

Article

Pneumonia Recognition by Deep Learning: A Comparative Investigation

Yuting Yang and Gang Mei * 

School of Engineering and Technology, China University of Geosciences (Beijing), Beijing 100083, China;
yuting.yang@email.cugb.edu.cn

* Correspondence: gang.mei@cugb.edu.cn

Abstract: Pneumonia is a common infectious disease. Currently, the most common method of pneumonia identification is manual diagnosis by professional doctors, but the accuracy and identification efficiency of this method is not satisfactory, and computer-aided diagnosis technology has emerged. With the development of artificial intelligence, deep learning has also been applied to pneumonia diagnosis and can achieve high accuracy. In this paper, we compare five deep learning models in different situations for pneumonia recognition. The objective was to employ five deep learning models to identify pneumonia X-ray images and to compare and analyze them in different cases, thus screening out the optimal model for each type of case to improve the efficiency of pneumonia recognition and further apply it to the computer-aided diagnosis of pneumonia species. In the proposed framework: (1) datasets are collected and processed, (2) five deep learning models for pneumonia recognition are built, (3) the five models are compared, and the optimal model for each case is selected. The results show that the LeNet5 and AlexNet models achieved better pneumonia recognition for small datasets, while the MobileNet and ResNet18 models were more suitable for pneumonia recognition for large datasets. The comparative analysis of each model under different situations can provide a deeper understanding of the efficiency of each model in identifying pneumonia, thus making the practical application and selection of deep learning models for pneumonia recognition more convenient.

Keywords: pneumonia recognition; deep learning; X-ray; CNN; transformer



Citation: Yang, Y.; Mei, G.

Pneumonia Recognition by Deep Learning: A Comparative Investigation. *Appl. Sci.* **2022**, *12*, 4334. <https://doi.org/10.3390/app12094334>

Academic Editors: Keun Ho Ryu and Nipon Theera-Umporn

Received: 24 March 2022

Accepted: 24 April 2022

Published: 25 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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 usually triggered by pathogenic infections, such as bacteria or viruses [1,2]. If pneumonia is not promptly treated, it can pose a great threat to the life and health of the patient [3,4]. Therefore, early identification of pneumonia is very important for timely detection and treatment. Chest X-ray images are widely used in the identification of pneumonia because of their affordability and the rapid film formation of X-ray images [5,6]. Currently, the most common method for pneumonia identification is manual identification by X-ray images by a physician or specialist sitting in the clinic [7]. This method is subjective, has high fluctuation in accuracy, relies heavily on the clinical experience of the diagnosing physician and is less efficient [8,9]. In less-developed regions, the lack of specialized doctors and specialized medical equipment prevents the timely diagnosis of pneumonia. As a result, the mortality rate caused by pneumonia is high in less-developed countries and regions, which seriously affects the life and health of people.

To improve the efficiency and accuracy of pneumonia recognition, computer-aided pneumonia diagnosis techniques have been utilized. With the rapid development of deep learning [10,11], this has been widely employed in the medical field [12], including for pneumonia recognition [13]. Due to the high accuracy and robustness of deep learning, the efficiency of pneumonia diagnosis has been greatly improved [14,15].

Various scholars have conducted extensive studies on the application of deep learning in pneumonia recognition.

For example, Jaiswal et al. [8] achieved pneumonia localization and recognition in chest X-ray images based on a deep learning approach. The adopted method makes key modifications to the training process and adds a new post-processing step that merges bounding boxes from multiple models. Karakanis et al. [16] proposed two models for the employed dataset with deep learning following a lightweight architecture, which was employed to detect COVID-19. Compared with the other model, the deep learning model was found to achieve higher accuracy and to be more robust and reliable. However, the study only compared the deep learning model with the ResNet model in the classification for COVID-19 and the dataset used was small. Gazda et al. [17] proposed a self-supervised deep neural network. The evaluation results indicated that the method could achieve better recognition without requiring a large quantity of labeled training data. The results were compared for different datasets, but not with other deep learning model approaches.

Panthakkan et al. [18] proposed an efficient deep learning model for COVID-19 recognition, the COVID-DeepNet model. The model can identify COVID-19, non-COVID-19 pneumonia, and normal conditions with high accuracy. However, the model was designed for small datasets only, and it is not possible to assess the model's pneumonia recognition with large datasets. Wang et al. [19] proposed the "deep fractional max pooling neural network (DFMPNN)" model, which replaced the common max pooling and average pooling methods of neural networks with a new pooling method, and obtained a higher accuracy for pneumonia recognition. However, the dataset used was relatively small, and more advanced pooling techniques could be tested. Alhudhaif et al. [20] developed a reliable convolutional neural network model for performing X-ray image classification of COVID-19. The results showed that the CNN model based on the DenseNet-201 architecture had the highest accuracy, precision, recall, and F1 score. However, in this paper, only three types of models were compared: DenseNet201, ResNet18 and SqueezeNet.

Tahir et al. [21] classified X-ray images of COVID-19, SARS, and MERS using deep convolutional neural networks. Four algorithms with excellent performance were finally reported: SqueezeNet, ResNet18, InceptionV3, and DenseNet201. A corresponding comparative analysis was performed. This article used diverse datasets, but the datasets were still small and no operations to expand the dataset, such as data augmentation, were used. Loey et al. [22] proposed a Bayesian optimization-based convolutional neural network model for pneumonia recognition in chest X-ray images. In practical application, this model was shown to be more accurate and reliable. However, a comparative analysis with other models in terms of computational efficiency was lacking in this paper. Sitaula et al. designed a novel attention-based deep learning model [23], based on the VGG-16 deep learning model, with a new feature extraction method based on the bag of deep visual words (BoDVW) [24], and a novel feature extraction method based on the multi-scale bag of deep visual words (MBoDVW) [25], which greatly improved the stability and accuracy of the deep learning model for COVID-19 disease diagnosis. There was still the limitation that the dataset needed further processing to improve the model performance. A comparative analysis with other models in terms of computational efficiency was lacking in this paper.

As illustrated in Table 1, many deep learning models have been proposed for pneumonia recognition. However, there are still few studies related to the comparative analysis of the effectiveness of typical deep learning models for pneumonia recognition. In complex and changing real-life situations, it is difficult to efficiently select the appropriate deep learning model for pneumonia recognition. The comparative analysis of typical deep learning models for pneumonia recognition in various situations is beneficial for selecting the most appropriate pneumonia recognition model in different real-life situations, thus making pneumonia diagnosis more efficient. In this paper, the LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models are compared in pneumonia recognition to derive the optimal pneumonia recognition models in various situations to provide a more theoretical basis for pneumonia recognition in real-life applications to improve the efficiency and accuracy of pneumonia recognition to better maintain people's lives and health.

Table 1. Comparison of related studies.

| Author | Contributions | Limitations |
|------------------------|---|--|
| Jaiswal et al. [8] | The research achieved pneumonia localization and recognition in chest X-ray images based on a deep learning approach. | There was no corresponding comparative assessment of pneumonia recognition accuracy and computational efficiency. |
| Karakanis et al. [16] | The research proposed two models for the employed dataset with deep learning following a lightweight architecture. | A comparison with the ResNet model was performed only for the classification of COVID-19 and with a small dataset. |
| Gazda et al. [17] | The research proposed a self-supervised deep neural network which could achieve better recognition without requiring a large quantity of labeled training data. | The results were compared on different datasets, but not with other deep learning model approaches |
| Panthakkan et al. [18] | The research proposed an efficient deep learning model for COVID-19 recognition, the COVID-DeepNet model. | The model was designed for small datasets only, and it is not possible to judge the model's pneumonia recognition under large datasets. |
| Wang et al. [19] | The research proposed the “deep fractional max pooling neural network (DFMPNN)” model, which obtained a higher accuracy of pneumonia recognition. | The dataset was relatively small and more advanced pooling techniques could be tested. |
| Alhudhaif et al. [20] | The research developed a reliable convolutional neural network model for performing X-ray image classification of COVID-19. | The model comparison analysis was limited, only three types of model were compared. |
| Tahir et al. [21] | The research classified X-ray images of COVID-19, SARS, and MERS using deep convolutional neural networks. | Diverse datasets were collected, but the datasets were still small and no operations to expand the datasets, such as data augmentation, were used. |
| Loey et al. [22] | The research proposed a Bayesian optimization-based convolutional neural network model for pneumonia recognition in chest X-ray images. | Comparative analysis with other models in terms of computational efficiency was lacking. |
| Sitaula et al. [23–25] | The research proposed three deep learning methods for COVID-19 pneumonia detection with high stability and accuracy. | The dataset needed further processing to improve the model's performance. |

First, we collected chest X-ray image datasets of pneumonia patients and normal subjects and organized and cleaned the datasets. Second, we built five pneumonia recognition deep learning models, LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer for pneumonia recognition. Third, the five pneumonia recognition deep learning models, LeNet, AlexNet, MobileNet, ResNet18, and Vision Transformer models were compared in various cases to filter out the optimal model for the application in actual pneumonia recognition. For practical applications, the comparative analysis of classical deep learning

models in this paper, contributes to the selection of deep learning models for pneumonia recognition in different situations. The analysis will be beneficial to further improve the efficiency and accuracy of pneumonia recognition.

The contributions of this paper can be summarized as follows.

1. Five classical deep learning models are compared, and the advantages and disadvantages of each model are analyzed.
2. The computational accuracy and efficiency of each model in different situations is compared.
3. Suggestions are given for the selection of deep learning models in different situations. This is beneficial for the rapid selection of suitable deep learning models in practical applications for pneumonia recognition to improve efficiency.

The rest of this paper is organized as follows. Section 2 describes the materials and methods in detail. Section 3 analyzes the comparison of the results in this paper. Section 4 provides discussion as well as considering possible future research. Section 5 concludes the paper.

2. Methods

2.1. Overview

In this paper, we compare the effectiveness of the LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models in pneumonia recognition, screen out the optimal models in various situations, and finally apply them to pneumonia recognition in practice to improve pneumonia recognition efficiency and maintain people's lives and health.

First, we collected chest X-ray image datasets of pneumonia patients and normal subjects. The datasets were organized and cleaned to obtain a suitable dataset for deep learning models. Second, we built five pneumonia recognition deep learning models, LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer for pneumonia recognition. GPU loading and data augmentation operations were also performed to compare the effectiveness of each model for pneumonia identification under different conditions. Third, the accuracy and computational efficiency of the five pneumonia recognition deep learning models, LeNet, AlexNet, MobileNet, ResNet18, and Vision Transformer models were compared in various cases and a comparative analysis was performed to filter out the optimal pneumonia recognition model for various realistic situations. The workflow of this paper is illustrated in Figure 1.

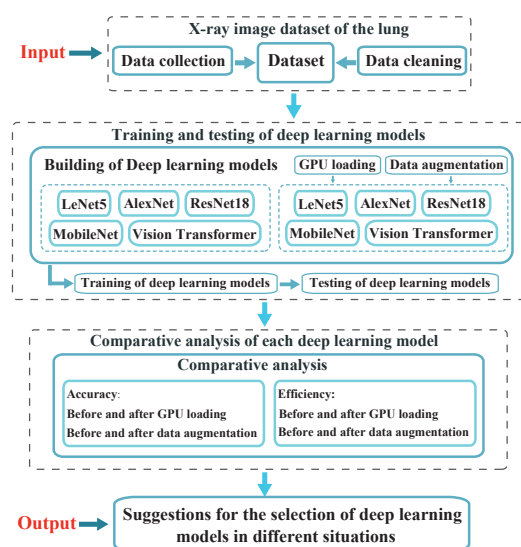


Figure 1. Workflow for pneumonia recognition.

2.2. Step 1: Data Collection and Cleaning

2.2.1. Data Collection and Processing

We acquired chest X-ray images of pneumonia patients and normal subjects from Guangzhou Women's and Children's Medical Center through a public dataset [26]. To ensure the model training and testing accuracy, we manually filtered the dataset to remove the more inconsistent and duplicate data, and only kept the pneumonia X-ray images of high quality. In addition, since there were more pneumonia X-ray images than normal X-ray images in the dataset collected in this paper, we removed some of the pneumonia X-ray images. Eventually, we selected the same number of images of pneumonia patients and normal subjects. Eighty percent of these were randomly selected as the training set, and twenty percent were selected as the testing set. To facilitate the training and testing, each image was scaled to a uniform size, such as 227×227 , before the training and testing.

The availability of the obtained datasets is as follows:

Dataset: Chest X-ray Images for Classification.

URL: <https://data.mendeley.com/datasets/rscbjbr9sj/2> (accessed on 1 February 2022).

2.2.2. Data Augmentation

Due to the small number of X-ray images and small datasets in this paper, data augmentation was employed to improve the model accuracy. Data augmentation refers to the effect of expanding the dataset by generating new images by performing a series of transformations on the images [27,28]. The employment of data augmentation can improve the generalization ability of the model and reduce the possibility of model overfitting.

Data augmentation generally includes image flipping, image scaling, image rotation, image cropping, adding disturbance factors, and color transformation of images. In this paper, image flipping, image scaling, image rotation, image cropping, image brightness, hue, and contrast transformations were used to achieve data augmentation. Examples of the functions that control the image flipping operation are: *transforms.RandomHorizontalFlip()*, *transforms.RandomVerticalFlip()*, denotes random horizontal flip and random vertical flip respectively. Examples of the functions that control the image scaling operation are: *transforms.Resize(1812,700)*, indicates that the image is deflated to 1812×700 size. Examples of the functions that control the image rotation operation are: *transforms.RandomRotation(30)*, indicates a random (clockwise or counterclockwise) rotation of 30° . Examples of the functions that control the image cropping operation are: *transforms.RandomCrop(500)* or *transforms.CenterCrop(500)*, indicates cropping an image of 500×500 size at the random position or cropping an image of 500×500 size at the center position. Examples of the functions that control the image brightness, hue, and contrast transformations operation are: *transforms.ColorJitter(brightness = 0.5, hue = 0.5, contrast = 0.5)*, indicates that the brightness, hue and contrast vary randomly from -0.5 to 0.5 .

2.3. Step 2: Deep Learning Model Construction for Pneumonia Recognition

2.3.1. Deep Learning Model Construction

In this paper, five classical deep learning models were selected for building deep learning models for pneumonia X-ray image recognition. The details of each model are as follows:

(1) LeNet5 [29]: The LeNet5 model is a classical CNN model with a simple structure and few network layers, with seven layers. In this paper, the most classical LeNet5 model was used for pneumonia recognition and compared with other models for analysis, thus observing the effect of classical CNN network structure in pneumonia recognition.

(2) AlexNet [30]: The AlexNet model is also a classical convolutional neural network model, which was the first convolutional neural network to attract attention due to winning the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition. Compared with the LeNet5 model, the AlexNet model has been improved in all aspects, especially in terms of network complexity and depth. The AlexNet model includes eight layers with

weights and three pooling layers and uses the ReLu activation function instead of the original sigmoid activation function, thus improving the model's recognition accuracy.

(3) ResNet18 [31]: The ResNet model is also one of the classical CNN models. The ResNet model firstly introduces a residual block to alleviate the gradient disappearance problem, of which the ResNet18 model is the earlier-appearing ResNet model. In this paper, the ResNet18 model was used for pneumonia recognition to examine the effect of adding the residual block on pneumonia recognition.

(4) MobileNet [32]: The MobileNet model is a lightweight deep neural network model proposed by Google in 2017, which is generally used for devices with low computational power requirements but high computational speed requirements, mainly applying the deep separable convolution module to improve computational efficiency. In this paper, the MobileNet model was chosen in the expectation that the analysis of the MobileNet model in pneumonia recognition could be used to further investigate the feasibility of mobile pneumonia recognition in the future.

(5) Vision Transformer [33]: The Vision Transformer model was developed from Transformer in the field of natural language processing (NLP), which is a combination of the field of computer natural language processing and computer vision. Vision Transformer utilizes the method of processing tokens in NLP to process image blocks and achieves better results and higher accuracy. The Vision Transformer model is a typical model combining NLP with CV. Through the analysis of the Vision Transformer model pneumonia recognition effect, the prospect of its application in the field of pneumonia recognition was explored.

2.3.2. Gpu Loading

The PyTorch network framework for deep learning can specify the computational device. The CPU is used for computation by default, or we can enter the relevant command to load the model into the GPU for computation, thus greatly speeding up the computation. The process of loading the GPU is divided into three parts. First, the device is set as `device = torch.device('cuda : 0' if torch.cuda.is_available() else 'cpu')`. The meaning of the above code is that, if there is a GPU, the calculation is performed on the GPU, and the calculation graphics card is cuda:0, which means that the calculation is performed on the first graphics card; otherwise, the calculation is performed on the CPU. Then, the model is loaded into the GPU: `model.to(device)`, and finally, both the input and output of the training module are loaded into the GPU at computation time. In this paper, the code is represented as a training module: `inputs, target = inputs.to(device), target.to(device)`, or test module: `images, labels = images.to(device), labels.to(device)`. When loading the GPU, it is necessary to ensure that the model, training module, and test module are all loaded on the same card.

We compared the execution time of the deep learning models in the form of “big O” notations, as illustrated in Table 2. “Big O” notation is generally used to describe the complexity of an algorithm, which is the worst-case speedup of the algorithm and is generally represented by $O(n)$, where n denotes the number of operations [34]. The representation of “big O” notation includes $O(1)$, $O(n)$, $O(n^2)$, $O(2n)$, $O(\log n)$, etc., which describes the upper session of asymptotic time complexity when the program runs, as well as the size and performance of the algorithm in the worst-case scenario when the elements in the data structure are increased [35,36].

In this paper, the complexity of the algorithm is calculated in the form of “big O” notation mainly by the analysis of the cyclic nested structure, in which LeNet5, AlexNet, ResNet18, MobileNet are all two-layer cyclic nested structures, and the other parts of the model are computed in parallel, the time complexity of which is $O(n^2)$. Vision Transformer includes three nestings, and its time complexity is $O(n^3)$.

Table 2. The time complexity of each model.

| Model | LeNet5 | AlexNet | ResNet18 | MobileNet | Vision Transformer |
|-----------------|----------|----------|----------|-----------|--------------------|
| Time complexity | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ | $O(n^3)$ |

2.4. Step 3: Comparative Analysis of Deep Learning Models for Pneumonia Recognition

In this paper, a series of comparative analyses of various deep learning models for pneumonia recognition was conducted to filter out the optimal models for pneumonia recognition in various cases and apply them to actual pneumonia recognition. The accuracy of five pneumonia recognition models without data augmentation, LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer, was compared and analyzed, and the recognition efficiency of each pneumonia recognition model was analyzed before and after loading into the GPU. After data augmentation, the accuracies of the LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer pneumonia recognition deep learning models were again compared and analyzed, and the recognition efficiency of each pneumonia recognition model was analyzed before and after loading the GPU. In addition, the accuracy and computational efficiency of the LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models were compared and analyzed before and after data augmentation. The accuracy and computational efficiency of these five classical deep learning models under different situations were compared, which can provide a more in-depth understanding of the advantages and disadvantages of each model under different situations. This is beneficial for the screening of deep learning models for pneumonia recognition in the face of different situations in real-life applications to improve the efficiency of pneumonia recognition.

3. Results

3.1. Experimental Environment

The hardware and software environment configuration used in this paper is illustrated in Table 3.

Table 3. Environment configurations.

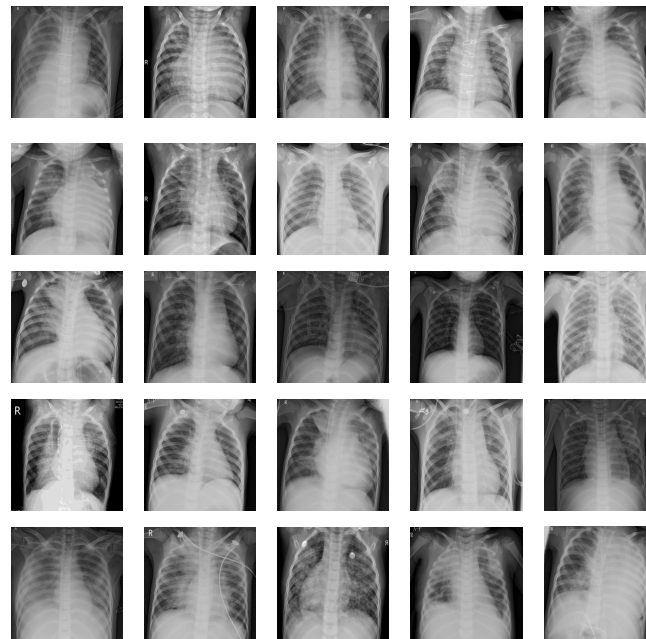
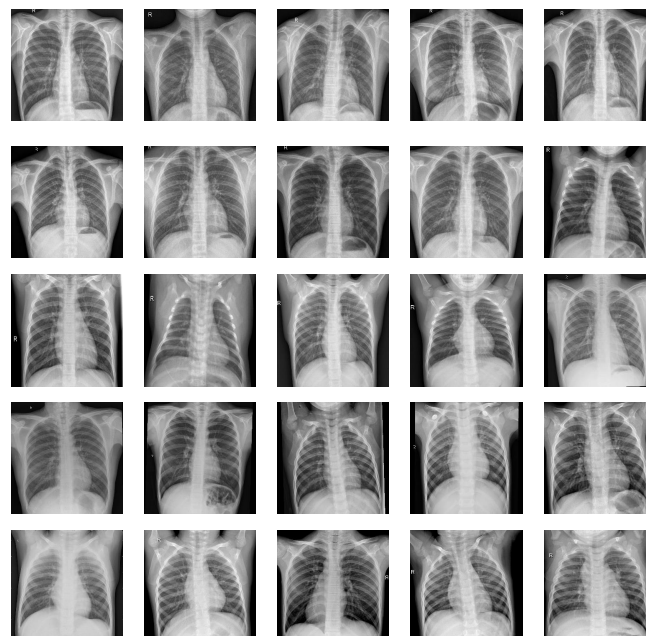
| Environment Configurations | Details |
|----------------------------|-------------------------------|
| OS | Windows 10 Professional |
| Deep learning framework | PyTorch |
| Dependent library | Torch, Torchvision, CUDA etc. |
| CPU | Intel Xeon Gold 5118 CPU |
| CPU RAM (GB) | 128 |
| CPU frequency (GHz) | 2.30 |
| GPU | Quadro P6000 |
| GPU memory (GB) | 24 |

3.2. Details of Experimental Data

The collected dataset was organized and cleaned, and the same number of pneumonia X-ray images and normal X-ray images were taken to ensure the accuracy of the final training and testing results. A total of 3150 X-ray images were obtained for pneumonia identification, of which 1575 were pneumonia images and 1575 were normal images. Eighty percent of images were randomly selected as the training set, and the remaining twenty percent were selected as the testing set. The final training set contained a total of 2520 X-ray images, of which 1260 were pneumonia images and 1260 were normal. The testing set contained a total of 630 X-ray images, of which 315 were pneumonia images and 315 were normal. The details of the datasets are illustrated in Table 4. Before conducting the training test, each image size was adjusted to the same size, the LeNet5 model was adjusted to 32×32 due to the model requirements, and the other models were adjusted to 227×227 , thus facilitating the training and testing. A comparison between the pneumonia image and the normal image datasets is illustrated in Figures 2 and 3, where the lung texture is blurred in the pneumonia X-ray image and the lung texture is clear in the normal X-ray image.

Table 4. Details of the datasets.

| | | | |
|------------------|-------------------|--------------------------------------|-------------------|
| Pneumonia | 1575 X-ray images | Training set (80% randomly selected) | 1260 X-ray images |
| | | Testing set (20% randomly selected) | 315 X-ray images |
| Normal | 1575 X-ray images | Training set (80% randomly selected) | 1260 X-ray images |
| | | Testing set (20% randomly selected) | 315 X-ray images |

**Figure 2.** Some images of the pneumonia dataset.**Figure 3.** Some images of the normal dataset.

3.3. Comparison of the Models before Data Augmentation

3.3.1. Recognition Accuracy

After constructing the pneumonia recognition models, LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models, each model was compared, and the accuracy comparison analysis was performed. Recognition accuracy is the accuracy of each model

for pneumonia identification, expressed as a percentage, obtained by performing tests in the testing set during the testing process. When no data augmentation was performed and no GPU was loaded, the accuracy comparison of each model at 10 epochs is illustrated in Figure 4a. The table is plotted in Table 5, and the histogram of the highest accuracy comparison of each model is illustrated in Figure 4b. From the graphical analysis, it can be seen that before GPU loading and before data augmentation, the LeNet5 model had the highest accuracy, which reached 86.349, the AlexNet model was second, with the highest accuracy reaching 85.238%. The ResNet18 model and MobileNet model also had the highest accuracy close to 80%, but the Vision Transformer model had lower accuracy and poorer pneumonia recognition, with the highest accuracy only reaching 53.333%, which may have been due to the small dataset.

Before data augmentation and after GPU loading, the accuracy comparison of each model at 10 epochs is illustrated in Figure 5a and Table 6, and the comparison graph and the maximum accuracy histogram are illustrated in Figure 5b. After GPU loading, the highest accuracy of each model decreased compared with that before GPU loading, the accuracies of the LeNet5, AlexNet, and MobileNet models decreased, and the accuracies of the ResNet18 and Vision Transformer models increased, which may have been due to the deeper depth of ResNet18 and Vision Transformer. However, the accuracy of the Vision Transformer model was still not high, reaching a maximum of 57.460%.

Table 5. Accuracy of pneumonia recognition by each model before data augmentation and before GPU loading.

| Epoch | LeNet5 (%) | AlexNet (%) | ResNet18 (%) | MobileNet (%) | Vision Transformer (%) |
|-------|------------|-------------|--------------|---------------|------------------------|
| 1 | 68.730 | 54.921 | 54.444 | 50.159 | 48.889 |
| 2 | 82.857 | 65.873 | 64.286 | 56.508 | 49.841 |
| 3 | 78.254 | 53.968 | 71.587 | 68.571 | 49.365 |
| 4 | 77.460 | 85.238 | 70.159 | 69.683 | 46.349 |
| 5 | 73.968 | 85.079 | 73.333 | 67.619 | 49.206 |
| 6 | 72.857 | 78.889 | 73.968 | 76.349 | 50.159 |
| 7 | 70.317 | 73.968 | 73.175 | 72.063 | 53.333 |
| 8 | 86.349 | 74.762 | 77.302 | 70.476 | 53.175 |
| 9 | 80.159 | 77.937 | 64.127 | 73.016 | 51.587 |
| 10 | 80.159 | 72.540 | 68.571 | 69.365 | 49.365 |

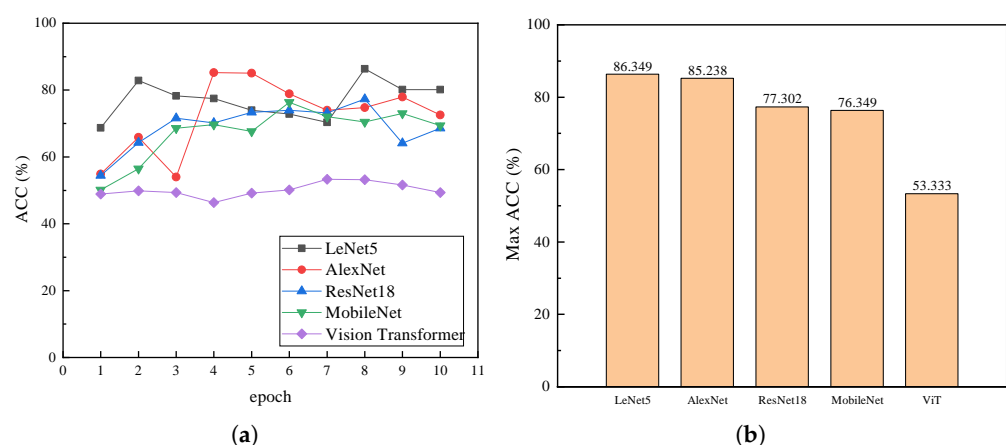
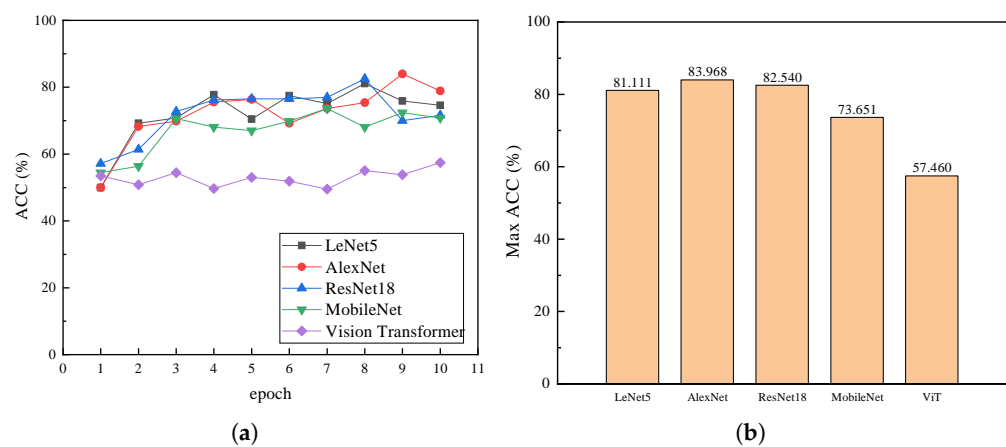


Figure 4. Comparison of accuracy before data augmentation and before GPU loading. (a) Comparison of accuracy of each model before data augmentation and before GPU loading. (b) Histogram comparing the highest accuracy of each model before data augmentation and before GPU loading.

Table 6. Accuracy of pneumonia recognition by each model before data augmentation and after GPU loading.

| Epoch | LeNet5 (%) | AlexNet (%) | ResNet18 (%) | MobileNet (%) | Vision Transformer (%) |
|-------|------------|-------------|--------------|---------------|------------------------|
| 1 | 50.000 | 50.000 | 57.143 | 54.444 | 53.492 |
| 2 | 69.206 | 68.254 | 61.429 | 56.349 | 50.794 |
| 3 | 70.794 | 69.841 | 72.698 | 70.635 | 54.444 |
| 4 | 77.778 | 75.556 | 76.190 | 68.095 | 49.683 |
| 5 | 70.476 | 76.349 | 76.508 | 66.984 | 53.016 |
| 6 | 77.460 | 69.206 | 76.508 | 69.841 | 51.905 |
| 7 | 75.079 | 73.651 | 76.984 | 73.651 | 49.524 |
| 8 | 81.111 | 75.397 | 82.540 | 68.095 | 55.079 |
| 9 | 75.873 | 83.968 | 70.000 | 72.381 | 53.810 |
| 10 | 74.603 | 78.889 | 71.587 | 70.794 | 57.460 |

**Figure 5.** Comparison of accuracy before data augmentation and after GPU loading. (a) Comparison of accuracy of each model before data augmentation and after GPU loading. (b) Histogram comparing the highest accuracy of each model before data augmentation and after GPU loading.

3.3.2. Computational Efficiency

Computational efficiency is the time consumed by each model for training and testing per epoch round, expressed in minutes. The lower the training and testing time, the higher the computational efficiency, and the longer the training and testing time, the lower the computational efficiency. Before data augmentation and GPU loading, the computational efficiency of each model at 10 epochs is illustrated in Figure 6a, and the table is illustrated in Table 7. Before GPU loading, the LeNet5 model took only 1.5 min to complete each epoch round because of its simple model and a small number of layers. The AlexNet model and MobileNet model also had shorter computation times per epoch; the former was due to the model structure being still relatively simple and the number of layers being shallow, while the MobileNet model pursues computational efficiency due to the use of the depth-separable convolutional module, so the computational time was shorter and the computational efficiency was higher. The ResNet18 model, in contrast, had a relatively long computation time due to the deeper model depth, while the Vision Transformer model had the longest computation time, which was because the Vision Transformer model is less used in image recognition, so the computational efficiency is lower and still needs some improvement and enhancement.

After GPU loading, the computational efficiency of each model at 10 epochs is illustrated in Figure 6b, and the table is illustrated in Table 8. A comparison of the efficiency of each model before and after GPU loading is illustrated in Figure 7. The computational efficiency of each model greatly improved. The LeNet5 model was not significantly improved because of its simplicity, but it improved from approximately 1.5 min to approximately 1.1 min. The AlexNet model improved from approximately 6 min to approximately 2 min

compared to the ResNet18 and MobileNet models, which improved to approximately 1 min. The AlexNet model has a simple and rough structure, so the improvement in computational efficiency was less compared to the more refined ResNet18 and MobileNet models. The Vision Transformer model also improved the computational time from more than 50 min per epoch to only 17 min per epoch.

Table 7. Computational efficiency of per epoch operation for each model before data augmentation and before GPU loading.

| Epoch | LeNet5 (min) | AlexNet (min) | ResNet18 (min) | MobileNet (min) | Vision Transformer (min) |
|-------|--------------|---------------|----------------|-----------------|--------------------------|
| 1 | 1.51 | 6.24 | 12.50 | 7.12 | 55.04 |
| 2 | 1.55 | 6.27 | 12.52 | 7.14 | 55.07 |
| 3 | 1.53 | 6.20 | 12.52 | 7.16 | 55.45 |
| 4 | 1.52 | 6.20 | 12.49 | 7.16 | 55.31 |
| 5 | 1.53 | 6.22 | 12.48 | 7.20 | 55.53 |
| 6 | 1.53 | 6.17 | 12.53 | 7.18 | 55.57 |
| 7 | 1.53 | 6.25 | 12.50 | 7.18 | 55.61 |
| 8 | 1.51 | 6.21 | 12.51 | 7.18 | 55.64 |
| 9 | 1.54 | 6.21 | 12.51 | 7.16 | 55.40 |
| 10 | 1.52 | 6.22 | 12.52 | 7.30 | 55.55 |

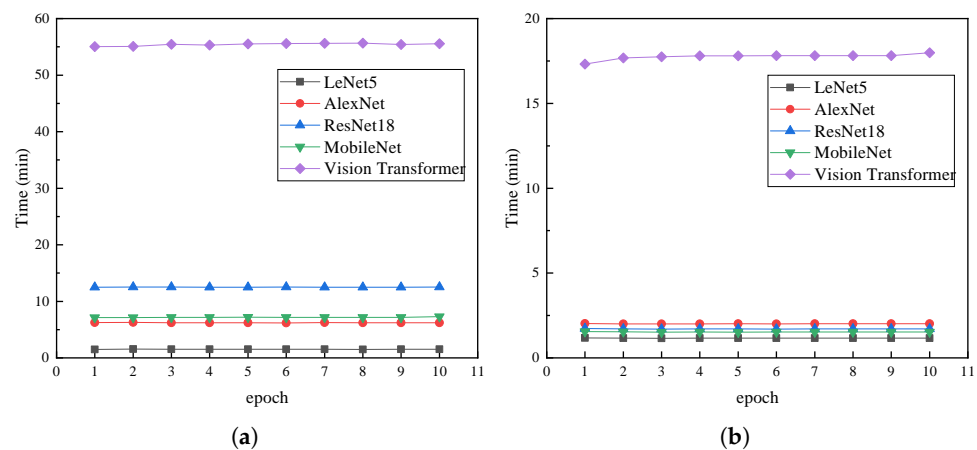


Figure 6. Comparison of the efficiency before data augmentation. (a) Comparison of the efficiency of each model before data augmentation and before GPU loading. (b) Comparison of the efficiency of each model before data augmentation and after GPU loading.

Table 8. Computational efficiency of per epoch operation for each model before data augmentation and after GPU loading.

| Epoch | LeNet5 (min) | AlexNet (min) | ResNet18 (min) | MobileNet (min) | Vision Transformer (min) |
|-------|--------------|---------------|----------------|-----------------|--------------------------|
| 1 | 1.18 | 2.03 | 1.73 | 1.55 | 17.32 |
| 2 | 1.17 | 2.00 | 1.71 | 1.54 | 17.68 |
| 3 | 1.16 | 2.00 | 1.70 | 1.52 | 17.75 |
| 4 | 1.17 | 2.00 | 1.71 | 1.53 | 17.80 |
| 5 | 1.17 | 2.01 | 1.71 | 1.52 | 17.80 |
| 6 | 1.17 | 2.00 | 1.70 | 1.53 | 17.81 |
| 7 | 1.17 | 2.02 | 1.71 | 1.53 | 17.81 |
| 8 | 1.17 | 2.01 | 1.71 | 1.53 | 17.82 |
| 9 | 1.17 | 2.01 | 1.71 | 1.53 | 17.82 |
| 10 | 1.17 | 2.02 | 1.71 | 1.53 | 17.99 |

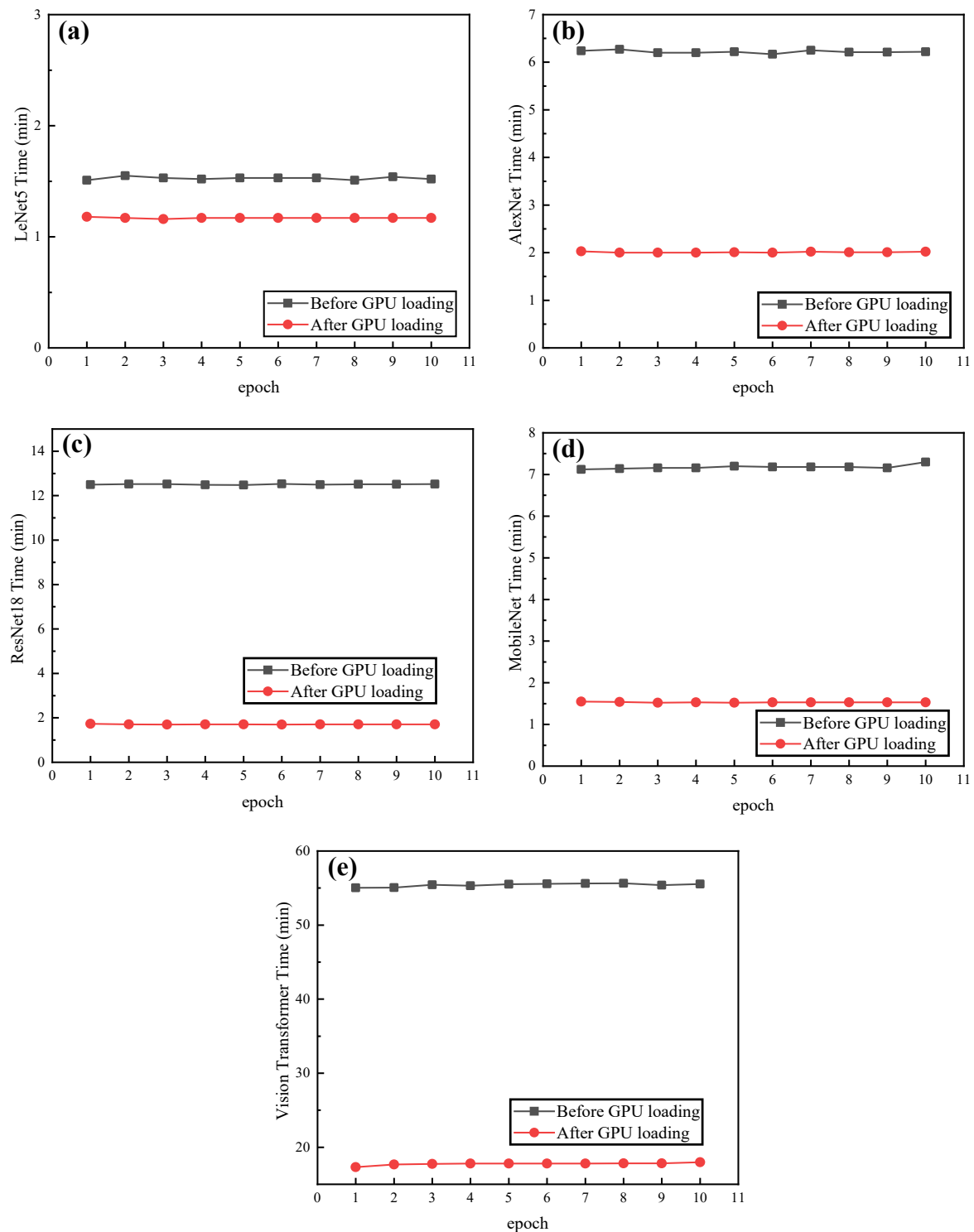


Figure 7. Comparison of the efficiency of each model before data augmentation before and after GPU loading. (a) Comparison of the efficiency of LeNet5 model. (b) Comparison of the efficiency of AlexNet model. (c) Comparison of the efficiency of ResNet18 model. (d) Comparison of the efficiency of MobileNet model. (e) Comparison of the efficiency of Vision Transformer model.

3.4. Comparison of the Models after Data Augmentation

3.4.1. Recognition Accuracy

When data augmentation was performed and the GPU was not loaded, the accuracy comparison of each model under 10 epochs is illustrated in Figure 8a, the table is plotted in

Table 9, and the histogram of the highest accuracy comparison of each model is illustrated in Figure 8b. As seen in the graphical analysis, after data augmentation and before GPU loading, the AlexNet model had the highest accuracy, with the highest accuracy reaching 83.968%; the LeNet5 model was second, with the highest accuracy reaching 82.857%; the ResNet18 model and MobileNet model had the highest accuracies close to 80%, and the Vision Transformer model, although still less accurate with poor pneumonia recognition, was the only model with improved accuracy after data augmentation compared to the other models, with a maximum accuracy of 57.143%. The lack of improvement in accuracy for the other models may have been due to the smaller dataset, and the models not being sensitive to data enhancement after data augmentation. The Vision Transformer model is more sensitive to changes in data augmentation for smaller datasets, so the model accuracy can be improved by expanding the dataset.

After the data augmentation and GPU loading, the accuracy of each model at 10 epochs is illustrated in Figure 9a and Table 10, and the comparison graph and the maximum accuracy histogram are illustrated in Figure 9b. After GPU loading, the maximum accuracy of each model decreased compared with that before GPU loading, except for the MobileNet model. The data augmentation operation did not improve the accuracy of the models in this paper, and the GPU loading operation also led to a decrease in the accuracy of the models, so data augmentation was not significant in improving the accuracy of the models for datasets that are too small.

Table 9. Accuracy of pneumonia identification by each model after data augmentation and before GPU loading.

| Epoch | LeNet5 (%) | AlexNet (%) | ResNet18 (%) | MobileNet (%) | Vision Transformer (%) |
|-------|------------|-------------|--------------|---------------|------------------------|
| 1 | 50.000 | 50.000 | 51.111 | 50.000 | 51.905 |
| 2 | 73.492 | 51.111 | 50.159 | 53.016 | 53.492 |
| 3 | 80.794 | 50.000 | 54.444 | 60.159 | 51.111 |
| 4 | 81.270 | 66.825 | 52.222 | 62.063 | 54.444 |
| 5 | 82.857 | 60.635 | 54.921 | 63.333 | 52.857 |
| 6 | 66.190 | 82.222 | 62.857 | 66.984 | 50.476 |
| 7 | 75.556 | 81.111 | 64.603 | 65.079 | 49.206 |
| 8 | 80.317 | 79.206 | 67.619 | 71.429 | 52.381 |
| 9 | 78.571 | 83.968 | 72.381 | 72.857 | 53.651 |
| 10 | 75.079 | 83.333 | 75.714 | 76.190 | 57.143 |

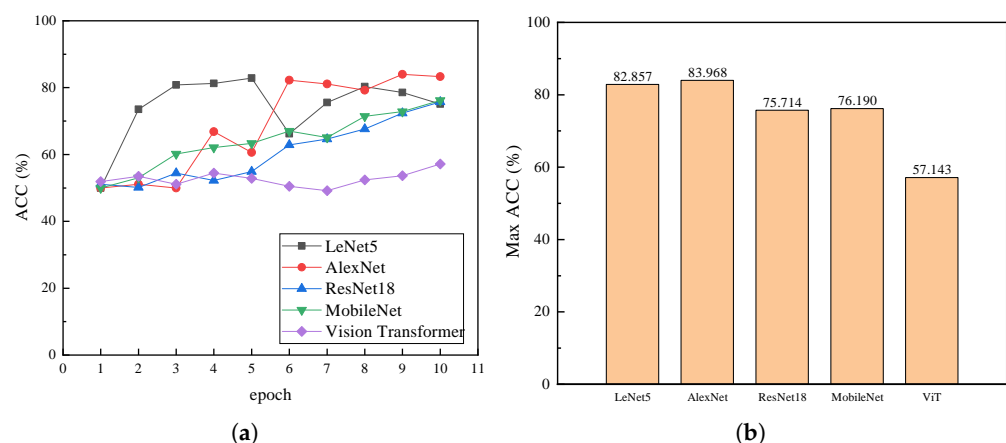
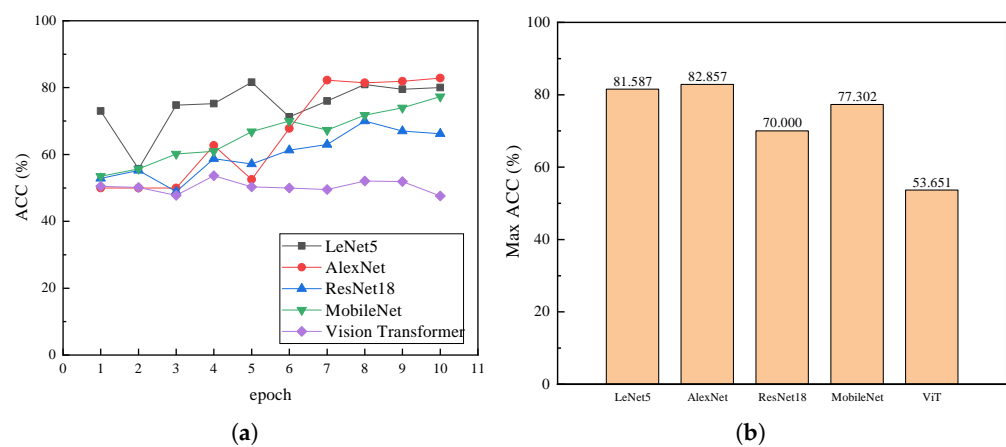


Figure 8. Comparison of accuracy after data augmentation and before GPU loading. (a) Comparison of accuracy of each model after data augmentation and before GPU loading. (b) Histogram comparing the highest accuracy of each model after data augmentation and before GPU loading.

Table 10. Accuracy of pneumonia identification by each model after data augmentation and after GPU loading.

| Epoch | LeNet5 (%) | AlexNet (%) | ResNet18 (%) | MobileNet (%) | Vision Transformer (%) |
|-------|------------|-------------|--------------|---------------|------------------------|
| 1 | 73.016 | 50.000 | 52.857 | 53.492 | 50.476 |
| 2 | 55.714 | 50.000 | 55.238 | 55.714 | 50.159 |
| 3 | 74.762 | 50.000 | 48.889 | 60.159 | 47.778 |
| 4 | 75.238 | 62.698 | 58.730 | 60.952 | 53.651 |
| 5 | 81.587 | 52.540 | 57.143 | 66.825 | 50.317 |
| 6 | 71.270 | 67.778 | 61.270 | 70.000 | 50.000 |
| 7 | 76.032 | 82.222 | 63.016 | 67.302 | 49.524 |
| 8 | 80.952 | 81.429 | 70.000 | 71.746 | 52.063 |
| 9 | 79.524 | 81.905 | 66.984 | 73.968 | 51.905 |
| 10 | 80.000 | 82.857 | 66.190 | 77.302 | 47.619 |

**Figure 9.** Comparison of accuracy after data augmentation and after GPU loading. (a) Comparison of accuracy of each model after data augmentation and after GPU loading. (b) Histogram comparing the highest accuracy of each model after data augmentation and after GPU loading.

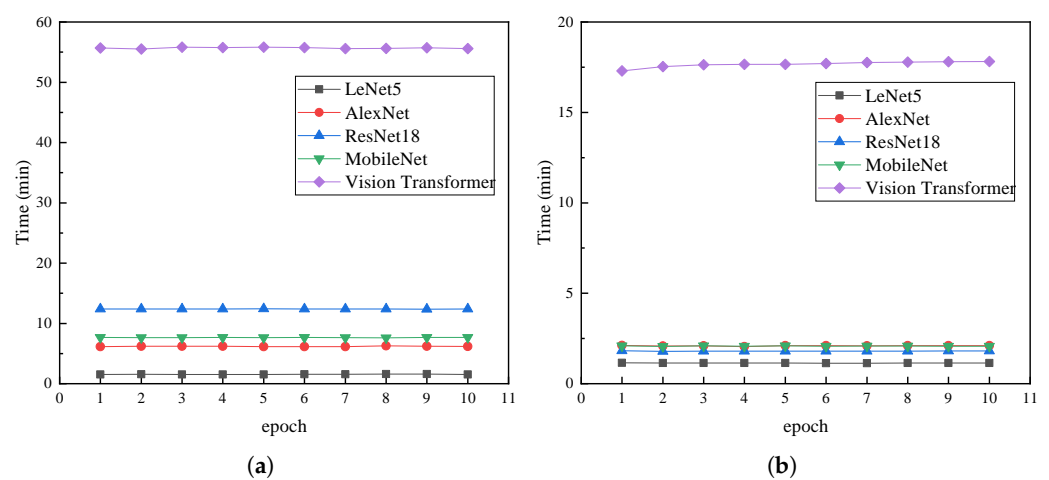
3.4.2. Computational Efficiency

When the data augmentation was performed without the use of GPU loading, the comparison of the computational efficiency of each model at 10 epochs before GPU loading is illustrated in Figure 10a, and the table is plotted in Table 11. The models before GPU loading showed a similar pattern as before the data augmentation, with the LeNet5 model having the highest computational efficiency, followed by the AlexNet model, then the MobileNet model, the ResNet18 model, and the Vision Transformer models.

After GPU loading, the computational efficiency of each model greatly improved, as illustrated in Table 12 and Figure 10b. The comparison of the efficiency of each model before and after GPU loading is illustrated in Figure 11. The changing pattern was still similar to that before data augmentation; the LeNet5 model was not significantly improved because of its simplicity, but it was also improved from approximately 1.5 min to approximately 1.1 min. The AlexNet and MobileNet models improved from approximately 6 min to approximately 2 min, while the MobileNet model was less efficient than before the data augmentation but still more efficient than the AlexNet model; the ResNet18 model still improved to more than 1 min, which was a large improvement in efficiency. The Vision Transformer model also improved from more than 50 min per epoch to just over 17 min per epoch.

Table 11. Computational efficiency of per epoch operation for each model after data augmentation and before GPU loading.

| Epoch | LeNet5 (min) | AlexNet (min) | ResNet18 (min) | MobileNet (min) | Vision Transformer (min) |
|-------|--------------|---------------|----------------|-----------------|--------------------------|
| 1 | 1.54 | 6.15 | 12.41 | 7.66 | 55.69 |
| 2 | 1.56 | 6.22 | 12.39 | 7.63 | 55.52 |
| 3 | 1.54 | 6.20 | 12.38 | 7.64 | 55.84 |
| 4 | 1.54 | 6.22 | 12.40 | 7.66 | 55.74 |
| 5 | 1.54 | 6.15 | 12.43 | 7.65 | 55.82 |
| 6 | 1.55 | 6.14 | 12.39 | 7.66 | 55.76 |
| 7 | 1.55 | 6.14 | 12.39 | 7.65 | 55.58 |
| 8 | 1.58 | 6.27 | 12.38 | 7.62 | 55.61 |
| 9 | 1.58 | 6.23 | 12.36 | 7.67 | 55.73 |
| 10 | 1.53 | 6.19 | 12.40 | 7.67 | 55.59 |

**Figure 10.** Comparison of the efficiency after data augmentation. (a) Comparison of the efficiency of each model after data augmentation and before GPU loading. (b) Comparison of the efficiency of each model after data augmentation and after GPU loading.**Table 12.** Computational efficiency of per epoch operation for each model after data augmentation and after GPU loading.

| Epoch | LeNet5 (min) | AlexNet (min) | ResNet18 (min) | MobileNet (min) | Vision Transformer (min) |
|-------|--------------|---------------|----------------|-----------------|--------------------------|
| 1 | 1.15 | 2.12 | 1.82 | 2.09 | 17.29 |
| 2 | 1.14 | 2.08 | 1.79 | 2.06 | 17.53 |
| 3 | 1.14 | 2.09 | 1.80 | 2.09 | 17.64 |
| 4 | 1.14 | 2.05 | 1.80 | 2.07 | 17.66 |
| 5 | 1.14 | 2.10 | 1.80 | 2.09 | 17.66 |
| 6 | 1.13 | 2.10 | 1.80 | 2.07 | 17.70 |
| 7 | 1.13 | 2.09 | 1.80 | 2.08 | 17.76 |
| 8 | 1.14 | 2.10 | 1.80 | 2.08 | 17.78 |
| 9 | 1.14 | 2.10 | 1.81 | 2.07 | 17.80 |
| 10 | 1.14 | 2.11 | 1.81 | 2.07 | 17.81 |

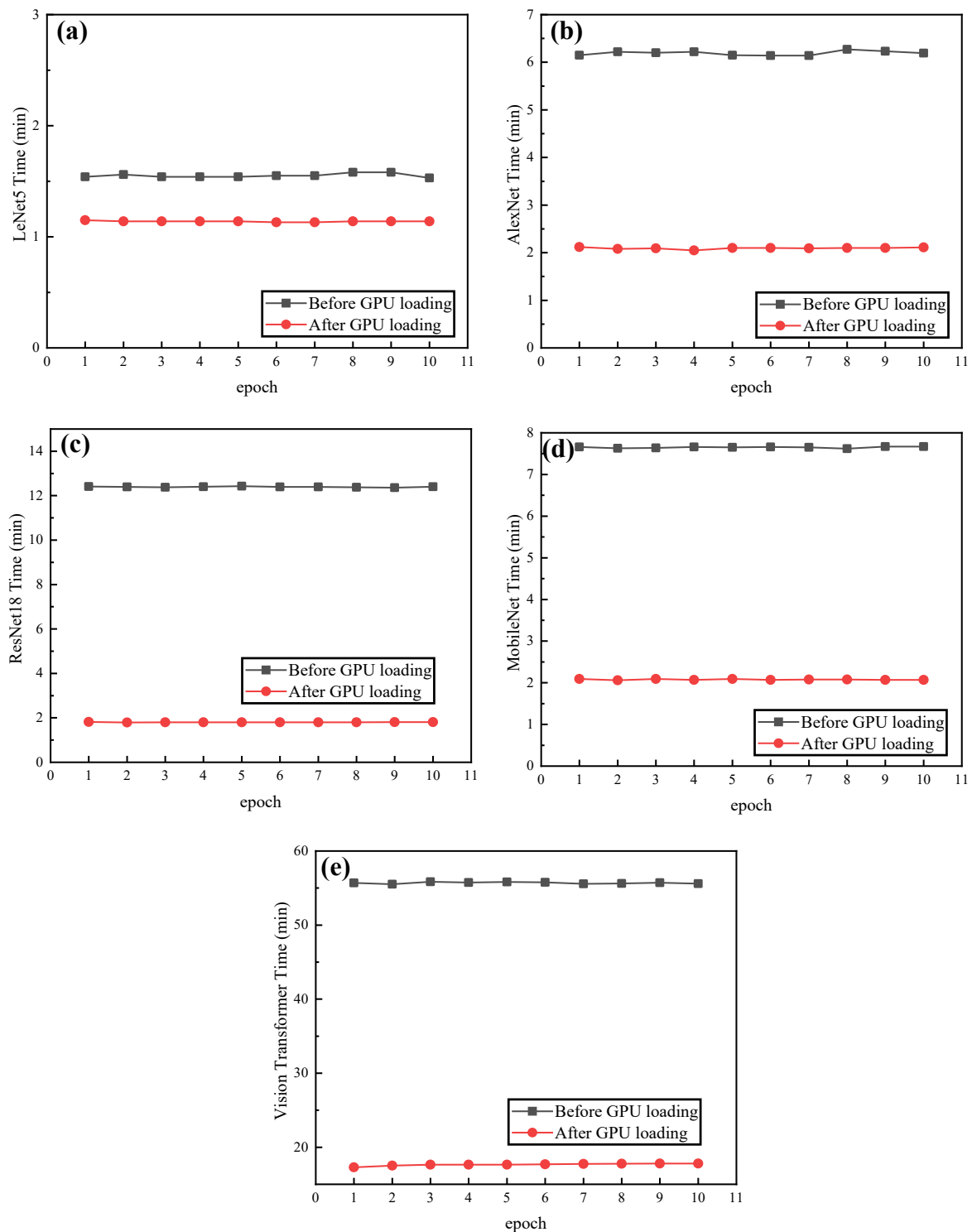


Figure 11. Comparison of the efficiency of each model after data augmentation before and after GPU loading. (a) Comparison of the efficiency of LeNet5 model. (b) Comparison of the efficiency of AlexNet model. (c) Comparison of the efficiency of ResNet18 model. (d) Comparison of the efficiency of MobileNet model. (e) Comparison of the efficiency of Vision Transformer model.

3.5. Comparison of Each Model before and after Data Augmentation

3.5.1. Recognition Accuracy

As seen in the Figure 12 and 13, the accuracy of each model did not improve greatly after the data augmentation and even decreased, which was caused by the fact that the

dataset in this study was too small. So, data augmentation should not be employed in datasets that are too small, as it will reduce the accuracy of the model.

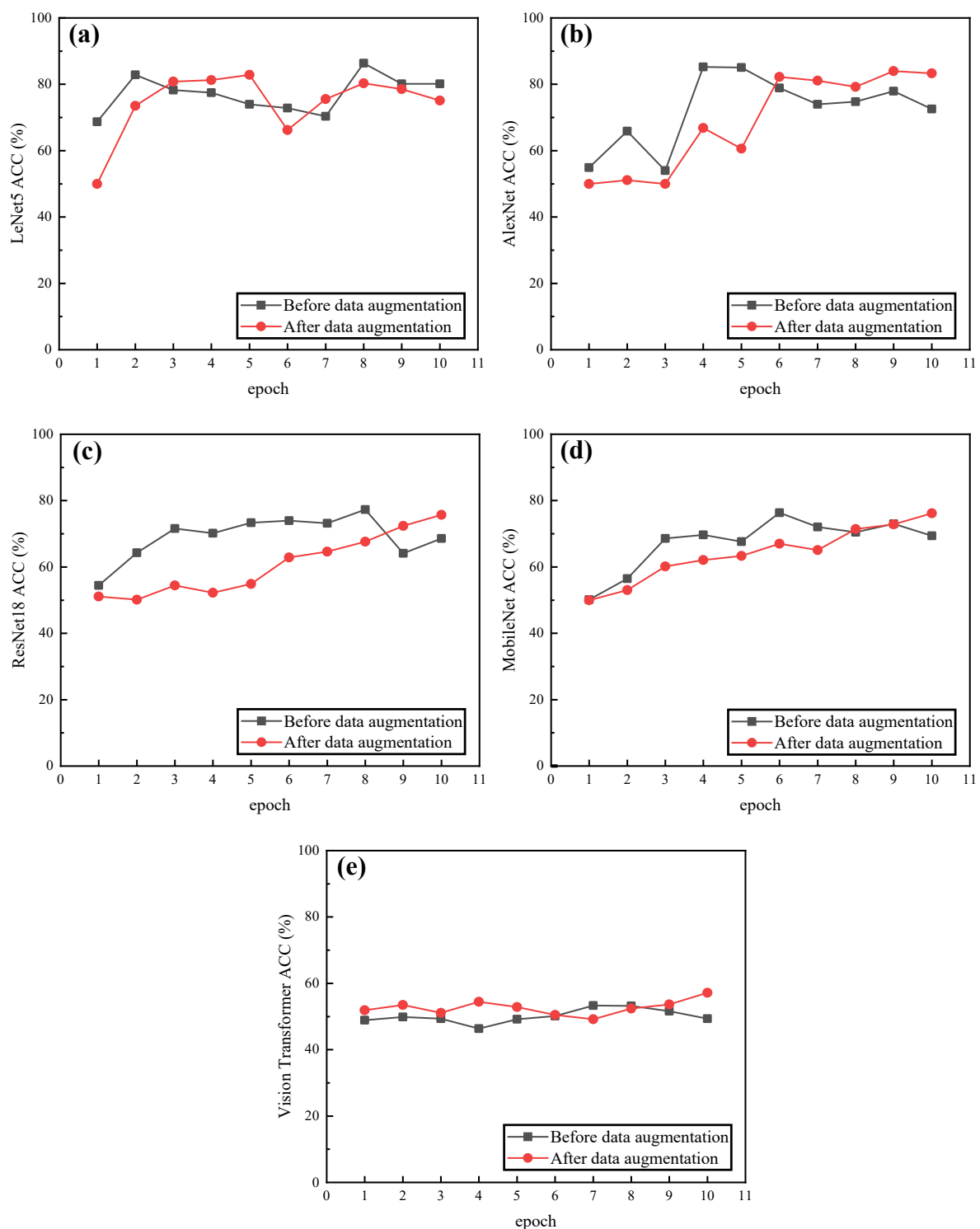


Figure 12. Comparison of accuracy of each model before GPU loading before and after data augmentation. (a) Comparison of the accuracy of LeNet5 model. (b) Comparison of the accuracy of AlexNet model. (c) Comparison of the accuracy of ResNet18 model. (d) Comparison of the accuracy of MobileNet model. (e) Comparison of the accuracy of Vision Transformer model.

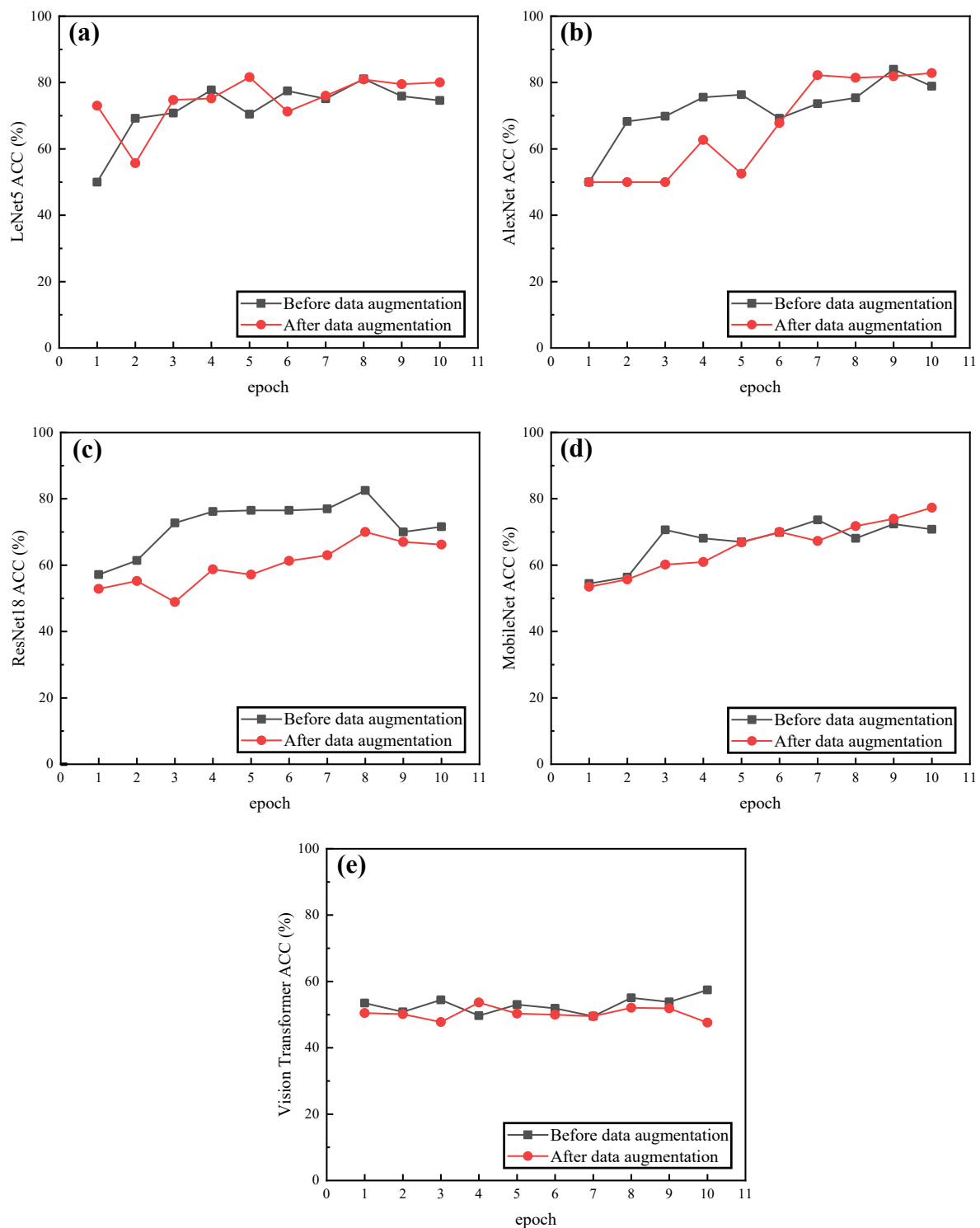


Figure 13. Comparison of accuracy of each model after GPU loading before and after data augmentation. (a) Comparison of the accuracy of LeNet5 model. (b) Comparison of the accuracy of AlexNet model. (c) Comparison of the accuracy of ResNet18 model. (d) Comparison of the accuracy of MobileNet model. (e) Comparison of the accuracy of Vision Transformer model.

3.5.2. Computational Efficiency

As seen in the Figure 14 and 15, the computational efficiency of each model did not decrease significantly after data augmentation, which encourages the use of a data augmentation operation when the dataset is large to improve the accuracy without affecting

the computational efficiency. However, in this paper, it is also possible that the dataset was too small, so the data augmentation had little effect on the computational efficiency.

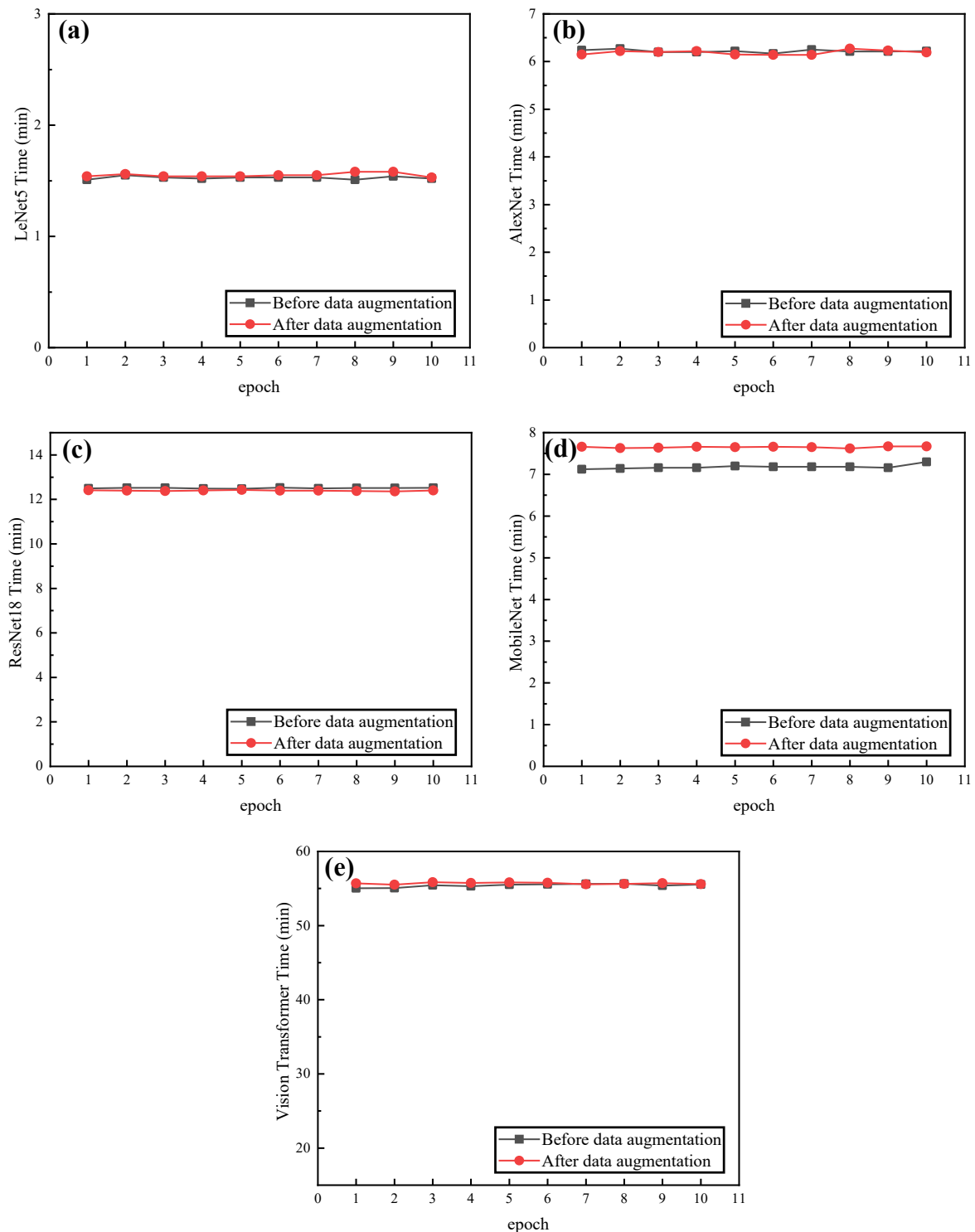


Figure 14. Comparison of the computational efficiency of each model before GPU loading before and after data augmentation. (a) Comparison of the computational efficiency of LeNet5 model. (b) Comparison of the computational efficiency of AlexNet model. (c) Comparison of the computational efficiency of ResNet18 model. (d) Comparison of the computational efficiency of MobileNet model. (e) Comparison of the computational efficiency of Vision Transformer model.

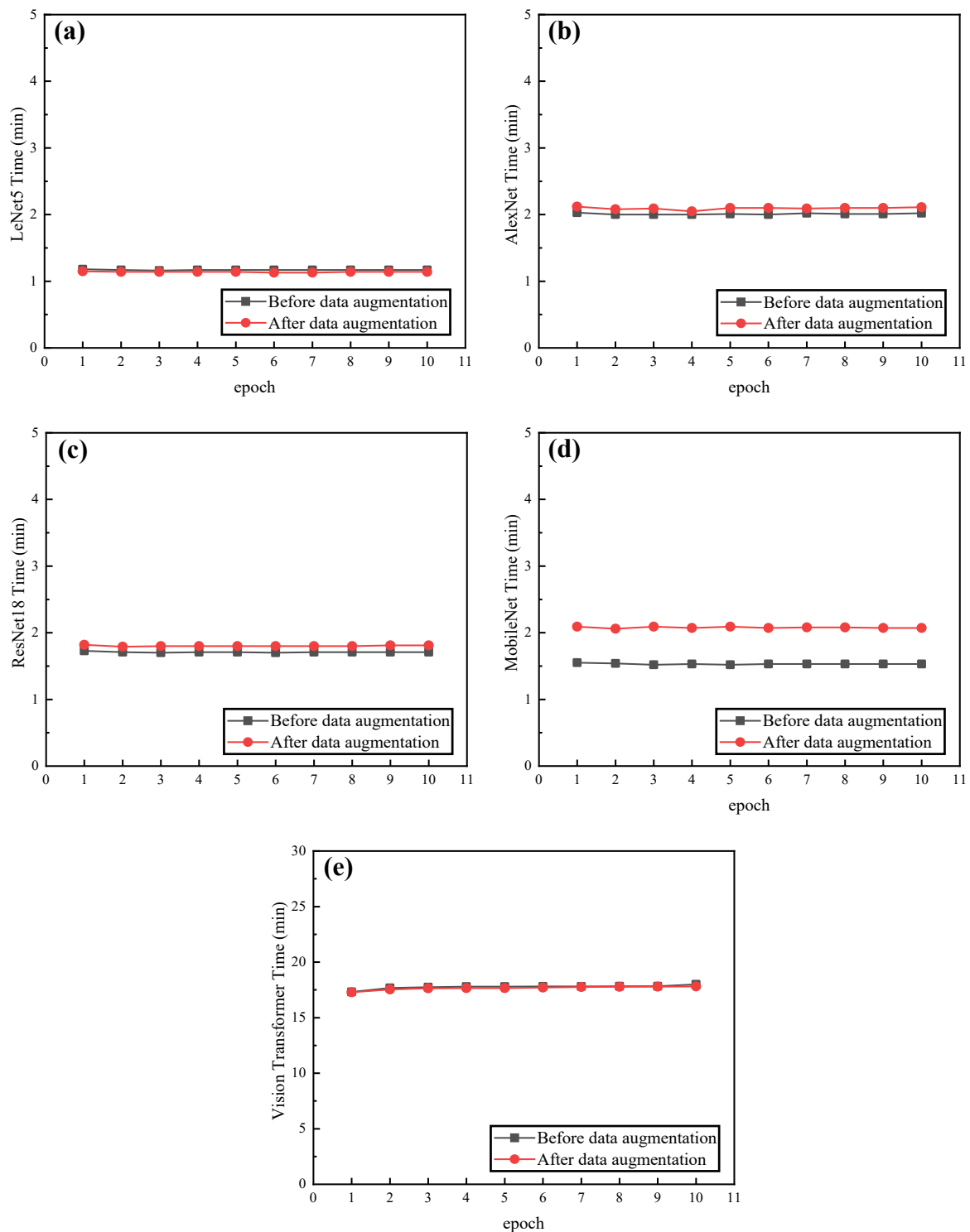


Figure 15. Comparison of the computational efficiency of each model after GPU loading before and after data augmentation. (a) Comparison of the computational efficiency of LeNet5 model. (b) Comparison of the computational efficiency of AlexNet model. (c) Comparison of the computational efficiency of ResNet18 model. (d) Comparison of the computational efficiency of MobileNet model. (e) Comparison of the computational efficiency of Vision Transformer model.

4. Discussion

In this paper, five deep learning models, including the LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models, were established to compare and analyze

them for pneumonia recognition and to filter out the optimal models for pneumonia recognition in various situations. By comparing these five classical deep learning models for pneumonia recognition, it was possible to achieve a deeper understanding of the strengths and weaknesses of each model, enabling rapid selection of the most appropriate model and improvement in the efficiency of pneumonia recognition for practical applications.

However, the dataset in this paper was small, and the effect of expanding the dataset by the data augmentation method was not satisfactory. The accuracy of each model could not be further improved, and the comparative analysis could only be performed for pneumonia recognition in small datasets. Therefore, the final results of the comparative analysis have some limitations and need to be further investigated. The collection of the X-ray images of pneumonia was subject to problems, such as difficulties in data acquisition, patient privacy, and the doctor-patient relationship.

Therefore, in the future, we hope to further expand the dataset by generating new images, using methods such as generative adversarial networks (GAN). The large-scale pneumonia image data will be employed for the recognition of pneumonia X-ray images in deep learning to achieve a comparative analysis of each model in a rich dataset and to make the pneumonia recognition more accurate. In addition, we will perform a comparative analysis of pneumonia recognition using more deep learning models applicable to large datasets, perform a more detailed analytical evaluation of the models, including an analytical study of precision, recall, f1-scores, etc., and investigate deep learning methods specifically for pneumonia recognition, which will lead to better applications in practice.

5. Conclusions

In this paper, we established five deep learning models for pneumonia recognition, including LeNet5, AlexNet, MobileNet, ResNet18, and Vision Transformer models, and compared and analyzed these five deep learning models for pneumonia recognition to identify the optimal models for pneumonia recognition in various situations. The essential idea was to employ five deep learning models to identify pneumonia X-ray images and identify the optimal models so that they can be used in actual pneumonia recognition to improve the pneumonia recognition efficiency and applied further to the computer-aided diagnosis of pneumonia species. The comparison lays the foundation for the practical application of deep learning in pneumonia recognition to maintain people's health and lives.

The main conclusions of this paper are that: (1) For the smaller datasets applied in this paper, both the LeNet5 model and the AlexNet model can achieve better recognition results; (2) For datasets that are too small, data augmentation cannot improve the accuracy of the pneumonia recognition model, which was caused by the small datasets used and the insensitivity of the model to changes in the dataset. (3) The computational efficiency of each model can be greatly improved after GPU loading. Regardless of before and after data augmentation, the computational efficiency of the models greatly improved after GPU loading.

For future research, it is necessary to collect richer datasets and expand the existing datasets to improve the accuracy of the pneumonia recognition models and further filter out deep learning models for pneumonia recognition that are more suitable for real-life applications to improve pneumonia recognition efficiency and maintain people's health and lives.

Author Contributions: Conceptualization, Y.Y. and G.M.; methodology, Y.Y. and G.M.; writing—original draft preparation, Y.Y. and G.M.; writing—review and editing, Y.Y. and G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was jointly supported by the 2021 Graduate Innovation Fund Project of the China University of Geosciences, Beijing (ZD2021YC009), and the Major Program of Science and Technology of Xinjiang Production and Construction Corps (2020AA002). The authors would like to thank the editors and the reviewers for their valuable comments.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The availability of the obtained datasets is as follows: Dataset: Chest X-ray Images for Classification. URL: <https://data.mendeley.com/datasets/rsbjbr9sj/2> (accessed on 1 February 2022).

Acknowledgments: The authors would like to thank the editor and the reviewers for their helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pereda, M.A.; Chavez, M.A.; Hooper-Miele, C.C.; Gilman, R.H.; Steinhoff, M.C.; Ellington, L.E.; Gross, M.; Price, C.; Tielsch, J.M.; Checkley, W. Lung Ultrasound for the Diagnosis of Pneumonia in Children: A Meta-analysis. *Pediatrics* **2015**, *135*, 714–722. [CrossRef]
2. Ruuskanen, O.; Lahti, E.; Jennings, L.C.; Murdoch, D.R. Viral pneumonia. *Lancet* **2011**, *377*, 1264–1275. [CrossRef]
3. Kalil, A.C.; Metersky, M.L.; Klompas, M.; Muscedere, J.; Sweeney, D.A.; Palmer, L.B.; Napolitano, L.M.; O’Grady, N.P.; Bartlett, J.G.; Carratala, J.; et al. Management of Adults With Hospital-acquired and Ventilator-associated Pneumonia: 2016 Clinical Practice Guidelines by the Infectious Diseases Society of America and the American Thoracic Society. *Clin. Infect. Dis.* **2016**, *63*, E61–E111. [CrossRef] [PubMed]
4. Dai, W.C.; Zhang, H.W.; Yu, J.; Xu, H.J.; Chen, H.; Luo, S.P.; Zhang, H.; Liang, L.H.; Wu, X.L.; Lei, Y.; et al. CT Imaging and Differential Diagnosis of COVID-19. *Can. Assoc. Radiol. J.* **2020**, *71*, 195–200. [CrossRef]
5. Apostolopoulos, I.D.; Mpesiana, T.A. COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys. Eng. Sci. Med.* **2020**, *43*, 635–640. [CrossRef] [PubMed]
6. Rahman, T.; Chowdhury, M.E.H.; 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]
7. Heckerling, P.S.; Tape, T.G.; Wigton, R.S. Relation of physicians predicted probabilities of pneumonia to their utilities for ordering chest X-rays to detect pneumonia. *Med. Decis. Mak.* **1992**, *12*, 32–38. [CrossRef] [PubMed]
8. Jaiswal, A.K.; Tiwari, P.; Kumar, S.; Gupta, D.; Khanna, A.; Rodrigues, J.J.P.C. Identifying pneumonia in chest X-rays: A deep learning approach. *Measurement* **2019**, *145*, 511–518. [CrossRef]
9. Mahmud, T.; Rahman, M.A.; Fattah, S.A. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Comput. Biol. Med.* **2020**, *122*, 103869. [CrossRef] [PubMed]
10. Zhao, W.D.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453.
11. Cui, H.; Guan, Y.; Chen, H. Rolling Element Fault Diagnosis Based on VMD and Sensitivity MCKD. *IEEE Access* **2021**, *9*, 120297–120308. [CrossRef]
12. Wu, E.Q.; Zhou, M.; Hu, D.; Zhu, L.; Tang, Z.; Qiu, X.Y.; Deng, P.Y.; Zhu, L.M.; Ren, H. Self-Paced Dynamic Infinite Mixture Model for Fatigue Evaluation of Pilots’ Brains. *IEEE Trans. Cybern.* **2020**, *50*, 1–16. [CrossRef]
13. Chowdhury, M.E.H.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Bin Mahbub, Z.; 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]
14. Stephen, O.; Sain, M.; Maduh, U.J.; Jeong, D.U. An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare. *J. Healthc. Eng.* **2019**, *2019*, 4180949. [CrossRef] [PubMed]
15. Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damasevicius, R.; de Albuquerque, V.H.C. A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images. *Appl. Sci.* **2020**, *10*, 559. [CrossRef]
16. Salehinejad, H.; Valaee, S.; Dowdell, T.; Colak, E.; Barfett, J. Generalization of deep neural networks for chest pathology classification in X-rays using generative adversarial networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 990–994.
17. Gazda, M.; Plavka, J.; Gazda, J.; Drotar, P. Self-Supervised Deep Convolutional Neural Network for Chest X-ray Classification. *IEEE Access* **2021**, *9*, 151972–151982. [CrossRef]
18. Panthakkan, A.; Anzar, S.M.; Al Mansoori, S.; Al Ahmad, H. A novel DeepNet model for the efficient detection of COVID-19 for symptomatic patients. *Biomed. Signal Process. Control* **2021**, *68*, 102812. [CrossRef] [PubMed]
19. Wang, S.H.; Satapathy, S.C.; Anderson, D.; Chen, S.X.; Zhang, Y.D. Deep Fractional Max Pooling Neural Network for COVID-19 Recognition. *Front. Public Health* **2021**, *9*, 726144. [CrossRef]
20. Alhudhaif, A.; Polat, K.; Karaman, O. Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. *Expert Syst. Appl.* **2021**, *180*, 115141. [CrossRef] [PubMed]
21. Tahir, A.M.; Qiblawey, Y.; Khandakar, A.; Rahman, T.; Khurshid, U.; Musharavati, F.; Islam, M.T.; Kiranyaz, S.; Al-Maadeed, S.; Chowdhury, M.E.H. Deep Learning for Reliable Classification of COVID-19, MERS, and SARS from Chest X-ray Images. *Cogn. Comput.* **2022**, *1*–21. [CrossRef] [PubMed]

22. Loey, M.; El-Sappagh, S.; Mirjalili, S. Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data. *Comput. Biol. Med.* **2022**, *142*, 105213. [[CrossRef](#)] [[PubMed](#)]
23. Sitaula, C.; Hossain, M.B. Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Appl. Intell.* **2021**, *51*, 2850–2863. [[CrossRef](#)] [[PubMed](#)]
24. Sitaula, C.; Aryal, S. New bag of deep visual words based features to classify chest X-ray images for COVID-19 diagnosis. *Health Inf. Sci. Syst.* **2021**, *9*, 24. [[CrossRef](#)]
25. Sitaula, C.; Shahi, T.B.; Aryal, S.; Marzbanrad, F. Fusion of multi-scale bag of deep visual words features of chest X-ray images to detect COVID-19 infection. *Sci. Rep.* **2021**, *11*, 23914. [[CrossRef](#)]
26. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.S.; Liang, H.; Baxter, S.L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* **2018**, *172*, 1122–1131. [[CrossRef](#)]
27. Ding, J.; Chen, B.; Liu, H.; Huang, M. Convolutional Neural Network With Data Augmentation for SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 364–368. [[CrossRef](#)]
28. Zhang, Y.D.; Dong, Z.; Chen, X.; Jia, W.; Du, S.; Muhammad, K.; Wang, S.H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimed. Tools Appl.* **2019**, *78*, 3613–3632. [[CrossRef](#)]
29. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2323. [[CrossRef](#)]
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Association for Computing Machinery: New York, NY, USA, 2012; Volume 2, pp. 1097–1105.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
32. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
34. Albert, E.; Alonso, D.; Arenas, P.; Genaim, S.; Puebla, G. Asymptotic Resource Usage Bounds. In Proceedings of the 7th Asian Symposium on Programming Languages and Systems, Seoul, Korea, 14–16 December 2009; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5904, pp. 294–310.
35. Kuo, F.Y. Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. *J. Complex.* **2003**, *19*, 301–320. [[CrossRef](#)]
36. Li, K.Q.; Pan, Y.; Zheng, S.Q. Efficient deterministic and probabilistic simulations of PRAMs on linear arrays with reconfigurable pipelined bus systems. *J. Supercomput.* **2000**, *15*, 163–181.1008103903338. [[CrossRef](#)]