

Original Paper

Research on Fire Detection in Laboratories Based on CNN and Transfer Learning

Ke Liu¹

¹ Assets and Equipment Department, Xi'an Medical University, Xi'an, Shaanxi, China

Received: April 25, 2023

Accepted: May19, 2023

Online Published: June 12, 2023

doi:10.22158/fet.v6n2p142

URL: <http://dx.doi.org/10.22158/fet.v6n2p142>

Abstract

To solve the problem that model may not be convergent owing to datasets with small sample sizes results from the secrecy of labs, a fire detection method for lab fire detection based on CNN (Convolutional Neural Network) and transfer learning is proposed. Experiments of transfer learning were conducted with configuration of super parameters and a small lab fire dataset, whose pre-trained models are GoogLeNet, ResNet and MobileNet trained with ImageNet dataset. We analyze and assess the loss function curve and the performances of the above models. Result shows that MoblieNet model is chosen to be the optimum model for fire detection, providing a reference to lab fire detection with small sample size datasets. Different from artificial fire surveillance, the proposed method based on CNN and transfer learning realize real-time, autonomy and efficient fire detection.

Keywords

CNN, transfer learning, indoor fire detection, lab safety

1. Introduction

Laboratory fires are characterized by an initial small range that gradually expands before suddenly spreading, often resulting in incalculable losses once detected. The presence of flammable and explosive chemicals in laboratories can further accelerate the fire's spread, posing a significant safety threat to laboratory personnel and expensive experimental equipment (Wilbraham, 1979; Gray, 2003; Dupuy & Maréchal, 2011). While many laboratories currently use camera-equipped monitoring methods to detect potential danger, human intervention is still necessary. This cannot guarantee the prompt identification and response to laboratory fires.

Artificial intelligence and deep learning technology has made remarkable progress in recent years, leading to the rapid development and widespread use of image classification and recognition technology. Facial recognition and augmented virtual reality are now deployed on mainstream portable

devices, while license plate recognition and assisted driving are applied in the commercial field and promote the construction of smart cities (Raji & Fried, 2021; Du, Ibrahim, Shehata, & Badawy, 2012; Silva, Khan, & Han, 2018). In comparison to traditional image recognition technologies, deep learning-based image classification and recognition technology effectively achieve fire detection and emergency response by continuously training and learning from sample data, dynamically optimizing models and collaborating with laboratory alarm systems. Among various deep learning algorithms, convolutional neural network (CNN) excels in reducing model complexity, making it widely used in image classification and recognition technology (Alzubaidi et al., 2021; Kaur et al., 2022; Chauhan, Ghanshala, & Joshi, 2018). However, massive amounts of data are required for deep learning leading to issues related to personal privacy or business secrets. To address this problem, transfer learning is often employed (Weiss, Khoshgoftaar, & Wang, 2016; Torrey & Shavlik, 2010; Zhuang et al., 2020). In this paper, we use transfer learning and CNN to achieve real-time and reliable fire detection in laboratories.

2. Correlation Theory

2.1 CNN Architecture

Before the emergence of CNN, deep learning faced obstacles such as long training time or low accuracy in image processing. It became necessary to reduce the dimensionality of data while preserving image features leading to the development of CNN. As a subclass of deep learning, CNN is widely used in image classification and recognition. Traditional neural networks used fully connected layers in their hidden layer with neurons linked between layers. CNN also has input, hidden, and output layers, while the hidden layer incorporates convolutional and pooling layers to preserve image features while reducing data dimensions. Compared to traditional neural networks, CNN can process images directly. The structure of a CNN is shown in Figure 1.

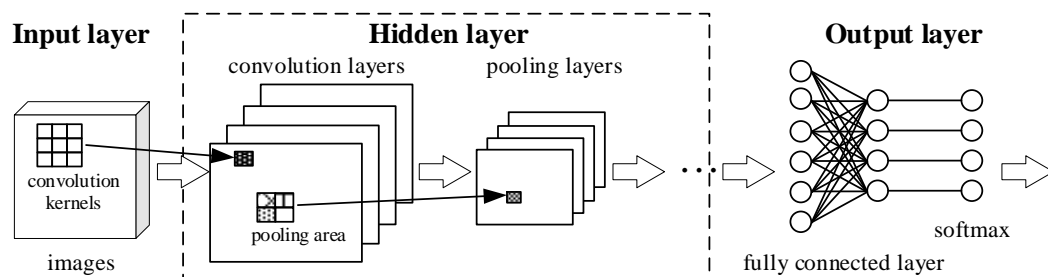


Figure 1. CNN Architecture

The input layer of CNN can process grayscale or RGB color images. Convolutional layers emulate the human visual cortex and can extract complex image features more effectively. Humans observe an image through pupils with initiating preliminary processing such as edge detection in the cerebral

cortex. Then, the observed image is identified. Figure 2 shows an example of the working principle of convolutional layers.

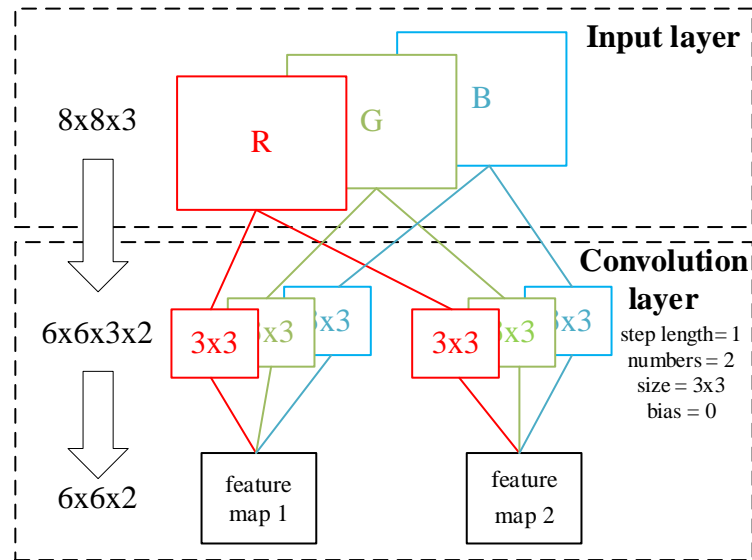


Figure 2. Principle of Convolution Layers

The convolutional layer utilizes convolution kernels that slide sequentially over image pixels to perform spatial filtering on the image. A convolution kernel represents a spatial filter of size $F_w \times F_h \times F_c$ with weights denoted by $\mathbf{W}_{F_w \times F_h \times F_c}$, where F_w and F_h are non-negative, odd numbers frequently set equal to each other. For a specific channel F_c , the central element of the spatial filter represents the current pixel weight of the original image and other elements represent the weight of adjacent pixels. The output is the sum of values from different image features that pass through the spatial filter from various channels. In the pixel coordinate system, the intensity of the image pixel point $\text{pixel}_{u,v}$ is represented by $f[u, v]$. If both F_w and F_h are odd numbers, the output intensity

$g_{\text{conv}}[u, v]$, which operated by a square spatial filter of size $F_w \times F_h \times F_c$, is shown in Equation (1),

where b is the offset.

$$g_{\text{conv}}[u, v] = \sum_{F_c} \sum_{i=-(F_w-1)/2}^{(F_w-1)/2} \sum_{j=-(F_h-1)/2}^{(F_h-1)/2} f[u+i, v+j] \mathbf{M}[i+(F_w+1)/2, j+(F_h+1)/2] + b \quad (1)$$

If $u+i$ or $v+j$ exceeds the boundary of the image, various filling methods can replace the ineffective pixel intensities. Additionally, the step size used by the spatial filter determines the pixel spacing for each sliding of the convolution kernel. The number of output channels with an image or feature map passing through a convolutional layer equals to the number of convolution kernels in that

layer. Assuming that a convolutional layer is composed of convolution kernels of size $F \times F$, a step size S , and a fill of P , the output size $N \times N$ is given in Equation (2).

$$N = (W - F + 2P) / S + 1 \quad (2)$$

The pooling layer significantly decreases the dimensionality of image data and prevents overfitting. Unlike standard sampling typically applied to one-dimensional sequences, the pooling layer samples high-dimensional image inputs or feature maps. The most commonly methods include maximum pooling and average pooling. For maximum pooling, if the pooling region is $N \times N$ and $f[u, v]$

represents the input with one channel, the maximum pooling result on $[u, v]$ is given in Equation (3) assuming N is odd.

$$g_{\text{poll}}[u, v] = \max\{f[u+i, v+j]\}, -(N-1)/2 \leq i, j \leq (N-1)/2 \quad (3)$$

The output layer consists of a fully connected layer, an activation function, and others. If M categories need to be recognized and classified, the number of fully connected layer neurons directly is M with each neuron being connected to a neuron in the previous layer. The input-output relationship of neurons is expressed as $\mathbf{Y} = \mathbf{W}\mathbf{X} + \mathbf{B}$, where the weight \mathbf{W} and bias \mathbf{B} create a linear component. Besides, the activation function provides the nonlinear transformation. The common activation functions include the Sigmoid and ReLU functions. The output layer often employs the Softmax function, allowing for more intuitive judgment of image categories. By combining convolutional and pooling layers with different dimensions and numbers, various CNN architectures have been developed to apply to different scenes with varying objectives.

2.2 CNN Model

2.2.1 GoogLeNet

In 2014, Google collaborated with several universities to propose GoogLeNet, a CNN architecture that won the first position in ImageNet's image classification challenge (Szegedy et al., 2015). In comparison to the previous year's champion ZFNet network and 2014's VGG network, GoogLeNet achieved significantly higher accuracy and notably lower error probability.

GoogLeNet is a 22-level CNN that implements multiple 1×1 convolution kernels and global average pooling. The 1×1 convolution kernels decreases the network's complexity while deepening its depth. Figure 3 illustrates the principle of reducing the computation in 1×1 convolution kernels. Moreover, GoogLeNet employs global average pooling to convert 7×7 feature maps to 1×1 size. This technique reduces the number of parameters to 0 in the layer and enhances the accuracy. Additionally, GoogLeNet proposed the Inception Module that uses 1×1 , 3×3 , and 5×5 convolutional layers alongside 3×3 maximum pooling layers. After stacking, the module can adapt to image features of different scales.

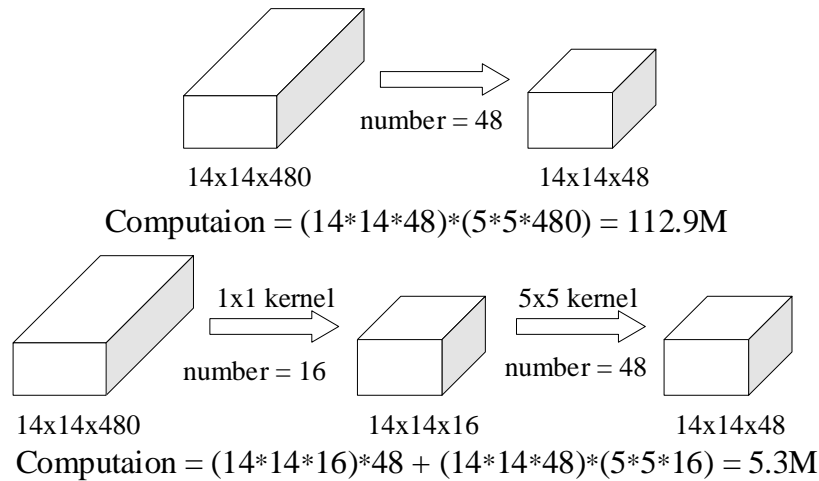


Figure 3. Principle of Reducing the Computation in 1x1 Convolution Kernels

GoogLeNet trained on the ImageNet dataset can classify and recognize over 1000 objects, including animals, numbers, stationery. Further trained on the Place365 dataset, GoogLeNet can classify and recognize 365 locations, such as fields, trails, banquet halls, and many others.

2.2.2 ResNet

Since the inception of the first CNN, AlexNet, at the ImageNet contest in 2012, the error rate has decreased progressively every year by increasing the number of network layers in deep learning. However, research has revealed that the performance of the network might diminish with endless layer incrementation leading to an increased error rate. This phenomenon is known as gradient vanishing or exploding. While adding layers increases the network's parameter number and enables deeper feature identification, neural networks use gradient descent for model performance optimization via backpropagation partial differential multiplication. Therefore, as the layer number increases, the gradient will exponentially increase resulting in a gradient explosion when the product is greater than 1. Conversely, the gradient can disappear.

Researchers from Microsoft in 2015 proposed the residual network (ResNet) (He et al., 2016). ResNet implements skip connection technology linking directly to the active layer, which effectively addresses the issue of vanishing or exploding gradients. When vanishing or exploding gradients in a middle layer, the weights of that layer are set to zero ensuring consistent input and output while preventing a degradation in performance of the entire network. Unlike traditional convolutional networks, where it is difficult for a large number of layers within a dashed line to remain unchanged, ResNet makes it easier to keep the output value unchanged. To illustrate this difference, Figure 4 shows the comparison.

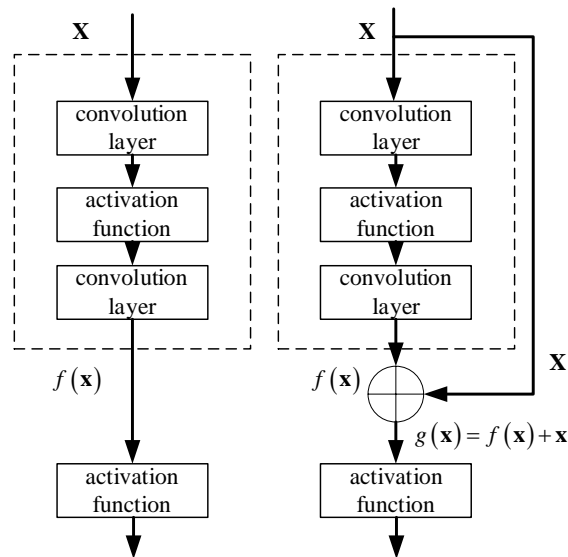


Figure 4. Comparison of Conventional CNN and ResNet

2.2.3 MobileNet

With the continuous development of CNN, the performance improvement has been achieved by increasing the number of layers. However, researchers have realized that although complex neural networks can improve performance but increase model complexity. As a result, the multiplication and addition operations of massive model parameters requires strong computing power in reality. To address this challenge, some researchers have shifted their research focus to small-scale CNN models which can maintain good model performance while avoiding the need for strong computing power. Google's MobileNet is a compact and efficient CNN model that uses depth-wise separable convolution to construct lightweight neural networks (Howard et al., 2017; Sandler et al., 2018). In traditional convolutions, the kernel is applied uniformly across all the input image channels. However, depth-wise separable convolution performs convolution separately on each channel as shown in Figure 5. To avoid compromising the effectiveness of the model, pointwise convolution is introduced. Pointwise convolution is a 1×1 convolution kernel for dimensionality enhancement. Unlike traditional convolution, depth-wise separable convolution achieves faster model convergence with fewer parameters.

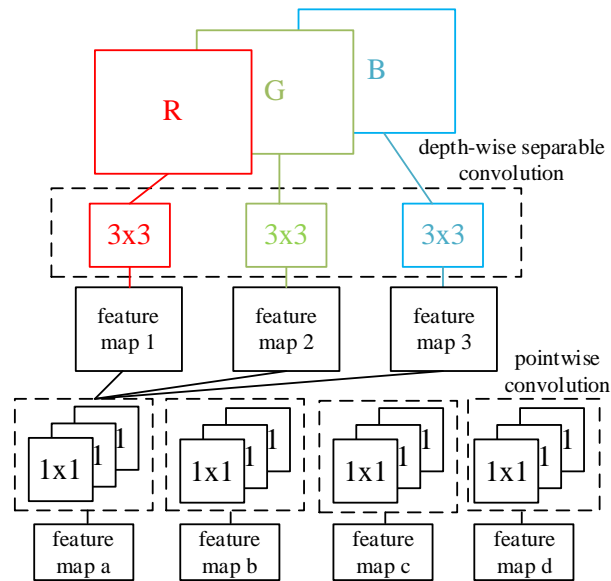


Figure 5. Principle of Depth-wise Separable Convolution

2.3 Transfer Learning

In practical applications of neural networks, it is common to encounter problems that lack relevant, large datasets for sufficient training neural network models. Moreover, the training process requires large amounts of computational resources and time. To address these issues, researchers have introduced transfer learning. Transfer learning uses pre-trained neural networks to train models for new tasks like human use experiences to solve new tasks in life. Figure 6 shows the comparison between traditional machine learning and transfer learning. The source domain refers to the sample and label set of the original network, while the target domain is ones in the new network. The process of transfer learning typically consists of three steps:

- 1). Selecting a pre-trained model correlated with the target domain features. For instance, if a neural network model can classify and recognize sunglasses, retraining on corresponding datasets can help the model acquire knowledge about backpacks to classify and recognize them.
- 2). Fine-tuning the pre-trained model by adjusting the model layer structure for specific applications. In some cases, certain layers are frozen to keep the parameters unchanged during training. For example, the fully connected layer of the original model composed of 1000 neurons can be replaced with a fully connected layer composed of 10 neurons to recognize 10 new objects.
- 3). Train and validate the new model by monitoring the training process, adjusting hyperparameters and evaluating model performance after convergence.

In fact, training an initial CNN model is rare. Instead, transfer learning is commonly used in different but relevant fields to solve new problems. It also extracts sample features in a relatively short time on small training datasets while enabling efficient classification and recognition.

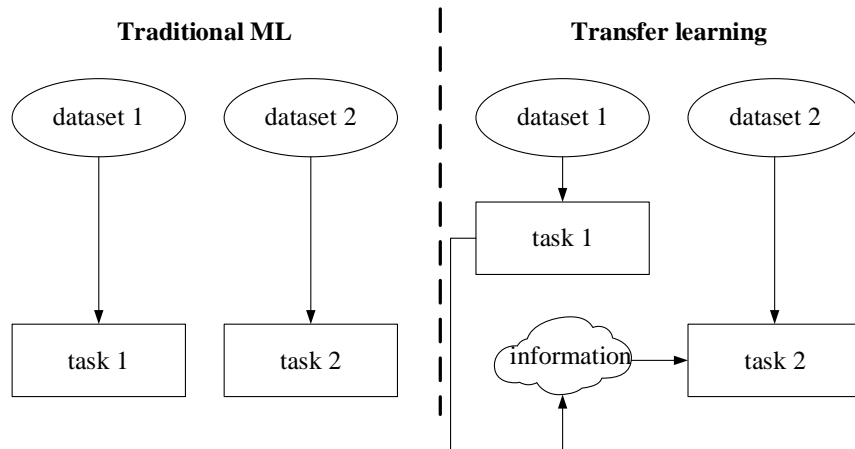


Figure 6. Comparison of Conventional Machine Learning and Transfer Learning

3. Experimental Preparation

3.1 Experimental Environment

This paper uses laboratory fire and non-fire images as samples to conduct transfer learning on three specific CNNs: GoogLeNet, ResNet18, and MobileNet-V2, which are pre-trained models. The goal is to identify whether there is a fire in the laboratory and to evaluate the performance of the three models. The experimental steps are shown in Figure 7.

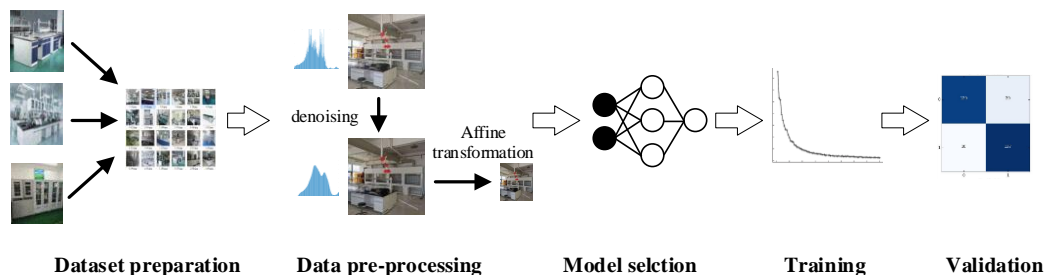


Figure 7. Procedure of Experiments

The hardware processor used in our paper is the AMD R5-3600. The software configuration operating system is Windows 10 64-bit, and the software is Matlab 2021b. The learning models in this research are based on the Deep Learning Toolbox, the Deep Learning Toolbox for GoogLeNet Network (Version 21.2.1), ResNet-18 Network (Version 21.2.0) and MobileNet-V2 Network (Version 21.2.0) frameworks. All these models were trained preliminarily on ImageNet. The dataset in our paper consists of 1510 actual laboratory fire and non-fire images sourced with noise from the internet. Wherein, 755 images are related to fires, while the others non-fires.

3.2 Data Preprocessing

In real-world scenarios, image acquisition devices often capture images with random noise due to non-ideal environmental factors such as lighting and temperature. This noise can impact image pixels while affecting image quality and even making it difficult to distinguish the image. Therefore, denoising the images is a crucial first step. Additionally, since the input layer size in CNN models often differs from the original image size, it is necessary to perform an affine transformation to adjust its size, which is necessary before proceeding with subsequent model training.

3.3 Gaussian Filtering Denoising

Gaussian filter denoising refers to using Gaussian kernels for image spatial filtering, which applies a weighted average to the noise of surrounding pixels and reduces the impact. Equation (4) shows the probability density function $G(x, y)$ of a two-dimensional Gaussian distribution with a mean of zero, where σ^2 represents the variance.

$$G(x, y) = e^{-\frac{(x^2 + y^2)}{2\sigma^2}} / 2\pi\sigma^2 \quad (4)$$

The weight $W[i, j]$ of Gaussian filtering is similar to the expression of $G(x, y)$ but needs to be normalized. The Gaussian kernel weight $W[i, j]$ of size $(2n+1) \times (2n+1)$ is shown in Equation (5).

$$W[i, j] = \frac{e^{-\frac{(i-n)^2 + (j-n)^2}{2\sigma^2}} / 2\pi\sigma^2}{\sum_{i=0}^{2n} \sum_{j=0}^{2n} e^{-\frac{(i-n)^2 + (j-n)^2}{2\sigma^2}} / 2\pi\sigma^2} \quad (5)$$

It can be observed that the weight value is solely determined by the standard deviation σ . Since the weight decreases as one moves farther from the center, a $(2n+1) \times (2n+1)$ size filter is typically used where $\lceil \bullet \rceil$ denotes the round function and usually $n = \lceil 2\sigma \rceil$. In contrast to mean filtering which assigns equal weight to each pixel, Gaussian filtering results in a smoother and softer filtered image, which better preserves edge features.

The Peak Signal-to-Noise Ratio (PSNR) is a widely utilized method to assess image quality. In this paper, a 5x5 Gaussian filter with $\sigma = 1$ was employed. Figure 8 shows raw and processed images along with their respective PSNR values. From both subjective and objective perspectives of evaluations, the image quality experienced significant enhancements following Gaussian denoising.

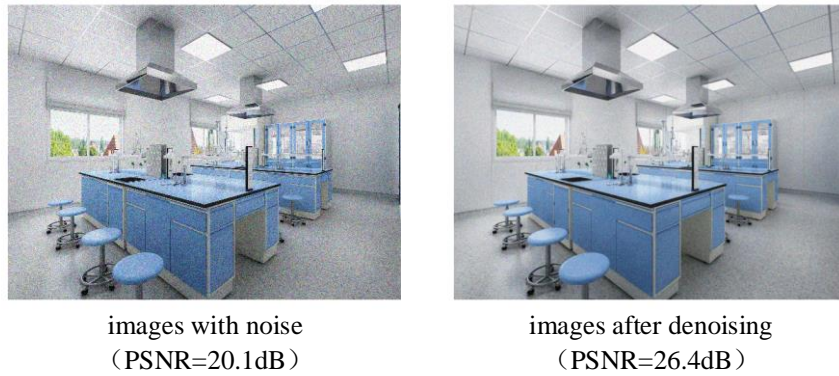


Figure 8. Comparison of Images w/ and w/o Noise

3.3.1 Bilinear Interpolation

During Affine transformations, such as zooming in or out, the coordinates of pixels in the transformed image are typically non-integer, as they are obtained by backward mapping the pixel coordinates from the original image. For instance, to scale down a 5x5 image with a pixel coordinate $f[u, v], 0 \leq u \leq 4, 0 \leq v \leq 4$ to 3x3. The corresponding pixel coordinate of $[1, 1]$ in the original image results in a non-integer coordinate. Rounding fractional values, also known as nearest-neighbor interpolation, is the most common method to tackle the aforementioned challenges. However, it may substantially alter the original image by distorting vital information.

Bilinear interpolation is a type of image affine transformation that utilizes adjacent four pixel points on the two-dimensional image to determine the value of transformed pixels. Bilinear interpolation attempts to preserve image characteristics as much as possible. Assuming the corresponding non-integer coordinates in the original image is $[u, v]$, the value after bilinear interpolation is shown in Equation (6).

$$\hat{f}[i, j] = (1-b)((1-a)f[i, j] + af[i+1, j]) + b((1-a)f[i, j+1] + af[i+1, j+1]) \quad (6)$$

Where $[i, j]$, $[i+1, j]$, $[i, j+1]$, and $[i+1, j+1]$ are four adjacent pixel points of $[u, v]$, $a = u - i$ and $b = v - j$. As the input layer sizes of GoogLeNet, ResNet, and MobileNet are 224x224x3 images, this paper utilizes bilinear interpolation to transform all laboratory fire and non-fire images to adapt to the above dimensions.

3.4 Model Training

To enable transfer learning for laboratory fire detection, fine-tuning GoogLeNet, ResNet, and MobileNet was executed. The modification is replacing the full connection layer and Softmax function located at the neural network's end, with a new full connection layer which consists of two neurons and the corresponding Softmax function. The rest of the neural network structure remained the same.

The dataset was divided into training and validation sets, where the images in the training set were used for model training, and those in the validation set were utilized for evaluating model performance during and after training. In this paper, the training set was randomly chosen and comprise of 80% of the labeled (fire or non-fire) images from the dataset, while the remaining 20% formed the validation set.

The cross entropy loss function was used for all models. Continuous training was applied to adjust the model parameter values until the function value approached zero. For binary classification, the equation of the cross entropy loss function is expressed in Equation (7), where N denotes the number of samples, y_i is the sample label (0 denotes no fire, while 1 denotes fire), p_i represents the predicted probability value, and $\log(\bullet)$ is a logarithmic function.

$$L = -\frac{1}{N} \sum_{i=1}^N (y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)) \quad (7)$$

Next, the hyperparameters involved in model training were set. The learning rate for the model was set to 0.0001. While keeping the number of training rounds (Epochs) constant, the maximum Epochs was set to 10. The model optimizer was used to locate optimal parameter solutions. In this paper, the mini-batch stochastic gradient descent with momentum (MSGD) method was applied with a momentum proportion of 0.9 and a small batch size of 128. The training set was shuffled before each batch division round. To prevent overfitting, the L2-norm regularization technique was utilized by adding a norm penalty term.

3.5 Evaluating Indicator

The rate and performance of model convergence are essential factors to consider. The rate of convergence may be evaluated by examining changes in the training and validation losses. During the training process, observing the changes in these two losses can facilitate early detection of underfitting or overfitting problems. Adjusting the hyperparameters of the model and beginning a new training process can help improve experimental efficiency.

Model performance is indicated by metrics such as accuracy, precision, recall, and the F1 score. To analyze these metrics, it is common to establish a confusion matrix, which employs rows to represent the actual sample classifications and columns to show the predicted classifications. The matrix's values represent the number of samples that conform to both the row and column traits. For binary classification, the confusion matrix can be presented using the four elements: TP , TN , FN and FP , as shown in Table 1.

Table 1. Confusion Matrix

	1 Predicted	0 Predicted
1 Actual	TP	FN
0 Actual	FP	TN

Accuracy is the ratio of the number of correct predictions to the total number of samples and equals $(TP+TN)/(TP+FP+FN+TN)$. The precision rate denotes $TP/(TP+FP)$. Additionally, the recall rate is defined as $TP/(TP+FN)$, which represents the proportion of correctly predicted positive samples in the total number of actual positive samples. Generally, a higher value for these indicators indicates better model performance. Moreover, the evaluation metric introduced to assess the overall model performance is the F1 score, which combines both accuracy and recall indicators, and is determined by $2 * (Precision + Recall) / (Precision * Recall)$.

4. Experimental Results and Analysis

Based on the above-mentioned experimental environment, dataset, data preprocessing and model training settings, this paper conducted transfer learning to detect laboratory fires using GoogLeNet, ResNet, and MobileNet models. The models were iteratively optimized to achieve convergence.

The loss curves of training and verification based on the three models are presented in Figure 9 and 10. Both the training and validation loss gradually decline with the continuous increase in model iteration and ultimately reach stability, which indicates that the final convergence results met expectations without overfitting or underfitting. Furthermore, the training and validation loss curves in GoogLeNet and ResNet have a faster decline rate compared to MobileNet. After approximately 20 iterations, the training and validation losses of GoogLeNet approach zero and those of ResNet and MobileNet only converge to zero after approximately 30 iterations. Additionally, the curve changes of MobileNet are more stable than those of GoogLeNet and ResNet without significant fluctuations. Based on these observations, it can be inferred that the GoogLeNet model is suitable as the primary model for laboratory fire detection. MobileNet may be used as the suboptimal model due to its slower convergence rate.

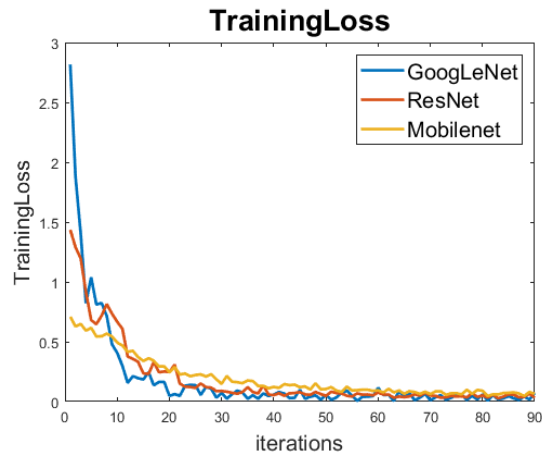


Figure 9. Training Loss

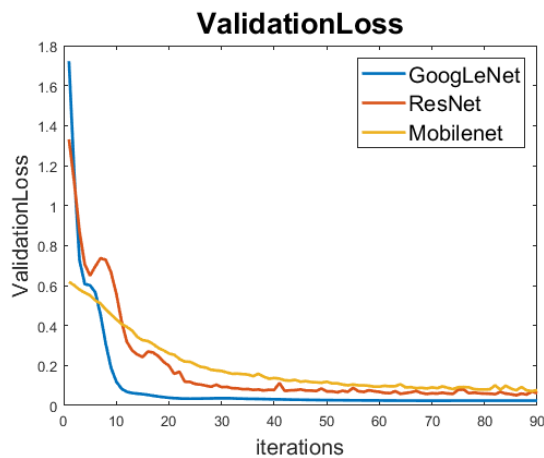


Figure 10. Validation Loss

In addition to monitoring changes in training and validation losses, the confusion matrix was constructed for the three models to provide evaluations of performance. The 2x2 confusion matrixes is shown in Figure 11, 12, and 13, respectively. The right part in matrixes indicates the proportion of each element after row normalization, which reflects the probabilities of correct and incorrect predictions based on the actual samples. Likewise, the lower part in matrixes displayed represents the proportion of each element after column normalization, which shows the probability of actual correctness and error in the predicted samples.

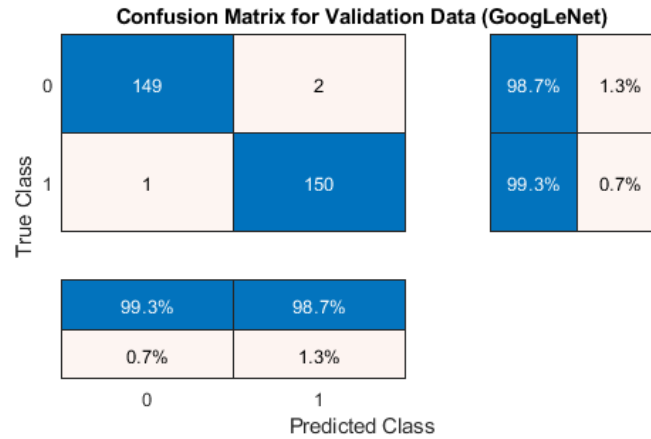


Figure 11. Confusion Matrix of GoogLeNet

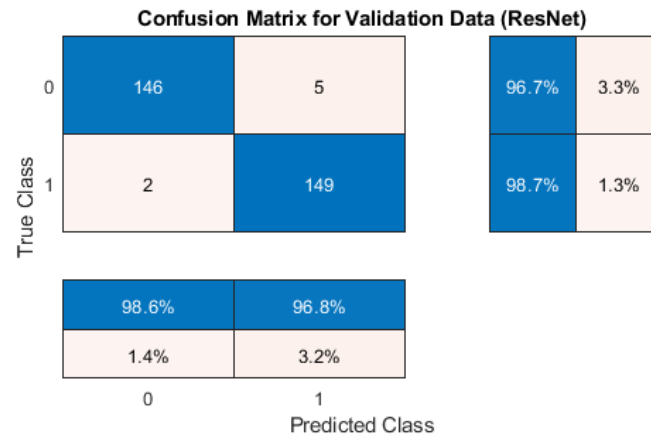


Figure 12. Confusion Matrix of ResNet

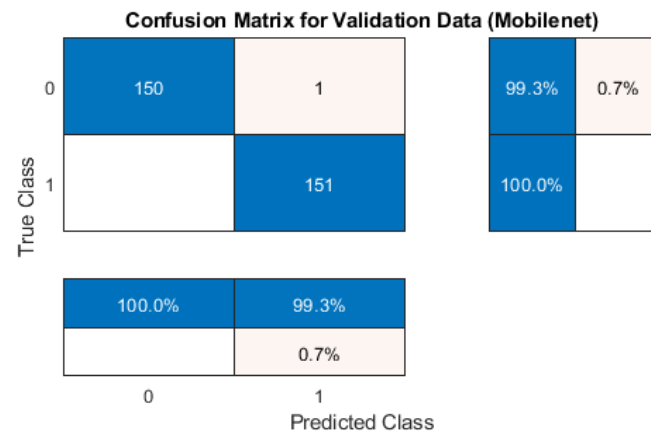


Figure 13. Confusion Matrix of MobileNet

This paper evaluated the accuracy rate, recall rate, and F1 score of three models. Table 2 presents the results showing that MobileNet outperformed GoogLeNet and ResNet with the highest scores in accuracy, precision, recall, and F1. Although there is a slight difference for the three models, the variation is insignificant especially between GoogLeNet and MobileNet. Based on these results, MobileNet can serve as the ideal model and GoogLeNet can be considered a secondary choice for laboratory fire detection in transfer learning.

Table 2. Performances of Learning Models

	accuracy rate	precision rate	recall rate	F1 Score
GoogLeNet	0.9901	0.9934	0.9868	0.9901
ResNet	0.9768	0.9868	0.9675	0.9770
MobileNet	0.9967	1.0000	0.9934	0.9967

To assess the rate of convergence and overall performance of the final models generated for small sample datasets, this paper compared MobileNet and GoogLeNet. While MobileNet exhibited a slightly lower rate of convergence than GoogLeNet, the difference in iteration algebra was within acceptable limits. Furthermore, the loss function of MobileNet was comparatively more stable and the performance results were optimal. Based on these observations, MobileNet among the three models is the preferred option for laboratory fire detection, as it is capable of obtaining optimal model performance with a fast rate of convergence.

5. Conclusion

This paper proposes a laboratory fire detection scheme that combines CNN and transfer learning to address the challenge of training models with limited laboratory fire images, which may lead to extended iteration times or even model failure. To assess the efficacy of the scheme, three pre-trained CNN models: GoogLeNet, ResNet, and MobileNet, were trained after conducting Gaussian denoising and bilinear interpolation on laboratory fire dataset. The results of the experiments demonstrated that all three models converged quickly and achieved high recognition rates, validating the effectiveness of the proposed approach. After analyzing and comparing the rate of convergence and model performance of the different CNN models, it was observed that the MobileNet achieved superior results for laboratory fire detection. Therefore, it can be concluded that the MobileNet-based transfer learning model is better suited for laboratory fire detection with small sample datasets under CNN while providing valuable insights for improving laboratory safety through the use of neural networks.

References

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8, 1-74. <https://doi.org/10.1186/s40537-021-00444-8>
- Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In *2018 first international conference on secure cyber computing and communication (ICSCCC)* (pp. 278-282). IEEE. <https://doi.org/10.1109/ICSCCC.2018.8703316>
- Du, S., Ibrahim, M., Shehata, M., & Badawy, W. (2012). Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Transactions on circuits and systems for video technology*, 23(2), 311-325. <https://doi.org/10.1109/TCSVT.2012.2203741>
- Dupuy, J. L., & Maréchal, J. (2011). Slope effect on laboratory fire spread: Contribution of radiation and convection to fuel bed preheating. *International Journal of Wildland Fire*, 20(2), 289-307. <https://doi.org/10.1071/WF09076>
- Gray, D. (2003). Automatic fire suppression for fume hoods promotes laboratory safety. *Research & Development*, 45(5), 47.
- He, K., Zhang, X., Ren, S., et al. (2016). *Deep residual learning for image recognition*[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). <https://doi.org/10.1109/CVPR.2016.90>
- Howard, A. G., Zhu, M., Chen, B. et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.
- Kaur, G., Sinha, R., Tiwari, P. K., Yadav, S. K., Pandey, P., Raj, R., ... Rakhra, M. (2022). Face mask recognition system using CNN model. *Neuroscience Informatics*, 2(3), 100035. <https://doi.org/10.1016/j.neuri.2021.100035>
- Raji, I. D., & Fried, G. (2021). *About face: A survey of facial recognition evaluation*. arXiv preprint arXiv:2102.00813.
- Sandler, M., Howard, A., Zhu M., et al. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. IEEE. <https://doi.org/10.1109/CVPR.2018.00474>
- Silva, B. N., Khan, M., & Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable cities and society*, 38, 697-713. <https://doi.org/10.1016/j.scs.2018.01.053>
- Szegedy, C., Liu, W., Jia, Y., et al. (2015). *Going deeper with convolutions*[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9). <https://doi.org/10.1109/CVPR.2015.7298594>
- Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264). IGI global. <https://doi.org/10.4018/978-1-60566-766-9.ch011>

- Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1), 1-40. <https://doi.org/10.1186/s40537-016-0043-6>
- Wilbraham, A. C. (1979). Safety in the chemical laboratory: Fire safety and fire control in the chemistry laboratory. *Journal of Chemical Education*, 56, N/A. <https://doi.org/10.1021/ed056pA311>
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76. <https://doi.org/10.1109/JPROC.2020.3004555>