이학습 모델에 비해 정확도, 손실, 혼동행렬 지표에서 동등한 성능을 출력함으로써, 화재 감시 장비의 효율성을 높이는 데 기여할 수 있음을 확인하였다.

키워드 : 레이어 중요도 평가, 전이 학습 모델, 합성곱 신경망 최적화, 실시간 화재 감지, 기여도

1. 서 론

1.1 개요

화재는 인명과 재산에 막대한 피해를 초래하는 주요 재난 중 하나로, 이를 감지하고 조기에 대처하는 것은 중요한 과제이다. 점진적으로 고층화, 대형화가 이루어진 현대 건축물 내에서 발생하는 화재는 초기 대응에 실패할 경우, 큰 재난을 초래할 수 있다. 방재 관점에서 이를 보완하기 위한 설비 기술 중 IT에 관련된 기술이 화재 감시 장비의 성능을 향상할 수 있는 적절한 해결책으로 부상하고 있고, 이러한 필요성에 따라, 화재를 감지할 목적으로 다양한 프로세스가 개발되어 사용되고 있으며, 최근에는 인공지능(AI) 기술을 활용한 화재 인식 모델이 주목받고 있다[1].

1.2 연구 동향 분석

기존 합성곱 신경망(Convolutional neural network) 기반

※ 이 논문은 2024년 ACK 2024의 우수논문으로 "IF2bNet: 화재 감지를 위한 설명 가능 AI 기반 최적화된 딥러닝 아키텍처"의 제목으로 발표된 논문을 확장한 것임.
※ 본 연구는 한성대학교 학술연구비 지원과제임.
† 비 회 원 : 세명대학교 소방방재학과 학사과정
†† 종신회원 : 한성대학교 컴퓨터공학부 부교수
Manuscript Received : July 15, 2024
Accepted : August 5, 2024
* Corresponding Author : Mi-Hwa Song(mhsong@hansung.kr)

의 화재 인식 모델은 주로 전이 학습(Transfer Learning)된 모델을 인계받아 개발되는데[2], 전이 학습은 대규모 데이터 세트로 사전 학습된 모델을 새로운 과제에 맞추어 재학습시키는 방법으로[3], 데이터의 수집이 어려운 화재 데이터 등의 제한된 데이터 환경에서도 우수한 성능을 발휘할 수 있었다. 그러나 모델은 이전 학습 데이터에 포함된 다양한 객체 인식을 기반으로 작동하기 때문에, 화재 감지와 직접적인 관련이 적은 부가적인 프로세스가 내장될 가능성을 시사하며, 부가적인 프로세스는 모델의 복잡성을 증가시키고, 불필요한 연산을 초래하여 실시간 화재 감지의 효율성을 저해할 가능성이 존재한다.

1.3 연구 목적

현재 인공지능 분야에서 모델의 최적화는 매우 중요한 연구 주제 중 하나로, 특히 실시간 데이터 처리와 높은 신뢰성을 요구하는 환경에서 그 중요성이 더욱 부각되고 있다.[4] 본 연구에서는 화재 인식 모델의 효율성을 극대화하기 위해, 기존 전이 학습 기반의 화재 인식 모델의 복잡성을 줄이고 성능을 향상시킬 수 있게 하는 최적화된 딥러닝 아키텍처를 제안한다. 본 연구의 목적은 Layer Importance Evaluation 기법을 활용하여, 화재 감지에 기여도가 높은 프로세스를 식별하고 이를 최적화된 형태로 재구성함으로써, 모델의 경량화와 동시에 성능을 향상시키는 것이다. 이를 위해 다양한 인공지능 모델에서 활용되는 다양한 방법론들을 검토하였다.

2. 이론적 배경

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN)은 이미지 인식 및 분류에 사용되는 인공 신경망의 한 유형으로[5], 이미지에서 특징을 추출하고 이를 기반으로 분류 작업을 수행하는 데 탁월한 성능을 보인다. CNN의 구조는 일반적으로 여러 개의 층(layer)으로 구성되며, 주요 구성 요소로는 합성곱 층(convolutional layer), 풀링 층(pooling layer), 그리고 완전 연결 층(fully connected layer)이 있다.

합성곱 층은 입력 이미지에서 국소적인 특징을 추출하는 역할을 한다. 이를 위해 필터(filter) 또는 커널(kernel)을 사용하여 입력 이미지에 대해 합성곱 연산을 수행한다.

2.2 Layer Importance Evaluation

Layer Importance Evaluation은 딥러닝 모델의 다양한 층(layer)들이 모델의 최종 성능에 미치는 영향을 분석하는 과정이다[6]. 이 과정은 모델의 각 층이 어떻게 기여하는지를 이해하고, 이를 바탕으로 모델을 최적화하거나 해석 가능성을 높이는 데에 중요한 역할을 한다. Layer Importance Evaluation의 다양한 방법론 중 본 연구에서는 레이어의 가중치와 활성화 값을 분석 및 이를 기준으로 레이어를 경량화하는 과정을 중점적으로 진행한다.

1) Weight Analysis

Weight Analysis은 딥러닝 모델의 각 층에 할당된 가중치를 평가하여, 해당 층이 모델의 성능에 얼마나 기여하는지를 분석하는 방법이다[7]. 이 접근법의 주요 가정은 가중치의 크기와 중요도 간의 연관성으로, 특정 층의 가중치가 크다면, 그 층이 모델의 예측에 더 큰 영향을 미친다고 가정한다. 구체적인 공식은 Equation(1)과 같다.

$$Importance(l) = \sum_{i=1}^{m}\sum_{j=1}^{n} \left| W_{ij}^{l} \right| \tag{1}$$

여기서 W_{ij}^{l}는 가중치 매트릭스 W^{l}의 요소이고, m과 n은 각각 가중치 매트릭스의 행(row)과 열(column)의 크기이다. 딥러닝 모델의 한 층 l에서, 해당 층의 가중치 매트릭스를 W^{l}로 정의할 수 있다. W^{l}의 크기는 층의 입력과 출력 차원에 따라 결정된다. 가중치 분석에서는 W^{l}의 절대 값을 합산하여 해당 층의 중요도를 평가할 수 있다.

2) 모델 학습 및 가중치 추출

추출된 매트릭스의 전체 가중치를 Equation(2)에 따라 정의한다. CNN 모델의 경우 합성곱 층과 완전 연결 층의 가중치를 의미한다.

$$W = w_{i,j} \tag{2}$$

여기서 w는 i번째 레이어의 입력과 j번째 레이어의 출력 사이의 전체 가중치를 나타낸다.

3) 기여도 계산 및 정렬

기여도는 가중치의 크기와 활성화 값의 곱을 기준으로 평가하는데, 이는 높은 가중치 값을 갖더라도 해당 뉴런이 활성화되지 않으면, 중요성이 낮다고 간주될 수 있기 때문이다. 구체적인 공식은 Equation(3)과 같다.

$$Importance(w_{i,j}) = \left| w_{i,j} \times a_{i,j} \right| \tag{3}$$

여기서 a는 i번째 레이어의 입력과 j번째 레이어의 출력 사이의 활성화 값의 합을 나타낸다.

4) Layer Removal

산출한 레이어의 기여도의 크기에 따라, 모델의 특정 층을 제거하거나 비활성화한 후, 전체 모델의 성능 변화를 관찰함으로써 해당 층의 중요도를 평가한다. 이 방법은 특정 층이 모델의 성능에 얼마나 기여하는지를 직접적으로 측정할 수 있다.

5) 최적화

추출된 레이어를 바탕으로 모델을 재구성하고, 세부적으

로 불필요한 가중치나 층을 제거하여 모델을 최적화한다. 이를 통해 모델의 경량화가 가능하며, 실시간 성능이 향상될 수 있다. 본 연구에서는 화재 감지에 최적화된 CNN 아키텍처를 개발하기 위해 Layer Importance Evaluation 기법을 활용하여, 효율적이고 신뢰할 수 있는 화재 인식 모델을 구축하고자 한다.

3. 연구방법론

3.1 개발 환경 구축

딥러닝 모델 학습을 위한 개발 환경을 Table 1과 같이 구축하였다.

최신 버전이 아닌 이전 버전의 개발 환경을 채택한 이유는 TensorFlow 2.0 버전이 안정적으로 전통 CNN 모델과의 호환성을 보장하기 때문이다[8]. 최신 버전의 TensorFlow는 종종 새로운 기능과 함께 주요 변경 사항을 포함하며, 이는 기존 코드와의 호환성 문제를 초래할 수 있다. 특히, 화재와 같은 재난 관련 데이터를 다루는 장비에선 안정성과 신뢰성이 중요한 만큼, 기존에 검증된 버전을 사용하는 것이 더 유리하다고 판단되었다.

3.2 데이터 수집 및 전처리

화재 인식 모델을 훈련하기 위해 Kaggle에서 제공하는 각 클래스당 800개, 총 1,200개의 다양한 화재 및 비화재 이미지 데이터 세트[9]를 Table 2와 같이 마련하였다.

1) GAN을 사용한 데이터 생성

건축물 내부에서 발생한 실제 화재 데이터는 그 특성상 데이터의 확보가 어려워, 이를 보완하기 위해 Generative Adversarial Networks(GAN)[10]를 사용하여 Fig. 1과 같은 과정을 통해 추가 생성하였다.

Table 1. Configuring the Development Environment

Category	Specification
Operating system	Windows 10 64-bit
Language	Python 3.7
Deep Learning Framework	TensorFlow GPU 2.0
CUDA Toolkit	Version 10.1
NVIDIA GPU 드라이버	Version 418.X
Tool	Visual Studio 2022
cuDNN	Version 7.6.5

Table 2. Combustion Product Data List

Class	Fire	Smoke	Non_fire
raw data	800	800	800
generated data	400	400	400

Fig. 1. Generated Combustion Product Data

GAN은 두 개의 신경망, 즉 생성자와 판별자로 구성되며, 생성자는 현실적인 데이터를 생성하고 판별자는 생성된 데이터가 실제 데이터인지 여부를 판별하는 역할을 한다. 이러한 상호 경쟁 과정에서 GAN은 점점 더 실제와 유사한 데이터를 생성하게 된다. 구체적인 프로세스는 다음과 같다.

a) 기본 데이터 셋 준비: Kaggle에서 수집된 화재와 비화재 이미지를 이용하여 GAN의 훈련 데이터로 사용하였다.

b) GAN 모델 훈련: GAN 모델은 생성자와 판별자로 구성되며, 생성자는 입력 노이즈로부터 이미지를 생성하고, 판별자는 입력된 이미지가 실제 이미지인지 생성된 이미지인지를 구분한다. 이 과정에서 생성자는 점점 더 실제와 유사한 이미지를 생성할 수 있도록 훈련된다.

c) 데이터 생성: 훈련된 GAN 모델을 사용하여 각 클래스당 400개의 화재 및 비화재 이미지를 추가 생성한다.

2) 생성된 데이터의 품질 평가

생성된 데이터의 품질과 신뢰성을 평가하기 위해, Inception Score[11] 기법을 활용하여 GAN이 생성한 이미지의 클래스 확률 분포를 계산하였다. 구체적인 과정은 다음과 같다.

a) 화재이미지가 학습된 InceptionV3 모델 활용: 사전 학습된 InceptionV3 모델에 생성된 이미지를 입력하여 각 이미지의 클래스 확률 분포를 계산하였다. 이는 생성된 이미지가 실제 화재 이미지를 얼마나 잘 반영하는지를 평가하기 위한 방법으로, 생성된 이미지가 특정 클래스에 얼마나 잘 속하는지를 확인하였다.

b) 품질 평가 기준: 생성된 데이터 중, 실제 화재 이미지와 유사한 클래스 확률 분포를 가지는 이미지들만을 선별하였다. 이 과정에서 비정상적으로 분류된 이미지는 학습 데이터셋 에서 제외하여, 데이터의 품질과 신뢰성을 확보하였다.

c) 데이터 편향 문제 해결: 생성된 데이터셋이 실제 데이터와 유사한 분포를 가지도록 조정함으로써, 데이터 편향 문제를 최소화하였다. 이를 통해, GAN을 사용한 데이터 생성 과정이 실제 화재 감지 시스템에 신뢰할 수 있는 데이터를 제공할 수 있음을 확인하였다.

이와 같은 평가 과정을 통해 생성된 데이터가 실제 화재 감지 시스템에서 활용 가능할 만큼 충분한 품질을 가지고 있음을 입증하였다. 본 연구는 이러한 과정을 통해 기존 화재 데이터의 부족을 보완하고, 모델의 일반화를 확보하고자 하였다.

3.3 모델 학습

최적화를 진행할 적절한 성능의 화재 감지 모델을 구현하기 위해, Xception, VGG19, ResNet50, EfficientNetB5 네 가지 전이 학습 모델을 사용하여, 화재 데이터를 학습시켰다. Xception 계열 모델은 Inception 아키텍처를 확장한 것으로[12], 깊이 분리 합성곱을 활용하여 연산 효율성을 높인 모델이다. VGG 계열 모델은 깊은 합성곱 신경망을 사용하여 이미지 분류 문제에서 높은 성능을 보이는 모델로[13], 단순하지만 매우 효율적인 구조를 갖추고 있다. ResNet 계열 모델은 잔차 연결(residual connection)을 도입하여 매우 깊은 신경망에서도 기울기 소실 문제를 극복한 모델이다[14]. 이는 특히 복잡한 패턴 인식에 유리하다. EfficientNetB5 모델은 신경망의 너비, 깊이, 해상도를 효율적으로 조정하는 컴파운드 스케일링 방법을 사용하여 높은 성능과 연산 효율성을 동시에 달성한 모델이다[15]. 네 가지 모델은 복잡한 패턴 인식 능력이 뛰어나 화재 데이터의 비정형적인 특성을 효과적으로 학습할 수 있다.

3.4 Layer Importance Evaluation

Layer Importance Evaluation 기법을 기반으로, 모델 내부의 레이어들의 규칙을 추출하기 위한 일련의 프로세스를 기획하였다. 각 레이어의 가중치와 활성화 값의 곱인 전체 기여도를 계산하고, 가장 기여도가 높은 상위 6개의 레이어를 선별하였는데, 이는 모델의 성능과 복잡성 간의 균형을 최적화하기 위함이다. 다양한 실험을 통해 Xception, VGG19, ResNet50, EfficientNetB5 모델 각각의 레이어별 기여도를 분석한 결과, 상위 5~8개의 레이어가 전체 Total Weight-Activation 분포의 80% 이상을 차지하는 것을 확인하였고, 이 결과를 바탕으로, 가장 효율적인 성능을 유지하면서도 모델의 복잡성을 최소화할 수 있는 최적의 레이어 수가 6개라고 판단되었다. 구체적인 프로세스는 다음과 같다.

1) 모델의 가중치/활성화 값의 추출 및 통계 계산

가중치 통계는 모델의 각 층(layer)에서 가중치의 평균(mean), 표준 편차(standard deviation), 최소값(min), 최대값(max)을 산출하였는데, 이는 각 층의 가중치와 활성화 값의 분포를 이해하고, 특정 층이 입력 데이터 및 모델 성능에 미치는 영향을 평가하기 위함이다.

2) 활성화 값 추출 및 통계 계산

활성화 값(Activation Value)은 딥러닝 모델에서 각 층의 뉴런이 입력을 받아 계산한 출력 값을 의미한다. 이 값은 뉴런이 입력을 처리한 후 활성화 함수(activation function)를 적용하여 계산된다. 활성화 값 통계는 이미지 데이터를 모델에 입력하여 각 층에서의 활성화 값의 평균, 표준 편차, 최솟값, 최댓값을 Equations (4)~(7)에 따라 산출하였는데, 이를 통해 각 층의 활성화 값 분포를 이해하고, 특정 층이 입력 데이터에 어떻게 반응하는지를 평가하기 위함이다.

$$\mu_a = \frac{1}{M}\sum_{i=1}^{M} a_i \tag{4}$$

$$\sigma_a = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(a_i - \mu_a)^2} \tag{5}$$

$$a_{\min} = \min(a_1, a_2, \ldots, a_M) \tag{6}$$

$$a_{\max} = \max(a_1, a_2, \ldots, a_M) \tag{7}$$

여기서 a_i는 i번째 활성화 값, μ_a는 활성화 값의 평균, σ_a는 활성화 값의 표준 편차, $a_{\min}$는 활성화 값의 최솟값, $a_{\max}$는 활성화 값의 최댓값이다.

3) 기여도 점수 계산

가중치와 활성화 값의 통계로부터 산출한 기여도 값의 곱 중 가장 높은 값을 가진 상위 6개의 레이어를 선별하였다. 전체 기여도 점수는 각 레이어의 평균, 표준 편차, 최솟값, 최댓값의 절댓값을 Equation (8), (9)에 따라 합산하여 산출하였다.

$$S_w = |\mu_w| + |\sigma_w| + |w_{\min}| + |w_{\max}| \tag{8}$$

$$S_a = |\mu_a| + |\sigma_a| + |a_{\min}| + |a_{\max}| \tag{9}$$

여기서 S_w는 가중치의 기여도 점수, S_a는 활성화 값의 기여도 점수이다.

4) 가중치와 활성화 값의 곱 계산 및 정렬

가중치와 활성화 값의 중요도 점수를 Equation (10)에 따라 곱한 후, 그 값을 기준으로 상위 6개의 레이어를 선별하였다.

$$P_i = S_{wi} \times S_{ai} \tag{10}$$

여기서 P_i는 i번째 레이어의 전체 기여도의 곱, S_{wi}, S_{ai}는 i번째 레이어의 가중치 및 활성화 값의 기여도 함수이다.

5) 선별된 레이어를 사용하여 새로운 모델 구축

다음과 같은 규칙을 통해 추출한 레이어의 기여도 값을 참조하여 모델을 재구성한다. 전체 기여도에 크기에 따라 레이어를 내림차순으로 정렬한 뒤, 이중 상위 6개의 레이어를 선정하여 그 레이어를 바탕으로 아키텍처를 다시 재구축하였다.

종합적으로, 이러한 프로세스를 통해 모델의 효율성을 높이고, 화재 감지 성능을 개선하고자 한다. 구체적인 Flow Chart는 Fig. 2와 같다.

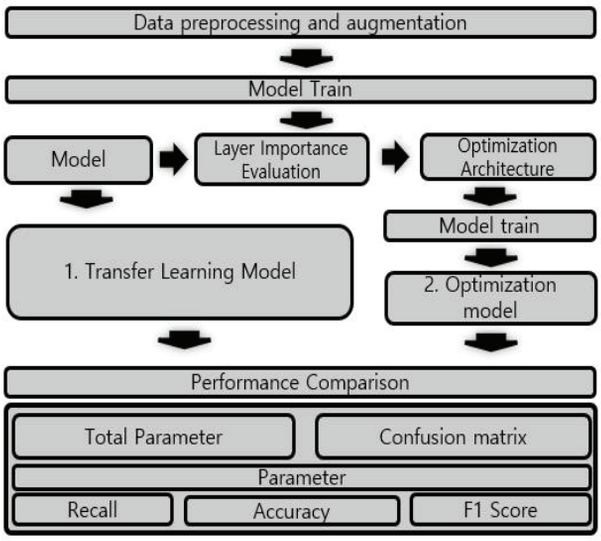


Fig. 2. Research Progress Flow Chart

4. 실험 결과

모델의 성능 평가는 정확도, 재현율, F1 스코어, 혼동 행렬, 학습 속도 등의 다양한 지표를 포함하여 수행하는데, 이는 모델의 성능 및 실제 활용 가능성을 평가하는 중요한 요소이다. 이러한 포괄적인 평가를 통해 최적화된 모델이 기존의 전이학습 모델 대비 어떤 관점에서 성능 향상과 효율성을 제공하는지 명확하게 분석하였다.

4.1 레이어 통계

레이어별 전체 기여도를 기반으로 추출한 세 가지 Xception, VGG19, ResNet50 모델의 통계는 Table 3~6과 같이 산출되었다.

Xception 모델의 경우, 가장 높은 중요도 점수를 기록한 레이어는 block1_conv2 레이어로, 가중치-활성화 곱의 값이 470.719로 나타났다. 그 외 주요 레이어로는 block7_sepconv1_bn, block4_sepconv1 등이 포함되었다.

VGG19 모델의 경우, 가장 높은 중요도 점수를 기록한 레이어는 block2_conv1 레이어로, 가중치-활성화 곱의 값이 170.708으로 나타났다. 그 외 주요 레이어로는 block1_conv2, block2_conv2 등이 포함되었다.

ResNet 모델의 경우, 가장 높은 중요도 점수를 기록한 레이어는 block1_conv1 레이어로, 가중치-활성화 곱의 값이 250.216으로 나타났다. 그 외 주요 레이어로는 block1_conv2, block2_conv2 등이 포함되었다.

EfficientNetB5 모델의 경우, 가장 높은 중요도 점수를 기록한 레이어는 Conv2D_1 레이어로, 가중치-활성화 곱의 값이 520.1263으로 나타났다. 그 외 주요 레이어로는 MBConv6, MBConv5 등이 포함되었다.

Table 3. Xception Layer's Contribution Statistics Top Rank

layer position	Layer Name	Weight-Activation Product
2	block1_conv2	470.719
32	block7_sepconv1_bn	143.442
14	block4_sepconv1	135.156
8	conv2d_57	128.539
26	block6_sepconv1_bn	126.229
34	block7_sepconv2_bn	118.471

Table 4. VGG19 Layer's Contribution Statistics Top Rank

layer position	Layer Name	Weight-Activation Product
2	block2_conv1	170.7083
1	block1_conv1	30.63228
14	block5_conv3	30.26214
3	block2_conv2	31.19704
8	block4_conv1	22.84488
5	block3_conv2	21.7744

Table 5. Resnet Layer's Contribution Statistics Top Rank

layer position	Layer Name	Weight-Activation Product
1	block1_conv1	250.2165
3	block1_conv2	36.4274
2	block2_conv2	21.1145
4	block2_conv1	18.6552
7	block3_conv1	22.84488
5	block4_conv4	13.2844

Table 6. EfficientNetB5 Layer's Contribution Statistics Top Rank

layer position	Layer Name	Weight-Activation Product
1	Conv2D_1	520.1263
61	MBConv6	148.3273
84	MBConv5	51.1001
91	Final Dense Layer	45.6610
42	MBConv4	43.4353
20	GAP Conv2D	16.2822

4.2 모델별 성능 지표

최적화된 모델의 성능을 기존 전이학습 모델과 비교 평가한 결과, 전체 파라미터(Total Parameter), 정확도(Accuracy), 손실(Loss), 재현율(Recall), F1 스코어(F1 Score), 혼동 행렬(Confusion Matrix) 및 학습 속도(Training Time) 등의 다양한 성능 지표에서 향상된 성능을 보였다.

1) Total Parameter

최적화된 모델들은 Table 7과 같이 기존 모델에 비해 전체 파라미터 수가 크게 줄어들었다.

Table 7. Resnet Layer's Contribution Statistics Top Rank

Model	Total Parameters (Original)	Total Parameters (Optimized)	(%)
Xception	22,910,480	4,126,560	81.96
VGG19	143,667,240	11,681,792	91.87
ResNet50	25,636,712	3,424,384	86.64
EfficientNetB	32,312,218	4,332,429	86.59

최적화된 Xception 모델은 81.96%의 파라미터 감소를 이루었고, VGG19 모델은 91.87%의 파라미터 감소, ResNet50 모델은 86.64%의 파라미터 감소, EfficientNetB5 모델은 86.59%의 파라미터 감소를 이루었다. 이러한 파라미터 감소는 모델의 메모리 사용량을 줄이고, 학습 및 추론 속도를 향상시켜 실제 환경에서의 적용 가능성을 높이는 중요한 성과를 나타낸다.

2) Inference Time

최적화된 모델들의 실행 시간(Inference Time)은 Table 8과 같이 최적화되었다.

3) Accuracy

Xception 모델의 학습 및 검증 정확도는 Fig. 3과 같이 초기에는 급격히 상승한 후 안정화되는 모습을 보인다. 특히, 학습 초기 단계에서 학습 정확도가 빠르게 상승하며, 검증 정확도도 이를 따라 안정적으로 증가한다. 40 에포크 이후에는 두 곡선 모두 거의 0.98에 도달하며, 이는 모델이 잘 수렴했음을 의미한다.

VGG 모델의 경우, Fig. 4와 같이 학습 및 검증 정확도가 에포크가 진행됨에 따라 꾸준히 증가하는 경향을 보인다. 두 곡선 간의 차이가 크지 않으며, 이는 모델이 과적합(over-fitting) 없이 학습되었음을 시사한다. 50 에포크가 끝날 무렵, 학습 정확도는 약 0.9에 도달하고, 검증 정확도는 약 0.89에 도달한다. 이는 VGG 모델이 안정적이며, 검증 데이터에 대해서도 좋은 일반화 성능을 보여줌을 나타낸다.

ResNet 모델은 Fig. 5와 같이 학습 초기 단계에서 급격한 증가 추세를 보이며, 이후 비교적 천천히 수렴하는 패턴을 나타낸다. 학습 및 검증 정확도 곡선 모두 20 에포크 이후 안정화되며, 50 에포크에서 약 0.92에 도달한다. 두 곡선의 유사한 형태는 모델이 잘 일반화되었음을 보여준다.

EfficientNetB5 모델의 학습 및 검증 정확도는 Fig. 6와 같이 초기에는 빠르게 상승한 후 점차 완만해지는 로그 함수형 성장 곡선을 따른다. 학습 초기 단계에서는 학습 정확도가 급격히 상승하며, 검증 정확도 또한 이를 따라 안정적으로 증가한다. 두 곡선은 약 50 epoch에 도달했을 때 각각 0.9와 0.88에 도달하며, 이는 모델이 잘 수렴하고 있음을 의미한다.

Table 8. Resnet Layer's Contribution Statistics Top Rank

Model	Inference Time (Original)	Inference Time (Optimized)	(%)
Xception	10.3452	7.2345	30.07
VGG19	15.6789	10.4567	33.32
ResNet50	12.5678	9.8765	21.42
EfficientNetB	13.7543	8.4612	33.32

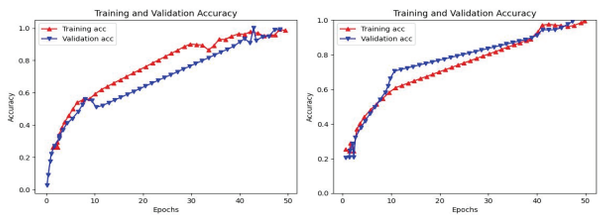


Fig. 3. Xception(L) and Optimization Model(R) Accuracy Progress

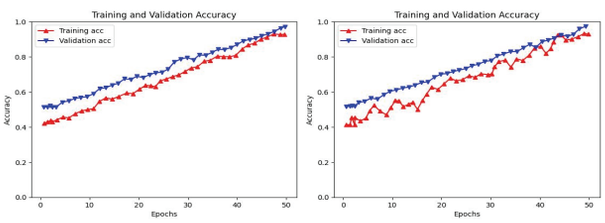


Fig. 4. VGG(L) and Optimization Model(R) Accuracy Progress

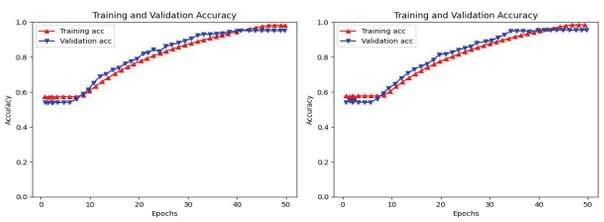


Fig. 5. Resnet(L) and Optimization Model(R) Accuracy Progress

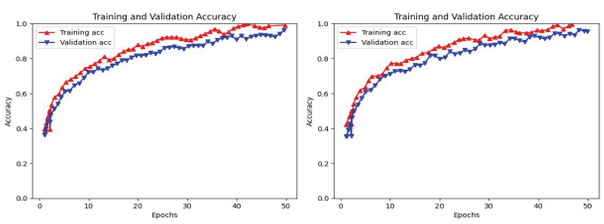


Fig. 6. EfficientNetB5(L) and Optimization Model(R) Accuracy Progress

4) Training Time and Loss over Epochs

최적화된 세 가지 모델은 Figs. 7~10과 같이 기존 모델과 거의 동일한 손실 감소 추세를 유지하면서도 학습 속도에서 큰 향상을 보였다.

최적화된 Xception 모델은 기존 모델과 비교하여 평균 학습 시간이 약 4.12배 빨랐다. 구체적으로, 기존 Xception 모델의 평균 학습 시간은 49.41초였으나, 최적화된 모델은 11.01초로 단축되었다.

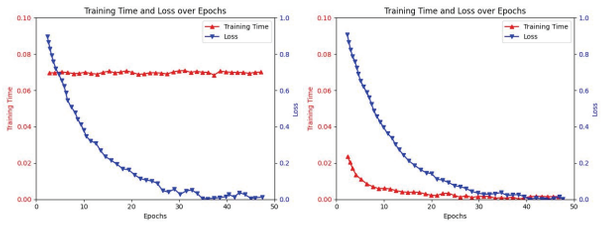


Fig. 7. Xception model Training Time and Loss over Epochs

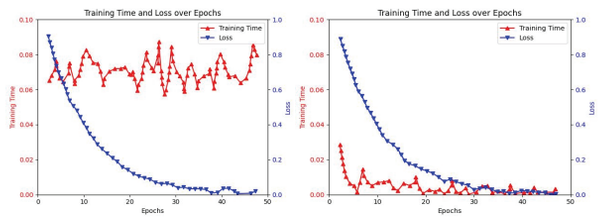


Fig. 8. VGG19 model Training Time and Loss over Epochs

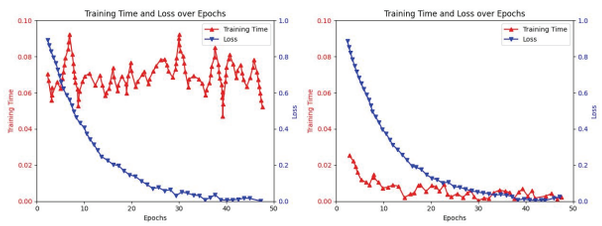


Fig. 9. ResNet50 model Training Time and Loss over Epochs

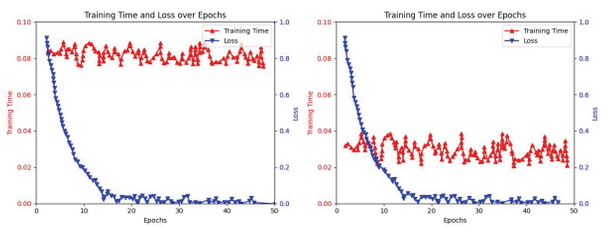


Fig. 10. EfficientNetB5 model Training Time and Loss over Epochs

최적화된 VGG19 모델은 기존 모델 대비 약 4.97배 빠른 학습 속도를 보였다. 기존 VGG19 모델의 평균 학습 시간은 59.58초였으나, 최적화된 모델은 13.01초로 단축되었다.

최적화된 ResNet 모델은 기존 모델 대비 학습 속도가 약 3.38배 빨랐다. 기존 ResNet 모델의 평균 학습 시간은 40.57초였으나, 최적화된 모델은 14.08초로 단축되었다.

최적화된 EfficientNetB5 모델은 기존 모델 대비 학습 속도가 약 2.88배 빨랐다. 기존 모델의 평균 학습 시간은 60.24초였으나, 최적화된 모델은 20.92초로 단축되었다.

5) Confusion Matrix

Xception, VGG19, ResNet50 모델과 최적화된 모델의 혼동 행렬을 Figs. 11~14와 같이 비교 분석하였다.

좌측 Xception 모델은 모든 클래스에서 100% 정확한 분류를 보여주었다. 우측 최적화된 모델의 경우 'Fire' 클래스에서

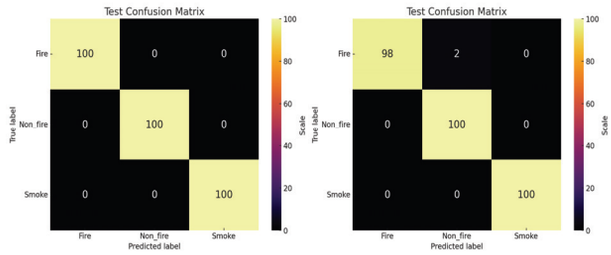


Fig. 11. Xception(L) and Optimization Model(R) Confusion Matrix

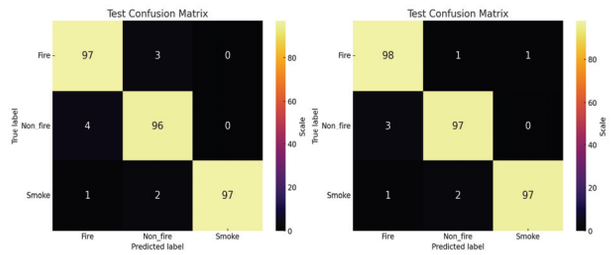


Fig. 12. VGG19(L) and Optimization Model(R) Confusion Matrix

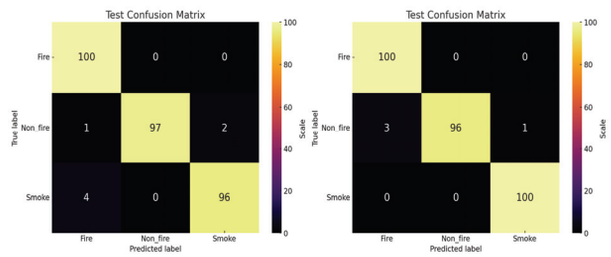


Fig. 13 ResNet50(L) and Optimization Model(R) Confusion Matrix

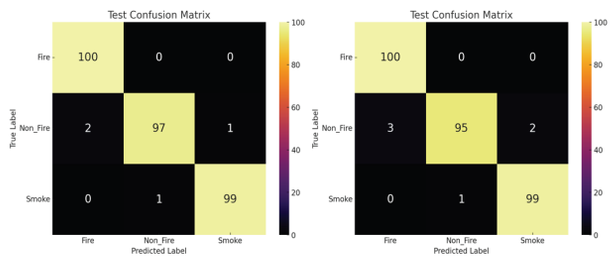


Fig. 14. EfficientNetB5(L) and Optimization Model(R) Confusion Matrix

98개를 정확히 예측하고 2개의 오차가 발생하였다. 'Non-fire'와 'Smoke' 클래스에서도 100%의 정확도를 보였다. 이는 최적화된 모델이 여전히 높은 성능을 유지하고 있음을 나타낸다.

좌측 VGG19 모델은 'Fire' 클래스에서 97%의 정확도를 보였고, 3개의 오차가 발생하였다. 'Non-fire' 클래스는 96%의 정확도를 보였으며, 'Smoke' 클래스는 97%의 정확도를 보였다. 우측 최적화된 모델의 경우 'Fire' 클래스에서 98%의 정확도를 보였고, 'Non-fire' 클래스는 97%, 'Smoke' 클래스는 97%의 정확도를 보였다.

좌측 ResNet50 모델은 'Fire' 클래스에서 100%의 정확도를 보였으며, 'Non-fire' 클래스는 96%, 'Smoke' 클래스는 100%의 정확도를 보였다. 우측 최적화된 모델은 'Fire' 클래스에서

Table 9. Model Evaluation Indicator Analysis Statistics

Name	Class	Accuracy	Precision	Recall	F1_score
Xception	Fire	0.98	0.97	0.98	0.98
	Non_fire		0.99	0.99	0.99
	Smoke		0.96	0.97	0.97
Optimized Xception	Fire	0.98	0.98	0.98	0.98
	Non_fire		0.99	0.99	0.99
	Smoke		0.97	0.97	0.97
VGG19	Fire	0.91	0.89	0.90	0.90
	Non_fire		0.90	0.91	0.90
	Smoke		0.92	0.91	0.90
Optimized VGG19	Fire	0.90	0.91	0.91	0.91
	Non_fire		0.91	0.90	0.91
	Smoke		0.92	0.91	0.91
ResNet50	Fire	0.92	0.91	0.92	0.92
	Non_fire		0.93	0.92	0.92
	Smoke		0.92	0.93	0.92
Optimized ResNet50	Fire	0.95	0.94	0.94	0.94
	Non_fire		0.94	0.94	0.94
	Smoke		0.95	0.95	0.95
EfficientNet B5	Fire	0.98	0.97	0.98	0.98
	Non_fire		0.99	0.96	0.97
	Smoke		0.99	0.99	0.99
Optimized EfficientNetB5	Fire	0.97	0.99	0.99	0.99
	Non_fire		0.96	0.97	0.96
	Smoke		0.97	0.96	0.96

97%의 정확도를 보였고, ‘Non-fire’ 클래스는 97%, 'Smoke' 클래스는 96%의 정확도를 보였다. 이는 최적화된 모델이 성능을 보존하였음을 나타낸다.

좌측 EfficientNetB5 모델은 ‘Fire’ 클래스에서 100%의 정확도를 보였으며, ‘Non-fire’ 클래스는 97%, ‘Smoke’ 클래스는 99%의 정확도를 보였다. 우측 최적화된 모델은 ‘Fire’ 클래스에서 100%의 정확도를 보였고, 'Non-fire' 클래스는 95%, ‘Smoke’ 클래스는 99%의 정확도를 보였다. 이는 최적화된 모델이 성능을 보존하였음을 나타낸다. 결과적으로, 최적화된 모델들은 원본 모델과 비교하여 모든 클래스에서 성능 보존을 보였다. 이러한 결과는 최적화된 모델이 화재 감지 시스템에서 신뢰할 수 있는 성능을 제공함을 입증한다.

6) Model evaluation indicator analysis

모델 평가 지표 분석에서는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 스코어(F1-score) 등을 통해 기존 전이 학습 모델 및 최적화 모델을 Table 9과 같이 정리하였다.

모델이 경량화 됨에도 불구하고 정확도, 정밀도, 재현율, F1-Score 모두 최적화된 버전에서 동등한 성능을 유지하였다. 모든 클래스에서 높은 정확도와 F1 스코어를 기록하며, 연소생성물 데이터 인식에 필요한 프로세스가 누락되지 않고, 모델이 온전히 유지되었음을 의미한다.

5. 결 론

본 연구에서는 다양한 전이학습 모델(Xception, VGG19, ResNet, EfficientNetB5)을 활용하여 화재 감지 시스템의 최적화를 수행하였다. 각 모델 내 레이어의 가중치와 활성화 값의 중요도를 평가하고, 상위 6개의 레이어를 선별하여 새로운 최적화 모델을 구축하였다. 최적화된 모델은 기존 전이 학습 모델 대비 학습 속도에서 큰 향상을 보였으며, 성능 지표에서도 거의 동등하거나 우수한 결과를 나타냈다.

제안된 방법론은 기존 화재 감지 시스템의 복잡성과 비효율성을 해결하는 데 기여할 수 있었다. 실시간 데이터 처리 환경에서 높은 신뢰성과 성능을 제공함으로써 화재 감지 시스템의 효율성을 향상시키고, 실제 적용 가능성을 높이는 데 중요한 역할을 할 것이다.

다만, 제한된 데이터 세트를 사용하여 실험을 진행하였기 때문에 다양한 환경에서의 일반화 능력을 보장하지 못한다는 한계가 있다. 향후 연구에서는 다양한 데이터 셋을 활용한 추가 실험을 통해 제안된 모델의 범용성을 검토하고, 다른 재난 감지 시스템에도 적용 가능 여부를 검토하고자 한다.

References

[1] W. Zeng, Y. Zhao, W. Xiong, and Z. Zhu, “Early fire detection using convolutional neural networks for CCTV surveillance cameras,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp.629-638, 2022.

[2] S. Han, H. Lee, K. Kim, and J. Park, “Fire detection in video surveillance using convolutional neural networks,” *International Journal of Advanced Robotic Systems*, Vol. 18, No.3, pp.1-12, 2021.

[3] J. L. Hennessy and D. A. Patterson, “TransTailor: Pruning the pre-trained model for improved transfer learning,” in *Computer Architecture: A Quantitative Approach*, 4th ed., San Francisco, CA: Morgan Kaufmann Pub., ch.2, pp.66-153, 2007.

[4] RapidCanvas, “Optimizing Machine Learning Models for Real-Time Data Processing,” RapidCanvas Blog, Jun. 7, 2024. [Online]. Available: https://www.rapidcanvas.ai/blogs/optimizing-machine-learning-models-for-real-time-data-processing. [Accessed: Aug. 15, 2024].

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, Vol.1, No.4, pp.541-551, 1989.

[6] Y. Shi, P. Wang, and J. Yang, “A comprehensive review of artificial intelligence applications in power systems,” *Neural Networks*, Vol.172, pp.372-393, 2024.

[7] Y. LeCun, J. S., Denker, and S. A. Solla, "Optimal brain damage," *Advances in Neural Information Processing Systems*, pp.598-605, 1990.

[8] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp.265-283, 2016.

[9] Phylake1337, "Fire Dataset," Kaggle, 2020. Available: https://www.kaggle.com/datasets/phylake1337/fire-dataset.

[10] I. Goodfellow et al., "Generative adversarial networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp.2672-2680, 2014.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Advances in Neural Information Processing Systems*, pp.2234-2242, 2016.

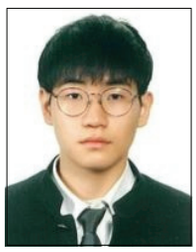
[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1251-1258, 2017.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770-778, 2016.

[15] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp.6105-6114, 2019.

[16] W. Jin and M. Song, "FCPNet: An optimized deep learning architecture for fire detection based on explainable AI," in *Proceeding of the Annual Symposium of KIPS 2024*, Vol.31, No.1, pp.719-720, 2024.



진 원

https://orcid.org/0009-0007-1009-9496
e-mail : wlsdnjs370@gmail.com
2024년 세명대학교 소방방재학과 학사과정
관심분야 : Model optimization, Fire detection, Convolutional Neural Network

송 미 화

https://orcid.org/0000-0001-7047-8032
e-mail : mhsong@hansung.kr
2013년 ~ 2023년 세명대학교 스마트IT학부 부교수
2024년 ~ 현 재 한성대학교 컴퓨터공학부 부교수
관심분야 : Explainable AI, HCI, Big Data Analysis