





## Article

# An Improved Weighted Cross-Entropy-Based Convolutional Neural Network for Auxiliary Diagnosis of Pneumonia

Zhenyu Song <sup>1,\*</sup>, Zhanling Shi <sup>1,2,\*</sup>, Xuemei Yan <sup>1</sup>, Bin Zhang <sup>1</sup>, Shuangbao Song <sup>3</sup> and Cheng Tang <sup>4</sup>

<sup>1</sup> College of Information Engineering, Taizhou University, Taizhou 225300, China; yanxuemei@tzu.edu.cn (X.Y.); zhangbin@tzu.edu.cn (B.Z.)

<sup>2</sup> School of Computer Science and Engineering, LinYi University, Linyi 276000, China

<sup>3</sup> School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China; leadingsong@cczu.edu.cn

<sup>4</sup> Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan; tang@ait.kyushu-u.ac.jp

\* Correspondence: songzhenyu@tzu.edu.cn (Z.S.); zhanlingshi430@gmail.com (Z.S.)

† These authors contributed equally to this study.

**Abstract:** Pneumonia has long been a significant concern in global public health. With the advancement of convolutional neural networks (CNNs), new technological methods have emerged to address this challenge. However, the application of CNNs to pneumonia diagnosis still faces several critical issues. First, the datasets used for training models often suffer from insufficient sample sizes and imbalanced class distributions, leading to reduced classification performance. Second, although CNNs can automatically extract features and make decisions from complex image data, their interpretability is relatively poor, limiting their widespread use in clinical diagnosis to some extent. To address these issues, a novel weighted cross-entropy loss function is proposed, which calculates weights via an inverse proportion exponential function to handle data imbalance more efficiently. Additionally, we employ a transfer learning approach that combines pretrained CNN model parameter fine-tuning to improve classification performance. Finally, we introduce the gradient-weighted class activation mapping method to enhance the interpretability of the model's decisions by visualizing the image regions of focus. The experimental results indicate that our proposed approach significantly enhances CNN performance in pneumonia diagnosis tasks. Among the four selected models, the accuracy rates improved to over 90%, and visualized results were provided.

**Keywords:** pneumonia diagnosis; convolutional neural network; cross-entropy



**Citation:** Song, Z.; Shi, Z.; Yan, X.; Zhang, B.; Song, S.; Tang, C. An Improved Weighted Cross-Entropy-Based Convolutional Neural Network for Auxiliary Diagnosis of Pneumonia. *Electronics* **2024**, *13*, 2929. <https://doi.org/10.3390/electronics13152929>

Academic Editor: Maciej Ławryńczuk

Received: 26 June 2024

Revised: 17 July 2024

Accepted: 20 July 2024

Published: 24 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Pneumonia is a common infectious disease that affects the respiratory system and poses a significant threat to humanity. The outbreak of the novel coronavirus (COVID-19) in 2019 has further heightened global awareness of the dangers posed by pneumonia [1]. Traditionally, pneumonia diagnosis has relied primarily on the observation and judgment of medical experts. However, this diagnostic approach is both time-consuming and susceptible to subjective factors, increasing the risk of misdiagnosis. Moreover, this method demands substantial medical human resources, limiting its feasibility for widespread application [2].

In recent years, with the rapid advancement of machine learning and deep learning technologies, these techniques have become crucial in many areas of modern society. From web search and social network content filtering to recommendation systems on e-commerce websites [3,4], their applications in consumer products such as cameras and smartphones have also become increasingly widespread [5]. Additionally, deep learning has been applied in the industrial sector to assist with anomaly detection [6] and fault diagnosis [7,8]. Machine learning systems are used to identify objects in images; transcribe speech to text; match news, posts, or products to user interests; and select relevant search results [9,10].

Notably, computer vision technology has demonstrated significant advantages in image recognition and analysis [11], and convolutional neural networks (CNNs) have emerged as a prominent technology in medical imaging [12]. The success of CNNs in image recognition has provided new insights and methods for medical image analysis, making the use of CNNs for pneumonia image diagnosis a popular research topic [13,14].

This study aims to increase the automation and accuracy of pneumonia diagnosis through the application of CNNs, thereby reducing the reliance on medical human resources and improving diagnostic efficiency. This is of significant practical importance for alleviating the burden on healthcare systems and enhancing pneumonia treatment outcomes. However, despite the substantial potential of CNNs in image recognition, their application in medical image diagnosis still faces challenges related to data imbalance and insufficient model interpretability [15,16]. To address these issues, this study proposes a weighted cross-entropy loss function based on an inverse proportional exponential function (IPEF) to address the imbalanced dataset problem. Additionally, we employ transfer learning (TL) techniques to optimize pretrained models, thereby enhancing their performance in pneumonia diagnosis tasks. To improve model interpretability, we introduce gradient-weighted class activation mapping (Grad-CAM) technology [17], which provides more reliable clinical decision support. The experimental results demonstrate that the methods proposed in this study effectively improve pneumonia diagnosis accuracy and model interpretability. This offers new research perspectives for medical image analysis and provides accurate and reliable support for future clinical applications.

In summary, the main innovations and contributions of this study are as follows: (a) A novel weighted cross-entropy loss function is proposed, which calculates weights via an IPEF to address the issue of imbalanced datasets more effectively. (b) This study adopts a TL approach, incorporating pretrained CNN model parameter fine-tuning to enhance classification performance. (c) By introducing the Grad-CAM method, this study achieves visualization of the regions in the images that the model focuses on, thereby enhancing the interpretability of CNN model decisions.

The remainder of this paper is structured as follows: Section 2 reviews related work pertinent to this study. Section 3 provides a detailed description of the CNN architecture, TL, and weighted cross-entropy loss function. Section 4 elaborates on the dataset, experimental setup, evaluation metrics, and analysis of the experimental results. Finally, Section 5 offers conclusions and directions for future work.

## 2. Related Work

In the early stages of medical image processing, basic techniques such as thresholding, region growing, and edge tracking were primarily used. While these methods remain effective in general cases, their performance is often limited when handling complex medical images [18]. With the increase in computational power and the emergence of big data, machine learning has entered a new phase. In this phase, statistical learning and neural networks have attracted increasing attention, leading to the development of various models that have been successfully applied across different fields [19–24].

Since Vapnik and others proposed the support vector machine (SVM) in 1963, it has become a powerful tool for pattern recognition and classification problems [25]. For example, Osareh et al. developed an automated system based on gene microarray data for robust and reliable cancer diagnosis [26]. Through carefully designed feature extraction methods, this system effectively distinguishes between malignant and benign tumors. Additionally, Yahyaoui et al. proposed a decision support system based on SVM and adaptive SVM algorithms to address chest disease diagnosis [27]. This system can be used to diagnose pneumonia and accurately classify chronic obstructive pulmonary disease, significantly improving diagnostic accuracy. Despite the good performance of SVM-based medical image classification systems in specific applications, they face several challenges. First, training an SVM involves solving a large quadratic programming problem, with computational complexity and memory requirements proportional to the square of the number of training

samples. This makes the training process extremely slow and sometimes infeasible with limited computational resources when dealing with large datasets. Second, the kernel function choice is crucial for SVM performance, but tuning and selecting appropriate kernel parameters for large datasets is a complex and time-consuming process. Additionally, traditional SVM optimization algorithms, such as sequential minimal optimization, may not be efficient for large-scale datasets. Although some optimized variants exist, these often require trade-offs between accuracy and computational efficiency [28].

In 2001, Breiman first introduced the random forest algorithm [29], an ensemble learning method composed of multiple decision trees. The output of a random forest is a combination of the outputs from each individual tree. In medical image classification, random forests are favored by researchers because of their excellent classification performance, ability to handle high-dimensional data, and robustness to noise in the training data. For example, Anthimopoulos et al. proposed a random-forest-based scheme for classifying interstitial lung disease patterns in high-resolution CT images [30]. Their results demonstrated the effectiveness of the method in handling complex medical image data. Additionally, Bhattacharjee et al. introduced a hybrid approach that combines an optimized random forest classifier with a K-means visualization algorithm for lung cancer diagnosis [31], significantly improving classification performance and diagnostic accuracy. Despite the various advantages of the random forest algorithm in medical image classification, there are several drawbacks. First, the interpretability of random forest models is relatively poor. Since the model comprises multiple decision trees, the overall decision-making process is not as straightforward or easily understandable as that of a single decision tree. In the medical field, model interpretability is especially crucial, as doctors and clinical practitioners must clearly understand how the model makes its predictions. Second, random forests have numerous hyperparameters that must be tuned, such as the number and depth of the trees and the minimum number of samples required for a node split. Finding the optimal combination of parameters often requires extensive time for experimentation and cross-validation.

The introduction of CNNs has marked a new era in medical image classification. CNN technology has been widely applied in medical image classification related to lung diseases. In 2017, Esteva et al. utilized CNNs to classify skin cancer [32], developing a deep CNN model to identify various skin lesions, including malignant melanomas and benign nevi. They trained the model on a large dataset of skin images, employed the pretrained GoogLeNet Inception v3 architecture, and trained it on a dataset of over 120,000 images representing more than 2000 different diseases. The study results indicated that the performance of the model in multiclass classification tasks was competitive with that of 21 experienced dermatologists. Rajpurkar et al. developed the CheXNet model [33], a deep-learning-based CNN model designed to analyze chest X-ray images for pneumonia detection. CheXNet is based on the DenseNet121 architecture, which improves feature utilization efficiency and reduces model parameters through its dense connectivity. The model was trained on the ChestX-ray14 dataset, which contains over 100,000 X-ray images labeled with 14 different chest diseases. By learning from a large quantity of labeled data, CheXNet successfully mastered the complex visual patterns necessary to distinguish pneumonia from other chest diseases. The performance of CheXNet in identifying pneumonia even surpassed the average level of radiologists, demonstrating the powerful potential of CNNs in medical image classification and emphasizing the significant value of deep learning technologies in assisting medical diagnosis. Despite the exceptional performance of CNNs in medical image classification, several challenges remain in their application [34,35]. For example, CNNs are highly dependent on large quantities of labeled data. However, acquiring sufficient and balanced data is often challenging, which can lead to suboptimal performance of the trained CNN models. Additionally, CNN models are typically considered “black box” models, with a lack of transparency in their decision-making processes. This opacity can be problematic in medical decisions that require high interpretability.

### 3. Methodology

In this section, we focus on explaining the methods and related theories proposed in this study, including classical CNN models, the proposed weighted cross-entropy loss function, TL, and Grad-CAM visualization technology.

#### 3.1. Convolutional Neural Network

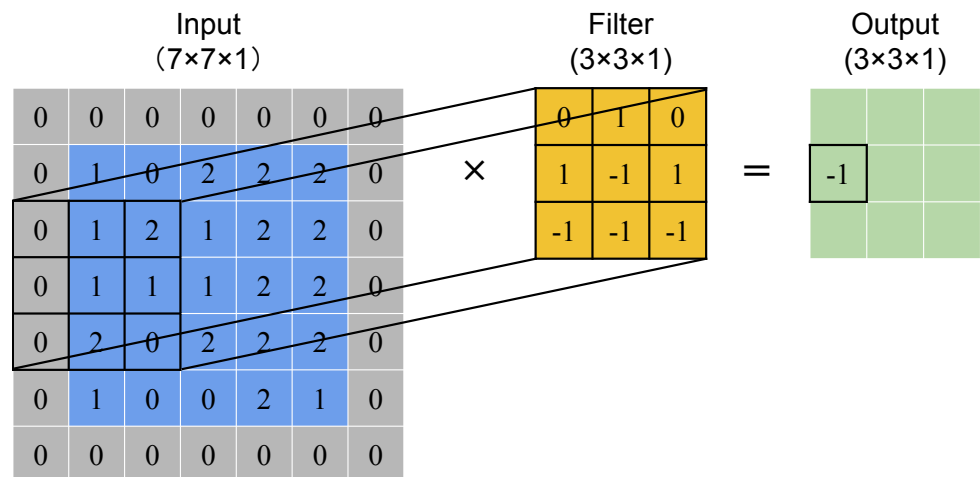
CNNs are a class of feedforward neural networks that include convolutional computations and have a deep architecture. They are among the representative algorithms in the field of deep learning. CNNs have been widely applied in image processing and computer vision, particularly in tasks such as object detection, image classification, and image segmentation. Their primary feature is the ability to automatically and adaptively learn local features of images and to combine and abstract these local features into higher-level global features. The basic structure of a CNN includes an input layer, convolutional layers, activation layers, pooling layers, and fully connected layers.

##### 3.1.1. Input Layer

In a CNN, the input layer is responsible for receiving raw data and passing them on to subsequent layers. The CNN input layer typically represents images or other multidimensional data. The number of nodes in the input layer is determined by the dimensions of the input data. For example, each pixel in an RGB image may have three channels (red, green, blue), so the number of nodes in the input layer is the number of pixels in the image multiplied by the number of channels. Specifically, the input layer passes the raw data to the neural network in the form of matrices or tensors. In image recognition tasks, the input layer receives image pixel values. Data preprocessing is usually required before the input layer. For image data, these preprocessing steps include scaling, normalization, or other image enhancement techniques. The input layer acts as a bridge in CNNs, connecting the raw data to the subsequent feature extraction process and ensuring that the data can be efficiently processed and learned.

##### 3.1.2. Convolution Layer

The convolutional layer is one of the core CNN components; it is used to process images and multidimensional data. The convolutional layer extracts features from the input data through the convolution operation, enabling the network to capture spatial structures and patterns effectively. This is achieved by applying filters that slide over the input data, performing local perception. Each convolutional kernel learns different features, such as edges, textures, or higher-level features. The convolution kernel slides over the input data, performing convolution operations at each position to generate an output. Multiple feature maps can be formed via multiple convolutional kernels, each corresponding to the features learned by one convolutional kernel. The stride of the convolution kernel determines the step size at which the kernel moves over the input data. Additionally, padding is used to maintain consistent dimensions between the input and output and to prevent information loss at the edges. A larger stride can reduce the size of the output feature map and improve computational efficiency but may result in some information loss. For example, assume that the input is a  $5 \times 5$  matrix, with padding set to 1 and stride set to 2. The process of a single convolutional kernel performing convolution on this input is illustrated in Figure 1.



**Figure 1.** Convolutional layer.

### 3.1.3. Activation Layer

In a CNN, the activation layer is a crucial component. Its primary role is to introduce nonlinearity, allowing the neural network to fit and solve complex problems better. The activation layer typically follows the convolutional layer and applies a nonlinear function, known as the activation function, to the output of the convolutional layer. This function maps the input from the neurons to the output. Common activation functions include the sigmoid function, rectified linear unit (ReLU) function, and hyperbolic tangent (tanh) function.

- The sigmoid activation function is a commonly used function in neural networks. The mathematical expression of the sigmoid function is shown in Equation (1). The most notable characteristic of the sigmoid function is that its output is bounded and lies between 0 and 1. This makes it particularly important in the output layer when dealing with binary classification problems. Additionally, the sigmoid function is continuously differentiable, which is a crucial property for optimization algorithms such as gradient descent.

$$F(x) = \frac{1}{1 + e^{-x}}. \tag{1}$$

- The ReLU activation function is a simple yet effective nonlinear function; its mathematical expression is shown in Equation (2). The ReLU function remains linear for positive values and outputs zero for negative values. This nonlinear transformation helps introduce nonlinear characteristics into the network, enabling it to learn more complex functional relationships.

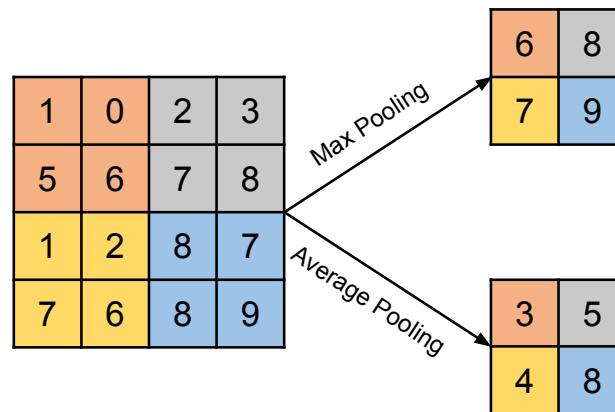
$$ReLU(x) = \max(0, x). \tag{2}$$

- The mathematical expression of the tanh activation function is shown in Equation (3). It maps any real number to the range (−1, 1). The advantage of the tanh function is that its output is centered at approximately 0, which often results in a faster learning process. However, similar to the sigmoid function, the gradient of the tanh function approaches 0 when the input values are large or small. This can lead to the vanishing gradient problem, making it difficult for the neural network to learn and update its weights.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3}$$

### 3.1.4. Pooling Layer

The pooling layer effectively improves the computational efficiency of the model and makes it more robust to positional changes by reducing the data dimensions and the number of parameters. Pooling operations aggregate the local regions of the input data, reducing the spatial dimensions of the data. Max and average pooling are two common pooling operations, as illustrated in Figure 2. Max pooling selects the maximum value from the local region of each pooling window as the output. This helps retain the most significant features in the image or feature map. Average pooling takes the average value of the local region within each pooling window as the output. It smooths the input data and reduces the sensitivity to local noise.



**Figure 2.** Feature pooling operation in the same color.

### 3.1.5. Fully Connected Layer

The fully connected layer is typically placed at the end of the network and is responsible for integrating the features extracted by the previous layers to produce the final prediction. The working principle of the fully connected layer is straightforward: each neuron is connected to all the neurons in the preceding layer. It performs a weighted sum of the outputs from the neurons of the previous layer, adds a bias term, and then passes the result through an activation function to produce the output. This process essentially learns the global relationships between the input features. This contrasts with the local focus of the convolutional and pooling layers; the introduction of fully connected layers enables the neural network to perform higher-level pattern recognition by learning global features. In a CNN, the convolutional and pooling layers primarily learn local features from the input, whereas the fully connected layer integrates these local features to produce the final prediction. This hierarchical structure allows CNNs to better capture the hierarchical and abstract representations of features when processing complex input data.

## 3.2. Loss Function

The loss function is used to measure the difference between the model output and the true labels. Its purpose is to optimize the model parameters by minimizing the loss function and improving the fit of the model to the training data and its performance on unseen data. The choice of loss function is crucial for the performance and training outcomes of the model. Cross-entropy loss is commonly used in image segmentation tasks. For imbalanced image classification, we propose the use of a weighted cross-entropy that is calculated via an inverse proportion exponential function to perform an auxiliary diagnosis of pneumonia.

### Improved Weighted Cross-Entropy

The cross-entropy loss function is employed to measure the difference between the predicted probability distribution of the model and the true probability distribution. For binary classification problems, the mathematical expression of the cross-entropy loss function is shown in Equation (4).

$$L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -[y_i \cdot \log(p_i) + (1 - y_i) \log(1 - p_i)], \quad (4)$$

where  $y_i$  represents the true label of sample  $i$  and  $p_i$  represents the probability that sample  $i$  is predicted as the positive class. For multiclass classification problems, the formula is defined as follows.

$$L = \frac{1}{N} \sum_i L_i = -\frac{1}{N} \sum_i \sum_{c=1}^M y_{ic} \log(p_{ic}), \quad (5)$$

where  $M$  represents the number of different classes in the samples.  $y_{ic}$  denotes the true class of sample  $i$ , and  $p_{ic}$  represents the predicted probability of belonging to class  $c$ . In practical applications, we often encounter the data imbalance problem, where the number of samples in some classes significantly exceeds that in others. In such scenarios, the use of a conventional cross-entropy loss function may cause the model to focus excessively on classes with more samples while neglecting those with fewer samples. This occurs because the model aims to minimize the total loss during training, and the total loss is the sum of losses for all samples. Consequently, classes with more samples have a greater influence on the total loss. To address data imbalance, we introduce the weighted cross-entropy loss function. By applying weights to different classes, we can adjust the loss function to a weighted form. A binary classification problem is defined as follows:

$$L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -[w_1 \cdot y_i \log(p_i) + w_2 \cdot (1 - y_i) \log(1 - p_i)], \quad (6)$$

where the weights  $w_1$  and  $w_2$  are the weights for the two classes in the binary classification problem. For multiclass classification problems, the formula for the weighted cross-entropy loss function is as follows:

$$L = \frac{1}{N} \sum_i L_i = -\frac{1}{N} \sum_i \sum_{c=1}^M w_c \cdot y_{ic} \log(p_{ic}), \quad (7)$$

where  $w_c$  is the weight assigned to the  $c$ -th class in the multiclass classification problem. Introducing weighted cross-entropy loss and setting appropriate weights helps address data imbalance. However, determining the optimal weights has always been challenging for researchers. To address this problem, we propose a method to calculate the weights in the weighted cross-entropy loss function via an IPEF. The calculation formula is as follows:

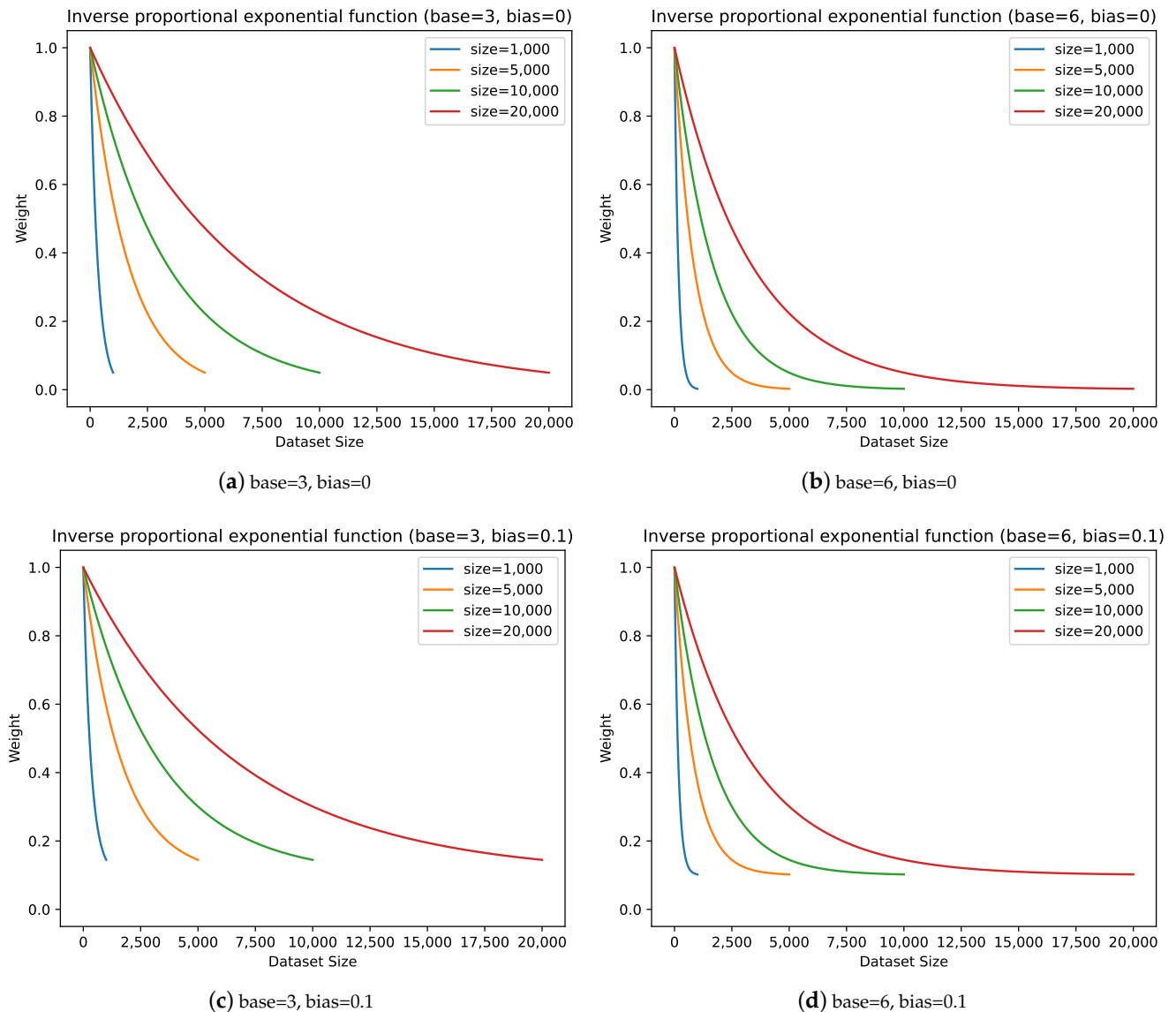
$$w_c = \text{bias} + (1 - \text{bias}) \cdot e^{-\alpha \cdot \text{size}_c}, \quad (8)$$

where *bias* is a bias term used to prevent the weights of high-frequency classes from dropping to zero. The calculation method for the parameter  $\alpha$  is defined as follows:

$$\alpha = \frac{\text{base}}{\text{size}}, \quad (9)$$

where *base* is the decay rate of the function, which affects the shape of the curve. The function curve tends to become concave as *base* increases; conversely, as *base* decreases, the curve becomes increasingly smooth. *size* represents the total number of samples in the entire dataset. We can dynamically adjust the weights for imbalanced data scenarios via the IPEF, thereby effectively improving the classification performance of the model on

imbalanced data. Figure 3 shows that the function maintains its basic shape across various data scales for different parameters, demonstrating good generalizability.



**Figure 3.** Inverse proportional exponential function with different parameters.

### 3.3. Transfer Learning

TL, a significant paradigm in machine learning, enables models to apply knowledge from one domain to other related, but different, domains. This concept is inspired by the way in which humans learn. We can shorten the model training time, reduce the need for large amounts of labeled data, and enhance the robustness of the model when the data distribution changes by transferring knowledge learned from one task. There are four main approaches to TL: sample-based, model-based, feature-based, and relation-based transfer. TL is widely applied in tasks such as visual object recognition [36], text classification [37], and speech recognition [38]. In computer vision, pretrained CNNs can be used on new image datasets.

TL addresses the issue of small sample datasets by performing similarity matching between source data and target domain data. Once the conditions are met, the transfer operation is completed. During the matching process, to reduce the distribution difference between the source dataset and the target domain dataset, the maximum mean discrepancy



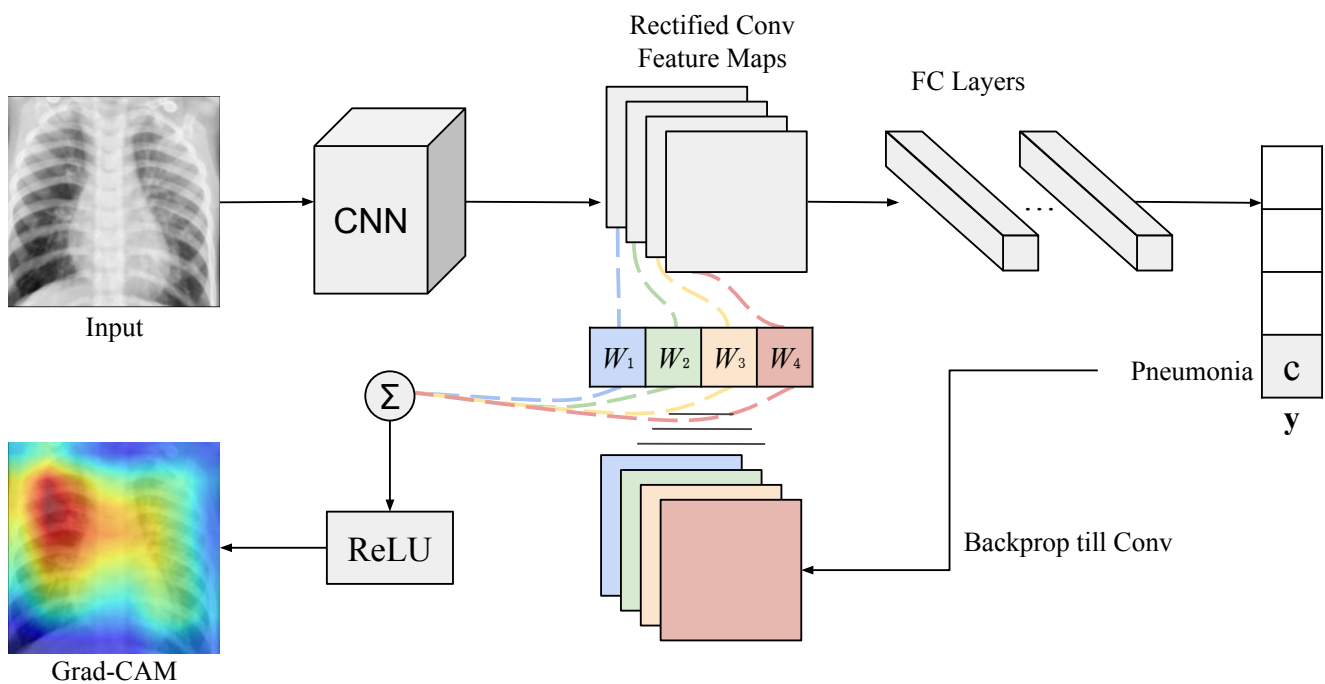
(MMD) is used primarily to measure the distribution distance between samples. It is defined as follows:

$$MMD(A, B) = \left\| \frac{1}{m} \sum_{i=1}^m H(a_i) - \frac{1}{n} \sum_{j=1}^n H(b_j) \right\|^2, \tag{10}$$

where  $a_i$  and  $b_j$  are two samples,  $m$  and  $n$  denote the sample sizes, and  $H$  is a function that maps the original variables into the reproducing kernel Hilbert space. If the MMD is sufficiently small, the distributions are considered approximately the same. Conversely, if the MMD is large, there is a significant distribution difference between the two. Through the fine-tuning method, models can quickly learn fine-grained features specific to the task [39]. In natural language processing, TL allows large language models such as GPT and bidirectional encoder representations from transformers (BERT) to perform well on specific downstream tasks [40]. In machine learning, TL also plays a role in improving learning efficiency and performance in situations where data are scarce [41].

### 3.4. Grad-CAM

In deep learning, interpretability and visualization techniques are crucial for understanding the decision-making process of models. Grad-CAM is a widely used visualization technique for convolutional neural networks. It generates heatmaps to highlight the salient regions in the input image that are important for predicting a specific class, thereby providing an intuitive explanation of the decision process of the model [42]. The basic idea of Grad-CAM is to use the gradient information of the predicted class to compute weights and generate the corresponding class activation map. Specifically, Grad-CAM first calculates the gradient of the target class with respect to the feature maps of the last convolutional layer through backpropagation. Then, by performing global average pooling on these gradients, the weights for each feature map are obtained. Finally, these weights are used to compute a weighted sum of the feature maps, followed by applying the ReLU activation function to generate a heatmap of the same size as the input image. The detailed implementation process is shown in Figure 4.



**Figure 4.** The detailed implementation process of Grad-CAM (Different colors correspond to different weights).

Grad-CAM helps researchers understand and interpret the CNN decision-making process and serves as a diagnostic tool to identify potential issues and weaknesses in the specific decisions of the model. For instance, in medical image analysis, Grad-CAM can assist doctors in locating and interpreting lesion areas, increasing trust in AI diagnostic results [43]. Through these analyses and visualizations, we can gain a deeper understanding of the behavior of the model, ensuring interpretability and reliability in clinical applications. This, in turn, enhances the effectiveness of AI models in actual medical diagnostics.

## 4. Experimental Studies

### 4.1. Dataset Description and Processing

The dataset used in this experiment was the COVID-19 radiography database [43,44]. This dataset was created by a team of researchers from Qatar University and Dhaka University in Bangladesh, along with collaborators and doctors from Pakistan and Malaysia. The dataset contains chest X-ray images (CXRs) of COVID-19-positive patients, normal lungs, and other pneumonia infections. The dataset was released in stages: the first version included 219 COVID-19 images, 1341 normal images, and 1345 viral pneumonia CXR images. In the first update, the number of images in the COVID-19 category increased to 1200. In the second version, the database was expanded to 3616 COVID-19-positive cases, 10,192 normal cases, 6012 non-COVID-19 lung infection (lung opacity) cases, and 1345 pneumonia images. The detailed distribution of each dataset is shown in Table 1.

**Table 1.** Detailed distribution of the dataset.

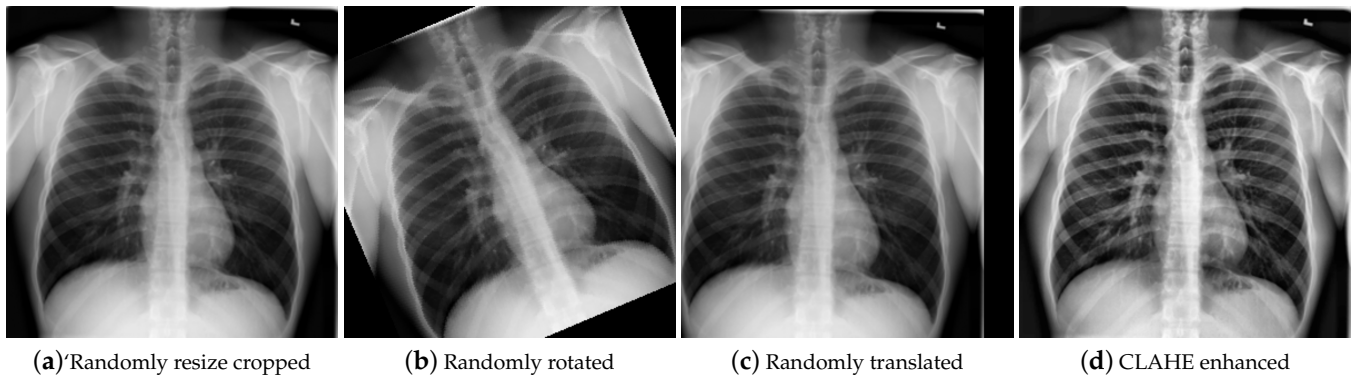
	Training Set	Validation Set	Test Set	Total
COVID-19	2351	555	710	3616
Lung opacity	3915	937	1160	6012
Normal	6478	1160	2071	10,192
Pneumonia	801	252	292	1345

To process the aforementioned dataset, we employed a series of data augmentation techniques to increase the quality of the training data and optimize model performance. These techniques have been proven by various studies to effectively improve the generalization ability and accuracy of models. The specific steps are as follows:

- **Resizing:** Random size cropping is performed on the images, followed by resizing them to  $224 \times 224$  pixels. This provides a consistent data foundation and enhances the robustness of the model to different perspectives and scales through the randomness of cropping (see Figure 5a). Random cropping is a basic data augmentation method that has been widely used [45].
- **Rotation and translation:** Random rotation and translation are applied to simulate changes in shooting angles in real-world scenarios, improving the applicability and accuracy of the model (see Figure 5b,c). Studies have shown that random rotation and translation significantly enhance the performance of the model in handling data with different shooting angles [46].
- **CLAHE image enhancement:** Contrast-limited adaptive histogram equalization (CLAHE) is used to improve image contrast, which is particularly suitable for CXR images, significantly enhancing image quality and better supporting model training (see Figure 5d). This method has been widely validated as effective in medical imaging [47].
- **Data normalization:** The three channels of the RGB images were normalized to mean values of 0.485, 0.456, and 0.406 and standard deviations of 0.229, 0.224, and 0.225. These parameters are considered effective normalization parameters in deep learning practice. Each channel of the data is normalized via these parameters, as expressed in

Equation (6). Equation (11), ensures a uniform distribution of the data, promoting the stability of neural network training [48].

$$Channel_{normalized} = \frac{Channel_{original} - mean}{std}. \quad (11)$$



**Figure 5.** Image augmentation.

#### 4.2. Experimental Setup and Parameter Settings

All algorithms in this study were implemented via Python. All the experiments were conducted on a personal PC with the following configuration: 13th Gen Intel(R) Core(TM) i7-13770F, 2.10 GHz CPU, and 32 GB of RAM. An NVIDIA GeForce RTX 4060 laptop GPU was used as the graphics accelerator. The software environment for running the experiments was PyTorch 2.1.

To conduct a more detailed and comprehensive validation of the performance of our method across different models and perform a comparative analysis, we selected four classic models with varying parameter sizes: AlexNet [49], VGG16 [50], GoogLeNet [51], and ResNet18 [52]. These models have a significant influence on deep learning, and each possesses unique structures and advantages, providing a multidimensional performance evaluation for our study:

- AlexNet is a milestone in deep learning, and its major contribution lies in achieving exceptional classification performance on the ImageNet dataset through a deep convolutional neural network. The core structure of AlexNet includes five convolutional layers and three fully connected layers. It introduces the ReLU activation function to accelerate training and uses dropout techniques to prevent overfitting. Additionally, AlexNet was the first model to use GPUs for large-scale parallel computing, significantly increasing the training speed.
- VGG16 increases network depth by using multiple stacked  $3 \times 3$  small convolutional kernels to extract high-level feature representations. VGG16 consists of 13 convolutional layers and three fully connected layers. Although the deeper structure increased the computational load, it demonstrated excellent performance on the ImageNet dataset. Its simple and uniform design makes it easy to transfer to other visual tasks.
- GoogLeNet (Inception V1) maintains relatively low computational complexity while capturing multiscale information through the inception module. The inception module fuses features of different scales through parallel convolution and pooling operations, better representing both local and global information in images. GoogLeNet has shown high efficiency and superior image classification performance across various computational platforms.
- ResNet18 is a member of the residual network family with 18 layers. Its core idea is to address the vanishing gradient and degradation problems in deep networks by introducing residual blocks. Residual blocks use skip connections to pass information directly between layers, ensuring effective gradient propagation. This makes it possible to train very deep networks.

We made corresponding adjustments to the structures of the aforementioned four models to accommodate the experimental requirements better. We replaced their original final fully connected layer (FC1000) with a fully connected layer (FC4) that matches the four-class classification requirement. This adjustment makes the models more suitable for data classification needs and provides a more accurate performance evaluation while preserving the model features. During training, we set the hyperparameters as follows: (a) The number of epochs was set to 30. The selection of 30 epochs ensured that the models had sufficient learning time on the dataset. (b) The batch size was set to 16, which helps balance computational efficiency and memory usage, making the training process more stable and efficient. (c) The initial learning rate was set to 0.01, with a decay to 50% of the original value every 5 epochs. This helps achieve rapid convergence in the early stages while fine-tuning the model parameters in later stages, thereby improving the final model accuracy.

For the parameter settings in the IPEF used to calculate the weights in the weighted cross-entropy loss function, we set the base to 6. This value adjusts the size of  $\alpha$ , adapting the calculated weights to the differences in the class distribution of the dataset. When the base value is larger, the value of  $\alpha$  is relatively smaller, and the rate of change in weights with class size is slower, preventing drastic changes in weights. This ensures that the model trains more smoothly and stably when handling different class samples. The bias was set to 0.1 to ensure that even if the number of samples in a particular class  $size_c$  was zero or very small, the minimum weight  $w_c$  remained near 0.1. This avoided making the classifier overly sensitive to small sample classes while ignoring larger sample classes.

#### 4.3. Evaluation Metrics

A confusion matrix is a summary tool that presents prediction results in tabular format. As Figure 6 shows, TP (true positive) and TN (true negative) denote the number of samples correctly predicted as positive and negative classes, respectively, and FP (false positive) and FN (false negative) represent the number of samples incorrectly predicted as positive and negative classes, respectively. It details the prediction accuracy for each class. This visual representation allows us to clearly identify the misclassifications made by the model, specifically which classes are easily misidentified as other classes by the model. This tool is crucial for analyzing and improving the classification performance of the model.

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Figure 6. Confusion matrix.

For classification problems, the accuracy based on the confusion matrix can be defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (12)$$

The receiver operating characteristic (ROC) curve is used to demonstrate the performance of a classifier under various classification threshold conditions. The true positive rate (TPR) and false positive rate (FPR) are calculated as shown in Equations (13) and (14). By plotting the relationship between the TPR and FPR at different thresholds, the ROC curve effectively evaluates and compares the performances of different classifiers.

$$TPR = \frac{TP}{TP + FP}, \quad (13)$$

$$FPR = \frac{FP}{FP + TN}, \quad (14)$$

where TP, TN, FP, and FN represent the number of samples predicted by the model. An efficient classifier can accurately identify all positive and negative cases, maximizing the TPR while minimizing the FPR, thereby demonstrating its strong classification capability and high prediction accuracy.

The area under the curve (AUC) represents the area under the ROC curve and ranges from 0 to 1. A higher AUC indicates better classifier performance. Therefore, the AUC reflects the ability of the classifier to distinguish between positive and negative cases. A higher AUC typically indicates that the model has better classification performance, providing more accurate positive predictions and correctly rejecting negative cases. In practice, the AUC is widely used as a simple and robust performance evaluation metric to compare different classifiers and select the optimal model.

#### 4.4. Performance Comparison

To verify the effectiveness and reliability of our proposed IPEWF, as well as to assess the impact of incorporating TL on model performance, we divided the comparative experiments into two parts. The first part involves a performance evaluation analysis of the four CNN models via the traditional loss function and the IPEWF. The second part discusses and evaluates whether the performance of the CNNs can be further improved by combining the TL with the IPEWF.

##### 4.4.1. Performance Comparison on Improved Cross-Entropy Loss Function

In this section, we select the four models described in Section 4.2 for comparative experiments. We conducted a comparative analysis between models using the conventional cross-entropy loss function and those using the inverse proportional exponential weighted cross-entropy loss function (IPEWF). As Table 2 shows, the accuracy of AlexNet (IPEWF) is 77.69%, which is an improvement of 1.41% compared with that of AlexNet when the conventional cross-entropy loss function (76.28%) is used. The accuracy of VGG16 (IPEWF) is 84.38%, an improvement of 2.22% compared with VGG16 when the conventional cross-entropy loss function (82.16%) is used. The accuracy of ResNet18 (IPEWF) is 85.40%, an improvement of 1.52% compared with ResNet18 when the conventional cross-entropy loss function (83.88%) is used. The accuracy of GoogLeNet (IPEWF) is 89.60%, an improvement of 1.20% compared with that of GoogLeNet when the conventional cross-entropy loss function (88.40%) is used. The accuracy of these models—AlexNet, VGG16, ResNet18, and GoogLeNet—improved to varying degrees after the IPEWF was adopted. This finding indicates that the IPEWF is effective in addressing data imbalance and can improve the overall performance of the models.

**Table 2.** Accuracy of models using the IPEWF.

	AlexNet	AlexNet (IPEWF)	VGG16	VGG16 (IPEWF)
Parameters		$5.70 \times 10^7$		$1.34 \times 10^8$
Accuracy	76.28%	<b>77.69%</b>	82.16%	<b>84.38%</b>
	ResNet18	ResNet18 (IPEWF)	GoogLeNet	GoogLeNet (IPEWF)
Parameters		$1.12 \times 10^7$		$9.94 \times 10^6$
Accuracy	83.88%	<b>85.40%</b>	88.40%	<b>89.60%</b>

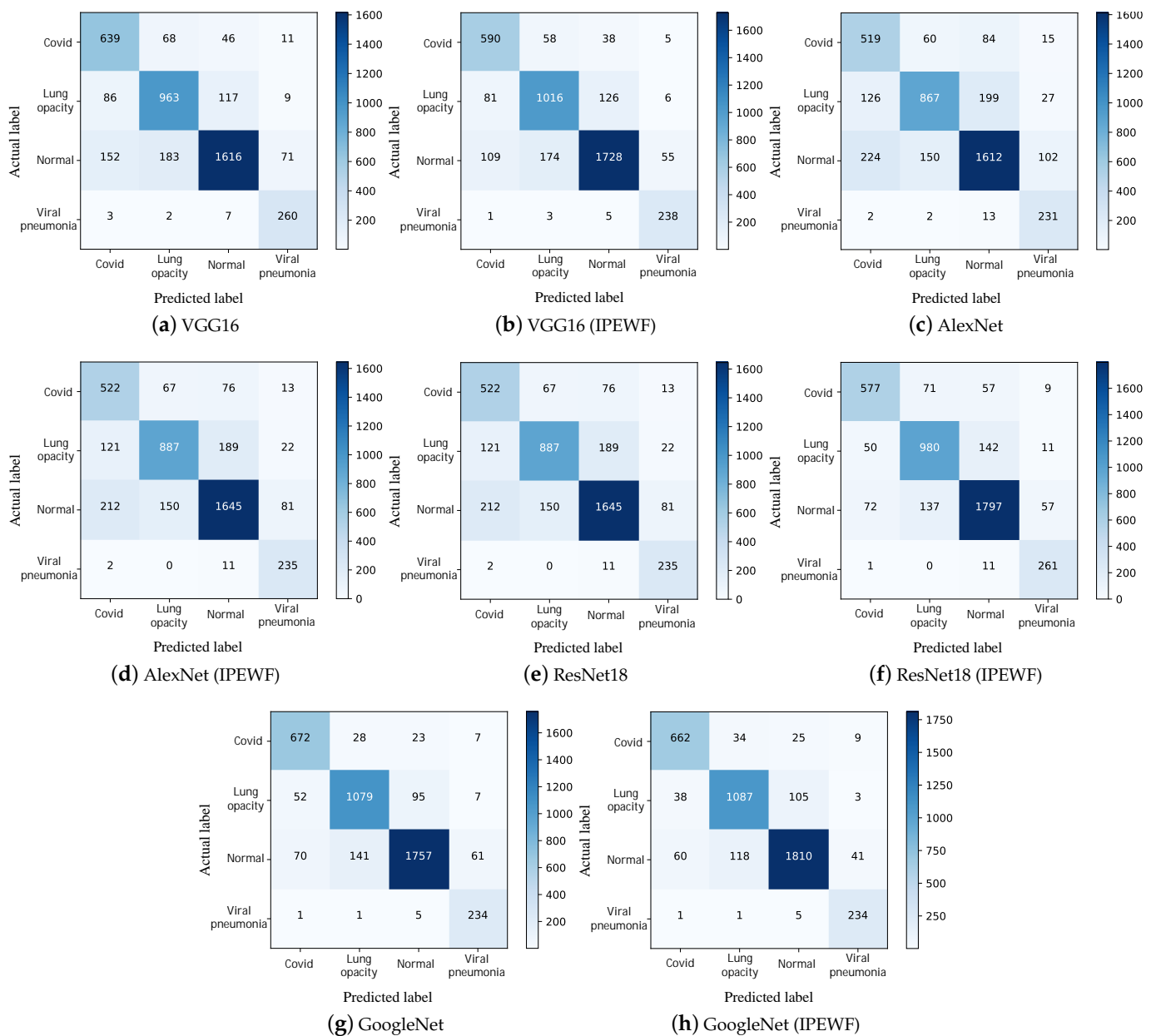
To evaluate the performance of various models comprehensively, we conducted a detailed analysis of the confusion matrices for each model. The confusion matrix reflects the number of correct classifications and clearly shows the misclassification situations for each class across different models. Figure 7 displays the classification results for the VGG16, AlexNet, ResNet18, and GoogLeNet models, along with their improved versions using the IPEWF. Figure 7a,b show the results of the VGG16 model and its IPEWF-improved version. The results indicate that the overall accuracy of the VGG16 model improved after applying IPEWF; however, its performance on small sample classes decreased. Figure 7c,d display the sample distributions of AlexNet and its IPEWF version. The IPEWF-improved AlexNet model maintained stable performance on large sample classes while showing some performance improvement on small sample classes. Figure 7e,f illustrate the confusion matrices for ResNet18 and its IPEWF version. The results demonstrate that the IPEWF-improved ResNet18 model significantly improved for small sample classes. Figure 7g,h show the predicted sample distributions for GoogLeNet and its IPEWF version. The results reveal that the IPEWF-improved GoogLeNet model maintains the classification performance on small sample classes and significantly improves the performance on large sample classes. In summary, although the IPEWF method did not perform well in terms of sensitivity to small samples in the VGG16 model, it performed well in the other three models. Additionally, the overall accuracy of all four models improved, indicating that our method has certain advantages in enhancing the overall model performance.

#### 4.4.2. Performance Comparison on TL

To further enhance model performance, we combined the IPEWF, which performed well in the previous section, with TL techniques. We continue to use AlexNet, VGG16, ResNet18, and GoogLeNet as our research objects. Table 3 shows that the accuracy of AlexNet (IPEWF+transfer) reached 90.36%, an improvement of 12.67% compared with that of AlexNet (IPEWF) (77.69%). The accuracy of VGG16 (IPEWF+transfer) was 93.97%, an increase of 9.59% compared with the 84.38% accuracy of VGG16 (IPEWF). ResNet18 (IPEWF+transfer) achieved an accuracy of 94.14%, an improvement of 8.74% over ResNet18 (IPEWF)'s 85.40%. The accuracy of GoogLeNet (IPEWF+transfer) was 92.58%, an increase of 2.98% compared with that of GoogLeNet (IPEWF), which was 89.60%. These results indicate that incorporating TL techniques led to a significant improvement in the accuracy of all four models, further demonstrating the significant role of TL in enhancing model accuracy in this study.

**Table 3.** Accuracy of models using the IPEWF.

	AlexNet (IPEWF)	AlexNet (IPEWF + Transfer)	VGG16 (IPEWF)	VGG16 (IPEWF + Transfer)
Parameters		$5.70 \times 10^7$		$1.34 \times 10^8$
Accuracy	77.69%	<b>90.36%</b>	84.38%	<b>93.97%</b>
	ResNet18 (IPEWF)	ResNet18 (IPEWF + Transfer)	GoogLeNet (IPEWF)	GoogLeNet (IPEWF + Transfer)
Parameters		$1.12 \times 10^7$		$9.94 \times 10^6$
Accuracy	85.40%	<b>94.14%</b>	89.60%	<b>92.58%</b>



**Figure 7.** The distribution of confusion matrices across the four sample datasets and the changes after applying IPEWF.

We continued to analyze the confusion matrices for the four models. By comparing Figure 8a,b, we find that the performance of the VGG16 (IPEWF+transfer) model, which combines the IPEWF method and TL techniques, improved significantly. Specifically, the number of correct classifications in the small sample class of pneumonia patients increased from 238 to 271. For the COVID-19 class, the number of correct classifications also increased significantly, from 590 to 670. Additionally, the correct classification numbers for the other two classes also showed substantial improvement. A comparison of Figure 8c,d reveals that the AlexNet (IPEWF + Transfer) model, which applies the IPEWF method combined with the TL technique, also has positive effects. The classification accuracy of the model improved significantly for both small and large sample classes. Comparing Figure 8e,f, the ResNet18 (IPEWF + Transfer) model, which incorporates TL techniques based on the IPEWF method, showed improved performance in most classes, although the correct classification number for the pneumonia class slightly decreased. However, it demonstrated better performance in other classes. A comparison of Figure 8g,h reveals

that the GoogLeNet (IPEWF + Transfer) model, which combines the IPEWF method and TL techniques, yields some differences in the confusion matrix. Although the correct classification number for the lung opacity class decreased slightly from 1087 to 1085, this minor performance loss is acceptable considering the overall performance improvement. More importantly, the correct classifications for the pneumonia class increased from 234 to 267, those for the COVID-19 class increased from 662 to 667, and those for the normal class increased from 1810 to 1888. In summary, by combining the IPEWF method and TL techniques, the classification performance of all four models significantly improved, especially in terms of accuracy for critical classes. This further validates the effectiveness of this approach in enhancing the overall performance of the models.

To further validate the effectiveness of our method, we specifically focused on the recognition ability of models improved by the IEWLF method and those improved by the TL method. By comparing the ROC curves of the various models under the combined methods, we can assess the classification performance of the models more intuitively. After the TL technique was applied, the AUC values for the COVID-19, lung opacity, normal, and pneumonia classes improved significantly. Figure 9 displays the ROC curves of the four models based on different strategies. The AUC results in Table 4 show that by combining the IEWLF method and TL techniques, we significantly improved the classification performance of the four models, especially in terms of the recognition ability for critical classes. These results further validate the effectiveness and potential of the proposed method.

**Table 4.** Accuracy of models using the IPEWF.

	COVID-19	Lung Opacity	Normal	Viral Pneumonia
VGG16 (IPEWF)	0.966	0.945	0.950	0.996
VGG16 (IPEWF + Transfer)	0.998	0.989	0.989	0.999
AlexNet (IPEWF)	0.921	0.915	0.898	0.994
AlexNet (IPEWF + Transfer)	0.991	0.976	0.976	0.998
ResNet18 (IPEWF)	0.972	0.950	0.953	0.996
ResNet18 (IPEWF + Transfer)	0.998	0.986	0.985	0.999
GoogLeNet (IPEWF)	0.989	0.972	0.974	0.998
GoogLeNet (IPEWF + Transfer)	0.997	0.984	0.984	0.999

Finally, we selected the best-performing model in the experiment for interpretability analysis. Figure 10a shows the Grad-CAM activation heatmaps for the VGG16 (IPEWF + Transfer) model. These heatmaps highlight the key areas in the original image, providing intuitive aid for understanding the decision-making process of the model. The left side of each subfigure shows the original X-ray image of the detection target, the middle side shows the Grad-CAM activation heatmap, revealing the specific image regions the model relies on during prediction, and the right side presents the composite image overlaying these key regions on the original image, emphasizing the areas the model considers highly relevant to the predicted class. Similarly, Figure 10b–d display the Grad-CAM activation heatmaps for the AlexNet (IPEWF + Transfer), ResNet18 (IPEWF + Transfer), and GoogLeNet (IPEWF + Transfer) models, respectively. By analyzing these heatmaps, we can see that the models effectively recognize lesion areas, demonstrating the correspondence between the features learned by the models and the key diagnostic regions in actual medical images. This interpretability analysis provides a strong visual basis for evaluating algorithm performance and offers a valuable tool for assisting doctors in diagnosis. By observing these heatmaps, doctors can better understand the basis of the model predictions, thereby increasing their trust in the diagnostic results of the model. This approach, which combines advanced algorithms and visualization techniques, has the potential to play a significant role in practical clinical applications. These visualization results enhance our understanding of the decision-making process and further validate the practicality and effectiveness of the proposed method in medical imaging diagnosis.



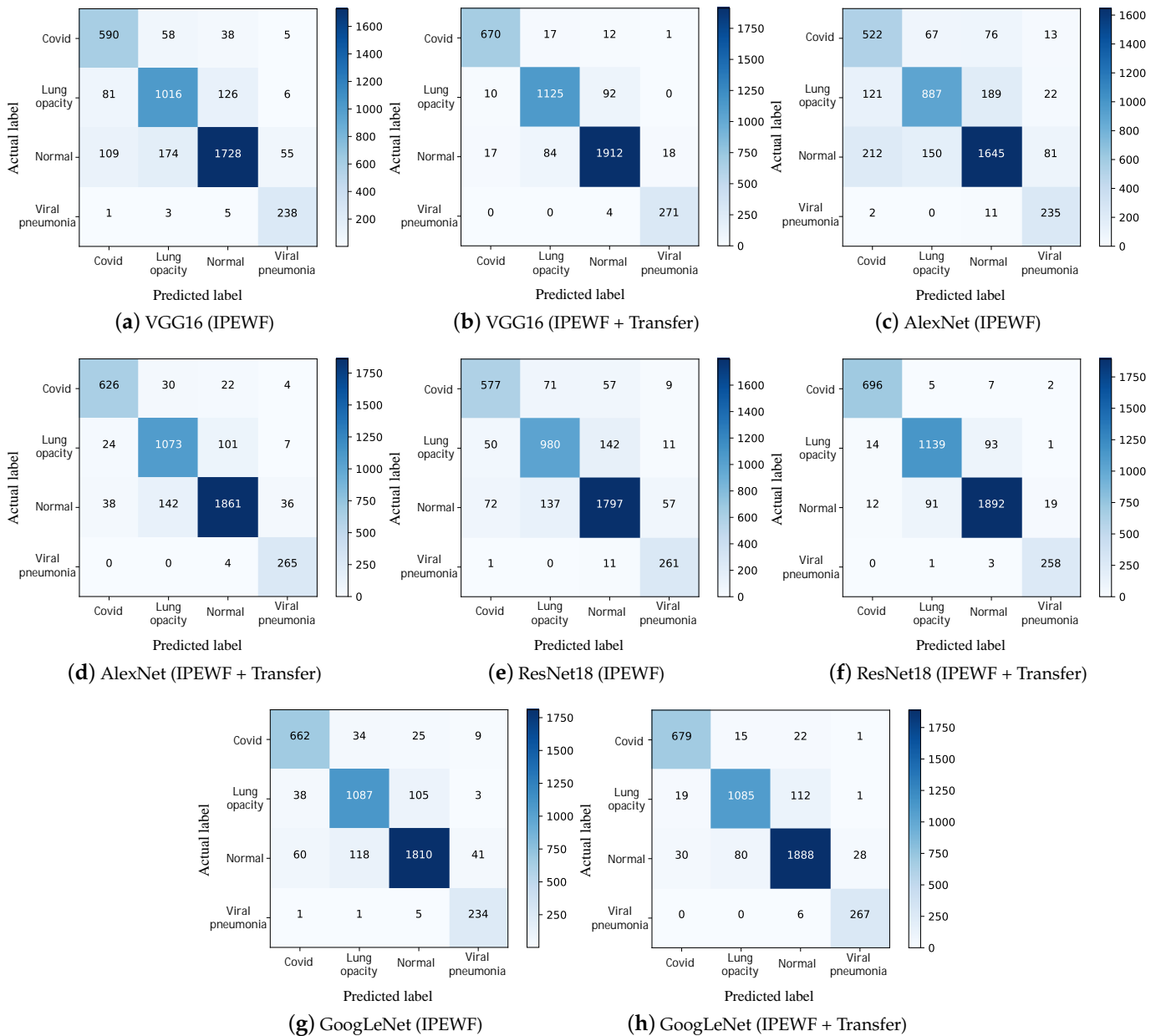


Figure 8. The distribution of confusion matrices across the four sample datasets and the changes after applying IPEWF combined with TL.

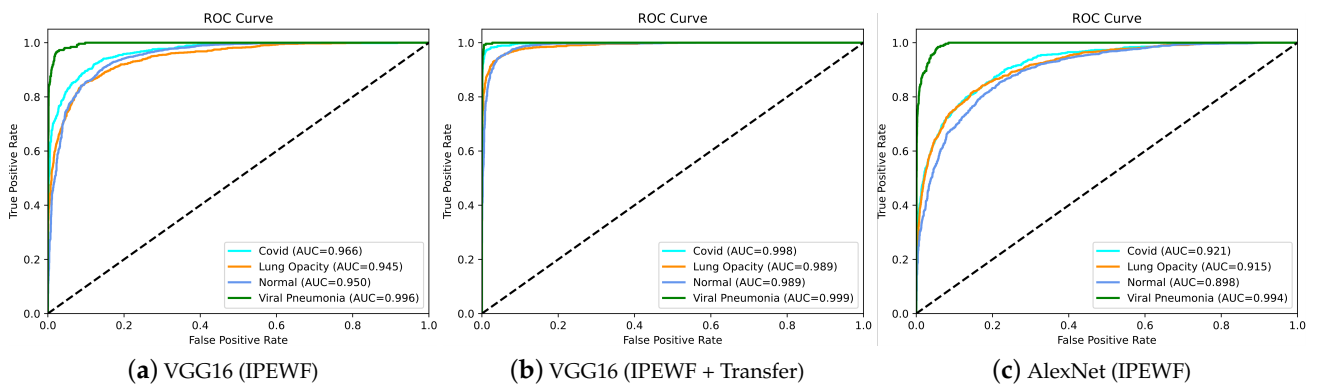
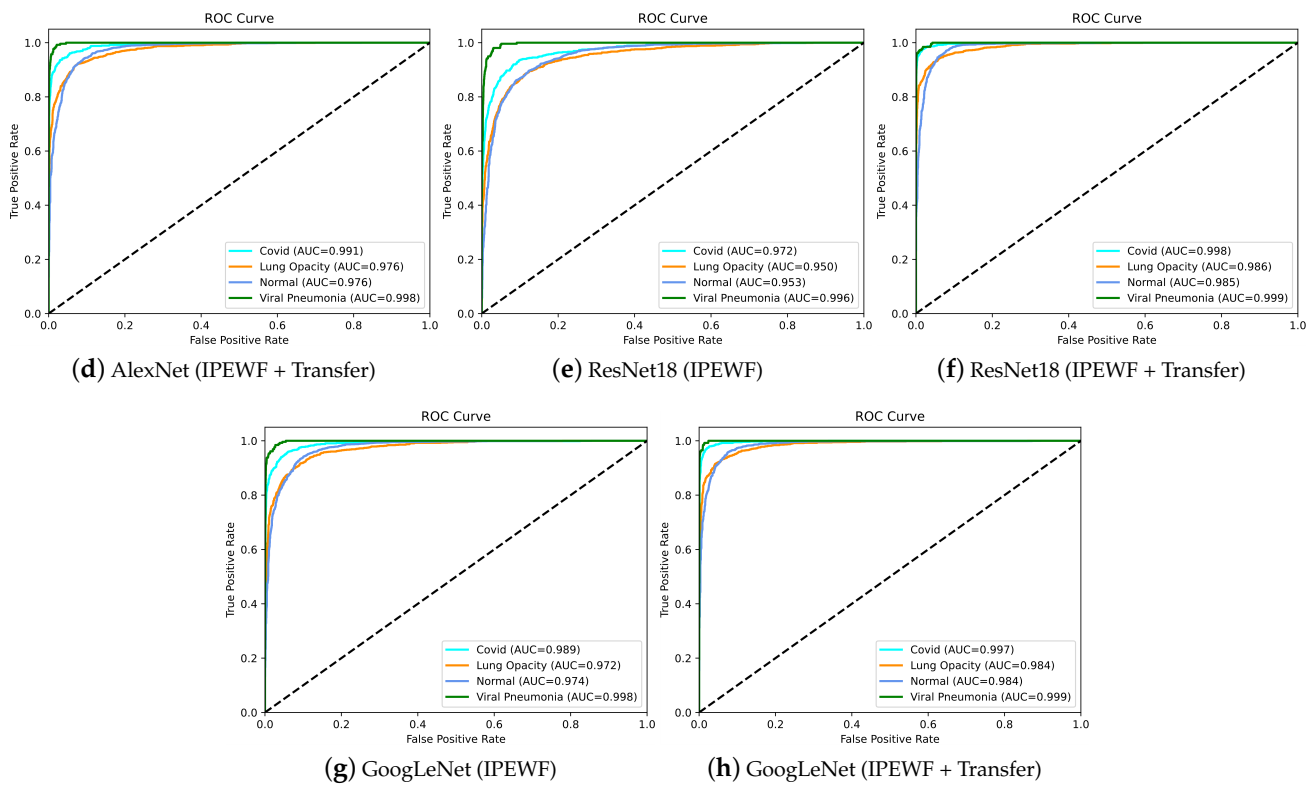
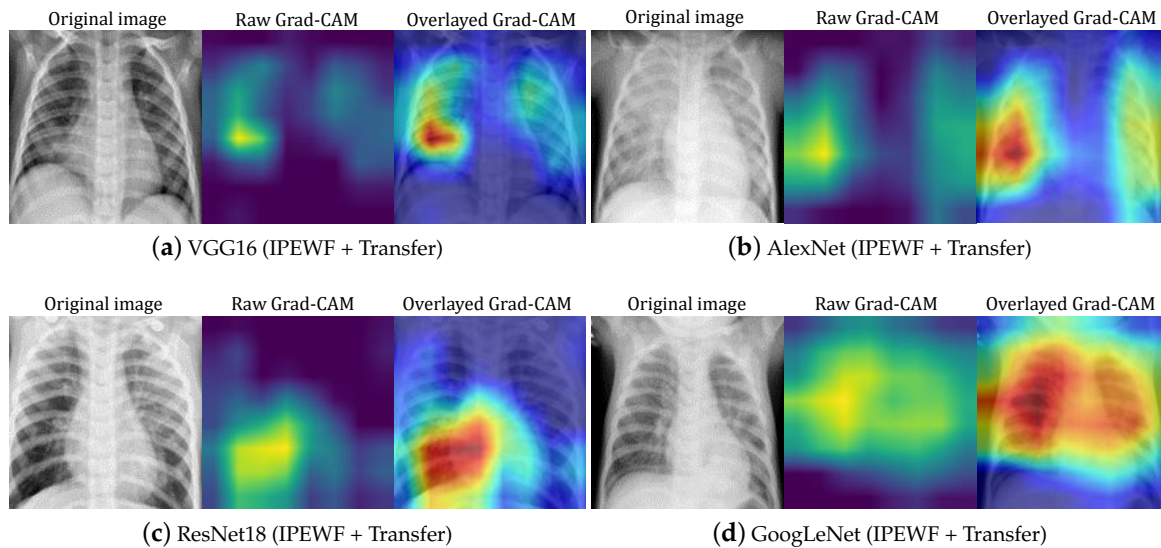


Figure 9. Cont.



**Figure 9.** The distribution of confusion matrices across the four sample datasets and the changes after applying IPEWF combined with TL.



**Figure 10.** Grad-CAM for four models.

### 5. Conclusions

This paper explores an automated diagnostic method for pneumonia that uses convolutional neural networks (CNNs). By applying CNNs, we enhanced the automation and accuracy of pneumonia diagnosis, reduced the reliance on medical resources, and improved diagnostic efficiency. However, CNNs face challenges in medical image diagnosis because of performance degradation caused by insufficient and imbalanced data, as well as poor model interpretability. To address these challenges, we designed a weighted cross-entropy loss function based on an inverse proportional exponential function and

optimized the models by incorporating transfer learning techniques to address the issue of imbalanced datasets. Additionally, we introduced gradient-weighted class activation mapping (Grad-CAM) technology to enhance model interpretability, providing more trustworthy decision support for clinical applications. The experimental results indicate that the proposed method significantly improved both the performance and interpretability of the models, with the accuracy of the four classic models increasing to over 90%. In this study, we identified a limitation due to the homogeneity of the dataset used, which may affect the generalizability of the models when applied to broader and more diverse data. In future work, we will use more diverse datasets for training and further optimize model performance. Additionally, while the inversely proportional exponential weighted cross-entropy loss function can partially address the issue of imbalanced training data, determining the most suitable parameter ratios still requires further experiments and validation.

**Author Contributions:** Conceptualization, Z.S. (Zhenyu Song), Z.S. (Zhanling Shi) and X.Y.; methodology, Z.S. (Zhenyu Song) and X.Y.; resources, X.Y. and S.S.; software, Z.S. (Zhenyu Song) and Z.S. (Zhanling Shi); writing—original draft, Z.S. (Zhenyu Song) and Z.S. (Zhanling Shi); writing—review and editing, Z.S. (Zhenyu Song), X.Y. and C.T.; visualization, X.Y.; validation, X.Y. and S.S.; formal analysis, Z.S. (Zhenyu Song), X.Y. and B.Z.; investigation, S.S.; data curation, B.Z.; project administration, Z.S. (Zhenyu Song); supervision, X.Y., Z.S. (Zhanling Shi) and C.T.; funding acquisition, Z.S. (Zhenyu Song) and S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was partially supported by the Qinglan Project of Jiangsu Universities, the Talent Development Project of Taizhou University (Grant No. TZXY2018QDJJ006), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20220619), the Young Science and Technology Talent Support Project of Taizhou, and the National Natural Science Foundation of China (Grant No. 62203069).

**Data Availability Statement:** The authors have used the publicly COVID-19 radiography datasets which was created by a team of researchers from Qatar University and Dhaka University in Bangladesh, along with collaborators and doctors from Pakistan and Malaysia. The link for the dataset is <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data> (accessed on 25 June 2024).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ciotti, M.; Ciccozzi, M.; Terrinoni, A.; Jiang, W.C.; Wang, C.B.; Bernardini, S. The COVID-19 pandemic. *Crit. Rev. Clin. Lab. Sci.* **2020**, *57*, 365–388. [[CrossRef](#)]
2. Miettinen, O.S.; Flegel, K.M.; Steurer, J. Clinical diagnosis of pneumonia, typical of experts. *J. Eval. Clin. Pract.* **2008**, *14*, 343–350. [[CrossRef](#)]
3. Portugal, I.; Alencar, P.; Cowan, D. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Syst. Appl.* **2018**, *97*, 205–227. [[CrossRef](#)]
4. Khanal, S.S.; Prasad, P.; Alsadoon, A.; Maag, A. A systematic review: Machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **2020**, *25*, 2635–2664. [[CrossRef](#)]
5. Liu, L. e-Commerce Personalized Recommendation Based on Machine Learning Technology. *Mob. Inf. Syst.* **2022**, *2022*, 1761579. [[CrossRef](#)]
6. Han, T.; Liu, C.; Yang, W.; Jiang, D. Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application. *ISA Trans.* **2020**, *97*, 269–281. [[CrossRef](#)] [[PubMed](#)]
7. Han, T.; Liu, C.; Yang, W.; Jiang, D. A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults. *Knowl.-Based Syst.* **2019**, *165*, 474–487. [[CrossRef](#)]
8. Chang, Z.; Zhang, A.J.; Wang, H.; Xu, J.; Han, T. Photovoltaic Cell Anomaly Detection Enabled by Scale Distribution Alignment Learning and Multi-Scale Linear Attention Framework. *IEEE Internet Things J.* **2024**, early access.
9. Melati, D.; Grinberg, Y.; Kamandar Dezfouli, M.; Janz, S.; Cheben, P.; Schmid, J.H.; Sánchez-Postigo, A.; Xu, D.X. Mapping the global design space of nanophotonic components using machine learning pattern recognition. *Nat. Commun.* **2019**, *10*, 4775. [[CrossRef](#)]
10. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [[CrossRef](#)]
11. Chen, Y.; Wang, S.; Lin, L.; Cui, Z.; Zong, Y. Computer Vision and Deep Learning Transforming Image Recognition and Beyond. *Int. J. Comput. Sci. Inf. Technol.* **2024**, *2*, 45–51. [[CrossRef](#)]

12. Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical image analysis using convolutional neural networks: A review. *J. Med. Syst.* **2018**, *42*, 1–13. [[CrossRef](#)] [[PubMed](#)]
13. Szepesi, P.; Szilágyi, L. Detection of pneumonia using convolutional neural networks and deep learning. *Biocybern. Biomed. Eng.* **2022**, *42*, 1012–1022. [[CrossRef](#)]
14. Rahman, T.; Chowdhury, M.E.; Khandakar, A.; Islam, K.R.; Islam, K.F.; Mahbub, Z.B.; Kadir, M.A.; Kashem, S. Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. *Appl. Sci.* **2020**, *10*, 3233. [[CrossRef](#)]
15. Jha, D.; Riegler, M.A.; Johansen, D.; Halvorsen, P.; Johansen, H.D. Doubleu-net: A deep convolutional neural network for medical image segmentation. In Proceedings of the 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS), Rochester, MN, USA, 28–30 July 2020; pp. 558–564.
16. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. [[CrossRef](#)]
17. Vinogradova, K.; Dibrov, A.; Myers, G. Towards interpretable semantic segmentation via gradient-weighted class activation mapping (student abstract). In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13943–13944.
18. Goel, N.; Yadav, A.; Singh, B.M. Medical image processing: A review. In Proceedings of the 2016 Second International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH), Ghaziabad, India, 18–19 November 2016; pp. 57–62.
19. Tang, J.; Deng, C.; Huang, G.B. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 809–821. [[CrossRef](#)]
20. Song, Y.Y.; Ying, L. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130. [[PubMed](#)]
21. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Washington, DC, USA, 4–10 August 2001; Volume 3, pp. 41–46.
22. Ji, J.; Tang, C.; Zhao, J.; Tang, Z.; Todo, Y. A survey on dendritic neuron model: Mechanisms, algorithms and practical applications. *Neurocomputing* **2022**, *489*, 390–406. [[CrossRef](#)]
23. Song, Z.; Tang, Y.; Ji, J.; Todo, Y. Evaluating a dendritic neuron model for wind speed forecasting. *Knowl.-Based Syst.* **2020**, *201*, 106052. [[CrossRef](#)]
24. Song, Z.; Tang, C.; Song, S.; Tang, Y.; Li, J.; Ji, J. A complex network-based firefly algorithm for numerical optimization and time series forecasting. *Appl. Soft Comput.* **2023**, *137*, 110158. [[CrossRef](#)]
25. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
26. Osareh, A.; Shadgar, B. Classification and diagnostic prediction of cancers using gene microarray data analysis. *J. Appl. Sci.* **2009**, *9*, 459–468. [[CrossRef](#)]
27. Yahyaoui, A.; Yumuşak, N. Decision support system based on the support vector machines and the adaptive support vector machines algorithm for solving chest disease diagnosis problems. *Biomed. Res.* **2018**. [[CrossRef](#)]
28. Nalepa, J.; Kawulok, M. Selecting training sets for support vector machines: A review. *Artif. Intell. Rev.* **2019**, *52*, 857–900. [[CrossRef](#)]
29. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
30. Anthimopoulos, M.; Christodoulidis, S.; Christe, A.; Mougiakakou, S. Classification of interstitial lung disease patterns using local DCT features and random forest. In Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 26–30 August 2014; pp. 6040–6043.
31. Bhattacharjee, A.; Murugan, R.; Goel, T. A hybrid approach for lung cancer diagnosis using optimized random forest classification and K-means visualization algorithm. *Health Technol.* **2022**, *12*, 787–800. [[CrossRef](#)]
32. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [[CrossRef](#)] [[PubMed](#)]
33. Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv* **2017**, arXiv:1711.05225.
34. Gaba, S.; Budhiraja, I.; Kumar, V.; Garg, S.; Kaddoum, G.; Hassan, M.M. A federated calibration scheme for convolutional neural networks: Models, applications and challenges. *Comput. Commun.* **2022**, *192*, 144–162. [[CrossRef](#)]
35. Xie, Y.; Zaccagna, F.; Rundo, L.; Testa, C.; Agati, R.; Lodi, R.; Manners, D.N.; Tonon, C. Convolutional neural network techniques for brain tumor classification (from 2015 to 2022): Review, challenges, and future perspectives. *Diagnostics* **2022**, *12*, 1850. [[CrossRef](#)]
36. Falco, P.; Lu, S.; Natale, C.; Pirozzi, S.; Lee, D. A transfer learning approach to cross-modal object recognition: From visual observation to robotic haptic exploration. *IEEE Trans. Robot.* **2019**, *35*, 987–998. [[CrossRef](#)]
37. Do, C.B.; Ng, A.Y. Transfer learning for text classification. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; Volume 18.
38. Shivakumar, P.G.; Georgiou, P. Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Comput. Speech Lang.* **2020**, *63*, 101077. [[CrossRef](#)]
39. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.

40. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
41. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [[CrossRef](#)]
42. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
43. Chowdhury, M.E.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; Al Emadi, N.; et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* **2020**, *8*, 132665–132676. [[CrossRef](#)]
44. Rahman, T.; Khandakar, A.; Qiblawey, Y.; Tahir, A.; Kiranyaz, S.; Kashem, S.B.A.; Islam, M.T.; Al Maadeed, S.; Zughaier, S.M.; Khan, M.S.; et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Comput. Biol. Med.* **2021**, *132*, 104319. [[CrossRef](#)]
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
46. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [[CrossRef](#)]
47. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Comput. Vis. Graph. Image Process.* **1987**, *39*, 355–368. [[CrossRef](#)]
48. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
49. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Van Esesn, B.C.; Awwal, A.A.S.; Asari, V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* **2018**, arXiv:1803.01164.
50. Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 169–175.
51. Yoo, H.J. Deep convolution neural networks in computer vision: A review. *IEIE Trans. Smart Process. Comput.* **2015**, *4*, 35–43. [[CrossRef](#)]
52. Ullah, A.; Elahi, H.; Sun, Z.; Khatoon, A.; Ahmad, I. Comparative analysis of AlexNet, ResNet18 and SqueezeNet with diverse modification and arduous implementation. *Arab. J. Sci. Eng.* **2022**, *47*, 2397–2417. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.