healthcare

MDPI

Article

Analyzing Lung Disease Using Highly Effective Deep Learning Techniques

Krit Sriporn [1,2], Cheng-Fa Tsai [2,*], Chia-En Tsai [3] and Paohsi Wang [4]

[1] Department of Tropical Agriculture and International Cooperation, National Pingtung University of Science and Technology, Pingtung 91201, Taiwan; krit22kmutt@gmail.com

[2] Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung 91201, Taiwan

[3] Department of Biochemistry and Molecular Biology, National Cheng Kung University, Tainan 70101, Taiwan; plusntsai@gmail.com

[4] Department of Food and Beverage Management, Cheng Shiu University, Kaohsiung 83347, Taiwan; 0627@gcloud.csu.edu.tw

* Correspondence: cftsai@mail.npust.edu.tw; Tel.: +886-08-770-3202 (ext. 7906)

Received: 21 March 2020; Accepted: 20 April 2020; Published: 23 April 2020

Abstract: Image processing technologies and computer-aided diagnosis are medical technologies used to support decision-making processes of radiologists and medical professionals who provide treatment for lung disease. These methods involve using chest X-ray images to diagnose and detect lung lesions, but sometimes there are abnormal cases that take some time to occur. This experiment used 5810 images for training and validation with the MobileNet, Densenet-121 and Resnet-50 models, which are popular networks used to classify the accuracy of images, and utilized a rotational technique to adjust the lung disease dataset to support learning with these convolutional neural network models. The results of the convolutional neural network model evaluation showed that Densenet-121, with a state-of-the-art Mish activation function and Nadam-optimized performance. All the rates for accuracy, recall, precision and F1 measures totaled 98.88%. We then used this model to test 10% of the total images from the non-dataset training and validation. The accuracy rate was 98.97% for the result which provided significant components for the development of a computer-aided diagnosis system to yield the best performance for the detection of lung lesions.

Keywords: convolutional neural network; optimizer methods; lung disease; image classification; image processing; Mish activation function

1. Introduction

The World Health Organization, using the latest statistics from the year 2018, reported that worldwide, there were 10.4 million patients and 1.6 million deaths from lung disease such as pulmonary tuberculosis. Lung disease is an infectious disease that causes a large number of deaths. Lung disease consists of many types such as pulmonary tuberculosis, pneumonia, effusion, mass, infiltration. lung disease often occurs in developing countries along with human immunodeficiency virus (HIV) and diabetes, which will immediately affect the immunity and infection of lung disease. This disease is a respiratory disease, meaning that it causes lung infections in the thoracic area of patients [1,2]. In early diagnosis by a doctor, chest X-ray (CRX) films are used. CRX films are used to determine the position and size of the lung disease in the chest. The patient's lung examination uses CRX films, which almost all hospitals have available due to their inexpensive cost compared with magnetic resonance imaging (MRI) and computed tomography (CT) scans; thus, CRX films are a popular method for diagnosis as they can represent the organ structure inside the thoracic area in the body [3–5]. To summarize the results, thoracic CRX films are used to take images to diagnose the thoracic region. The patients infected by lung disease are numerous and still increasing in number and it takes a great deal of time for the doctor to diagnosis the disease due to a lack of radiologists. A computer-aided diagnosis (CAD) system is also used in lung disease screening [6].

Presently, doctors use CAD to reduce time for diagnosis disease and increase the convenience of diagnostics. A CAD system can be divided into three different basic technologies [7]. The first technology is image processing for extracting and enhancing the specific characteristics of the images, such as finding the lesions of the patient's disease in the CXR films to learn and diagnose the location of the lung disease, as well as for training CAD schemes. CAD diagnostics are also inaccurate because the CRX films of each patient have different characteristics and anatomic structures, such as body fat or distorted bones. Various image processing techniques have been employed for various types of lesions. Some of the most commonly used techniques include filtering analysis according to morphologic filtering, Fourier transform, different image techniques, and transformations are also used and may cause a diagnosis fault [8].

The second technology is used to measure the quantitation of image features such as contrast, size and the shape of the lung lesions. It is possible to define many features as some mathematical formulas may not be easily understood by human observation. However, these formulas are generally helpful in determining, at least in the beginning phase of CAD development, the image features that are known and subjectively understood by radiologists. Further, the accuracy of their diagnosis is generally very high and reliable. One of the most important factors in the development of CAD schemes is to find the unique features that can distinguish between lesions and other normal anatomic structures [9,10].

The third type of technology is data processing of the differences between normal and abnormal patterns based on the image features. The simplest and most common method used in this step is a rule-based method, which may be established depending on the understanding of the lesion and other normal anatomic structures. Thus, it is important to understand that a rule-based method may provide useful information to improve CAD schemes. Other techniques used include discriminant analysis and decision-trees. It is our experience that a combination of a rule-based method with other methods like an artificial neural network (ANN) tends to produce the best results in terms of the high performance of the CAD system. As the basic concept of CAD is wide and general, CAD can be applied to all photographic styles, including conventional irradiation, CT, MRI, ultrasound imaging and nuclear medical imaging. CAD schemes, have been developed for many types of examinations, on all parts of the body, including the abdomen, chest, skull and the vascular system [11–13].

At present, many researchers improved the performance of the CAD system by using artificial intelligence (AI) technology, which is developing significantly, resulting in the technique used to analyze high-accuracy data. Deep learning uses the principle of machine learning, which can determine the functioning of the human brain. Data are used to create patterns for decision making, and deep learning can be applied to image processing analysis. Deep learning applies to various sciences, including medical diagnosis, and uses the convolutional neural network (CNN) model concept developed based on an ANN to select the features and classify the information that the CNN has developed to classify multimedia sources of data, such as video, sound, images and text, among others. Presently, the CNN's structure adds greater performance, such as an activation function and an optimizer, which are significant components of the CNN's structure. The traditional CNN model and various techniques have been developed to produce higher accuracy image classification and a greater speed of CNN processing for preprocessing, transfer learning, etc., as well as adjust the weights of image training from the ImageNet dataset and parameters in the CNN structure [14–17]. Moreover, transfer learning is important to give the CAD system the ability to accurately recognize images for diagnosis of the lung disease, that is, a dataset with enough data should be used for the training of a CNN model; the characteristics of the dataset must have features associated with the image's recognition. Many organizations are aware of the importance of such a dataset for the large-scale training of up to 1.2 million or 1000 approximately image types to teach models more accurately to classify images, such as in ImageNet. Previous research also used transfer learning techniques to enhance high-accuracy models [18,19]. The development of the CNN model has increased its performance using transfer learning, but the CNN model is a complex construction. If the CNN model's development uses incorrect patterns, the CNN model may reproduce false or inaccurate images. Computer processing is now faster than in previous periods and can support the CNN model with a more sophisticated structure, as well as large-scale dataset processing, though using transfer learning is still more popular [20].

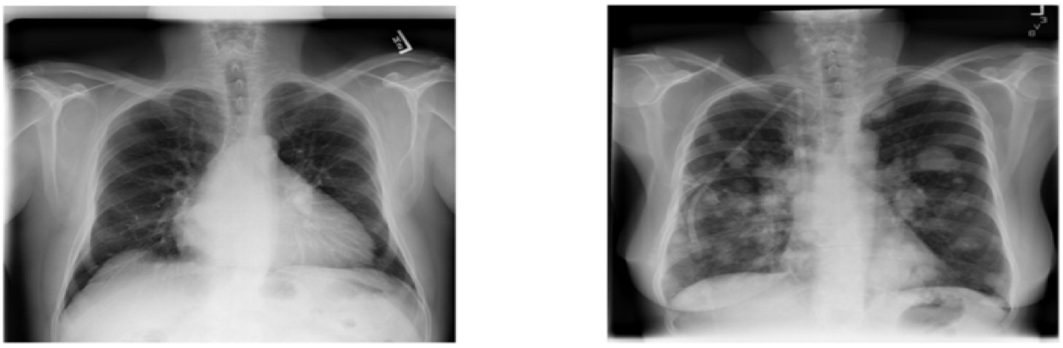
2. Materials

2.1. Problem Definition

Lung disease is an epidemic that can be easily infected and cause many lung related problems in patients. X-ray method is popular treatment used to diagnose lung disease due to its efficiency and low price compared to other methods [5]. For example, many doctors and radiologists use an X-ray film to diagnose lung disease. As these are time taking, medical personnel are unable to respond for medical treatment on time [6]. Therefore, the use of CAD system can help medical screening to reduce the duty of medical personnel. Many researchers applied image analysis technology, such as Deep Learning with CNN models to improve the recognition of CAD system to analyze accurate images [9]. Deep Learning restrictions are inappropriate image feature that will be used to train CNN models, such as shape, size, color and dimension; moreover, the medical image is highly complex. Therefore, the diagnosis of CAD system is necessary to study the image analysis method efficiently by developing CNN models and image processing to be more efficient [14,15].

2.2. Dataset

The dataset used in this research was taken from the National Institutes of Health Clinical Center-America's Research Hospital [21], which contains X-ray images from patients. The X-ray image is two dimensions, black and white; and image size is different. The dataset divided into normal lung and a lung disease, which are open-access medical images. We selected this dataset that consists of 1000 normal lung images and 1500 lung disease images with different image features by adjusting the selected images with image processing method for appropriate image with training of CNN models in order to be able to learn the image features (color, size, dimension and shape) from X-ray images. The dataset contains images from the scoliosis patients' spines are used for dataset training in the CNN models. An example of a normal lung and a lung disease is shown in Figure 1.



(**a**) (**b**)

Figure 1. A normal lung (**a**) and a lung disease (**b**).

2.3. Methodology

The research divided this section into four parts. The first part involved preparation of the three CNN models for training and validating and testing the dataset. The second part involved transfer learning, dropout techniques and the Mish activation function to improve the efficiency of the CNN models. The third part used seven optimizer methods and Cross-entropy loss function to optimize the performance of CNN models in predicting lung disease from chest X-ray images. The fourth part involved evaluating CNN models performance for lung disease prediction from the chest X-ray film, as shown in Figure 2.

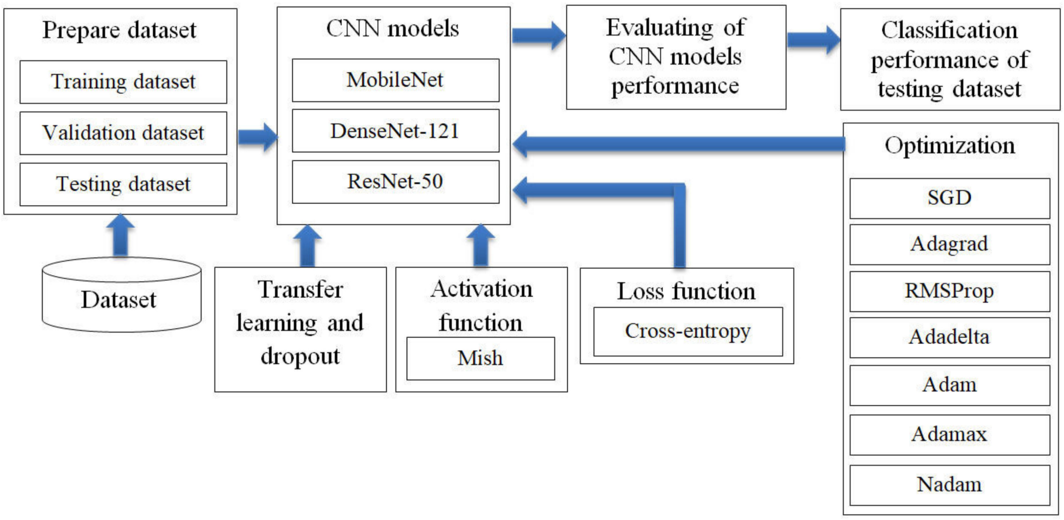


Figure 2. Research methodology: the optimization of convolutional neural network (CNN) models in predicting lung disease from chest X-ray images by using seven optimizer methods such as stochastic gradient descent (SGD).

2.3.1. Preparation of the CNN Models

MobileNet

There are various deep learning architectures in the computer vision field. Many previous studies compared the accuracies of different architectures and determined the parameter values suitable to adjust the CNN architecture designed with computer vision. Computer vision emphasizes the accuracy and time savings, although it can also save the overuse of hardware. MobileNet architectures are added to convolutional layers with a kernel with size of 1 × 1 to decrease the number of times of multiplication iterations. Computer vision is determined by using a kernel with a size of 1 × 1 apply inception architectures within the MobileNet architecture. A new type of inception-based CNN architecture is called MobileNet. MobileNet uses inception architecture to build kernels of three dimensions with a size of 1 × 1 × M, called a depth wise convolution kernel; we build a pointwise convolution kernel, as shown in Figure 3a–c [22–25].

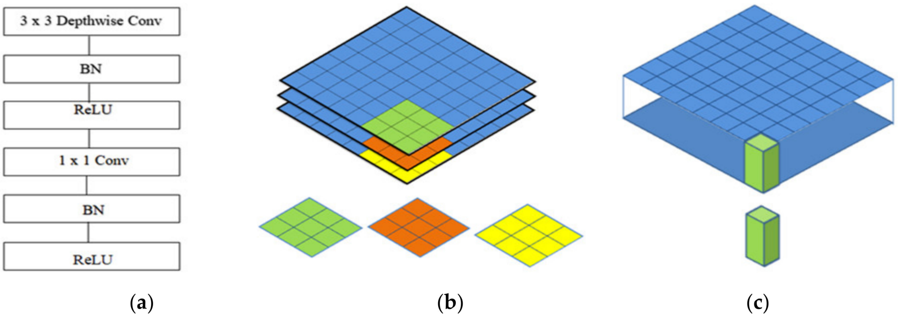


Figure 3. The architecture of MobileNet: (**a**) the convolutional layers of MobileNet; (**b**) the depth-wise convolution kernel; (**c**) the pointwise convolution kernel.

Resnet-50

The residual network (ResNet) was proposed by the Microsoft Corporation and won the 2015 ILSVRC competition. Global average pooling instead of fully connected layers is also used by ResNet-50, so the model of its 50-layer network is not too large. Convolution kernels of different sizes, such as 1 × 1 and 7 × 7, are used in the network to increase the diversity of its convolutions. Consequently, ResNet-50 has become very popular in the classification of image datasets. ResNet-50 elevates the concept of residuals where the image in the input goes directly to the output without experiencing a neural network. Consequently, the original image is maintained. This shortcut connection of ResNet-50 was introduced via the basic idea of a deep residual network, which can pass over some layers. In addition, two mappings were proposed in ResNet-50, one is identity mapping, and the other is residual mapping [26,27], as shown in Figure 4.

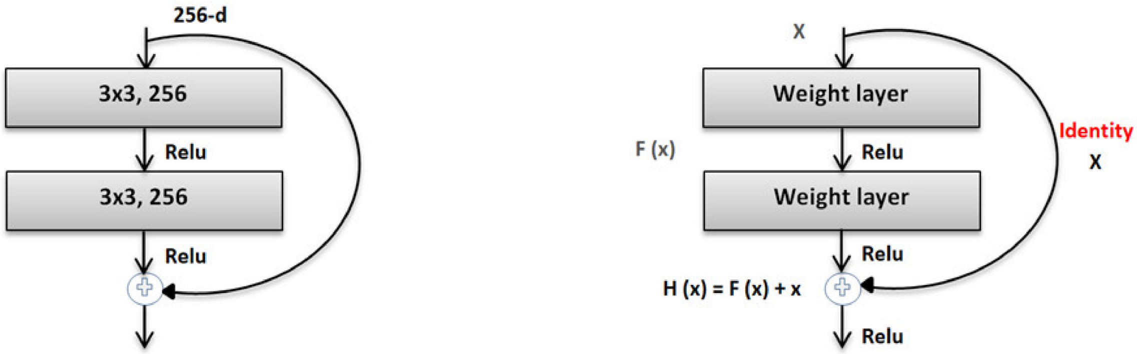


Figure 4. The architecture of ResNet-50.

DenseNet-121

In DenseBlock, each layer has feature maps of an equal size, and within the channel dimension, it is possible for these DenseBlocks to be connected. The function for a nonlinear DenseBlock employs a BN + ReLU + 3 × 3 convolution. It is also important to consider the different types of ResNet; after all, DenseBlock convolutions of the individual layers, the output characteristics of the images are the image features. Assuming a number of channel input feature map layers, the feature is then entered into the channel layer, which increases the number channels of layers. Thus, despite setting a smaller input, DenseBlock is large as a consequence of the features resulting from reuse, whereby each layer has own unique features, this is known as a DenseNet-A structure. The latter layer will have a rather large input and therefore, the interior portion of the DenseBlock can make use of an additional layer that acts as a bottleneck in order to reduce the number of arithmetical manipulations required, primarily through the addition of a 1 × 1 convolution to the initial form, as can be seen in Figure 5, specifically for the BN + ReLU + 1 × 1 Conv + BN + ReLU + 3 × 3 convolution, which is known as a DenseNet-B structure. When a 1 × 1 convolution-obtained image feature seeks to lower the number of features, the computational efficiency is improved [28–30], as shown in Figure 5.

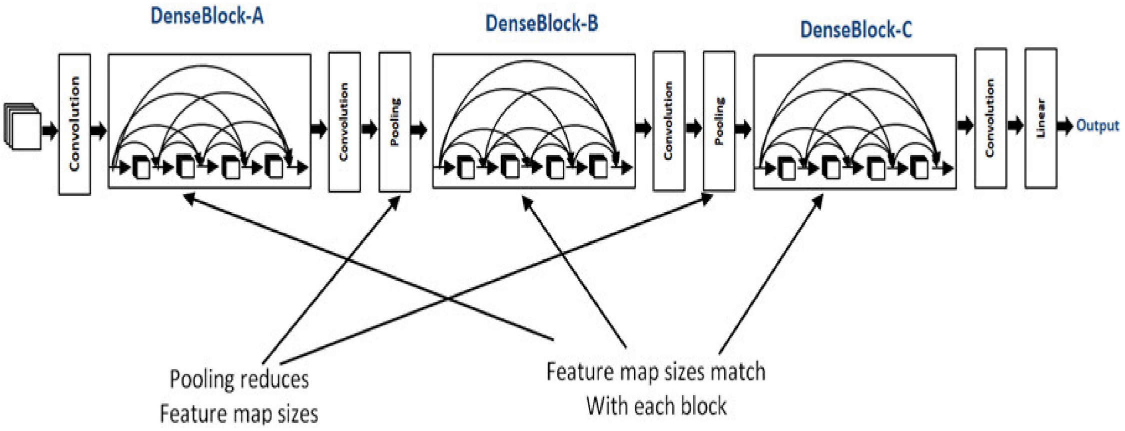


Figure 5. The architecture of DenseNet-121.

2.3.2. Improvement of CNN Models Efficiency

Transfer Learning

During its training and validation, a CNN model has a complex structure, which involves many Parameters (Weights); thus, the CNN model has a complex structure. For the initial training and validation of transfer learning, we used a large dataset (the ImageNet dataset), containing 1000 classes or 1.2 million images, which is effective for timely and accurate image classification [19,20]. This research takes advantage of transfer learning to improve the CNN models as shown in Figure 6.

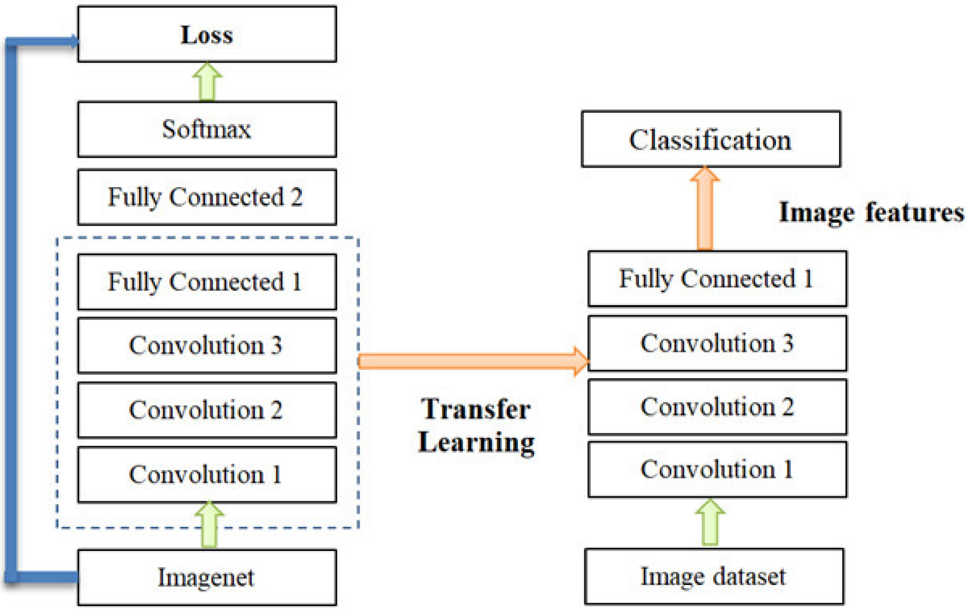


Figure 6. Convolutional neural network (CNN) transfer learning.

Dropout Technique

In order to make the CNN model more effective for the classification of images we adjusted the parameter values to suit the dataset for the training model; these parameters included batch size, activation function and the optimizer. Another main problem of CNN models with complex structures is overfitting. In this research, the dropout technique is used to solve the overfitting problem. Teaching a neuron network requires a large dataset, which is used to train the systems several times to create a deep model for classifying accurate images. However, there is the possible risk of overfitting, which may cause the model to classify inefficient images [31,32]. However, a dropout can prevent overfitting, as shown in Figure 7.

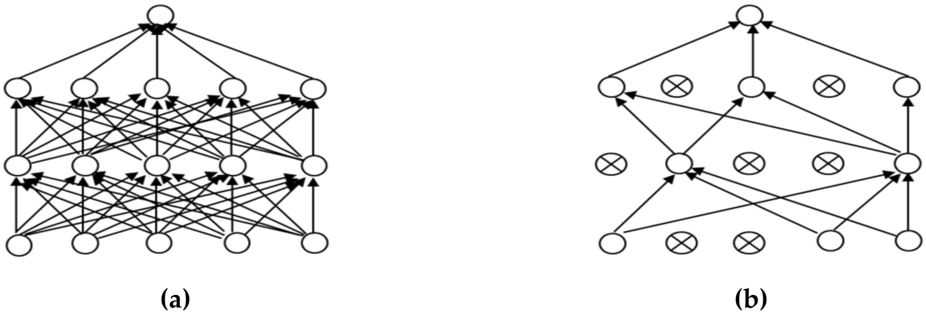


Figure 7. Dropout: (**a**) a neuron before a dropout; (**b**) a neuron after a dropout.

Mish Activation Function

Mish is superior to ReLU at high significance levels ($p < 0.0001$). Mish has been tested using more than 70 benchmarks, including image classification, segmentation and generation and has been compared with 15 other activation functions. In addition, the Mish function also guarantees the smoothness of each point. The characteristics of the self-selection gate are able to replace activation functions such as ReLU (point function), as shown in Formula (1) [33]. These functions can receive a single scalar to change the network parameters without entering any scalar. Mish takes inspiration from swish's self-gating property, where scalar input is provided to the gate. Self-gating is able to replace activation functions like ReLU without changing the network parameters. Variable (x) in Formulas (1) and (2) represents the input values of the activation function. Mish has no upper bound but does have a lower bound; Mish also has smooth and non-monotonic properties that improve the results [34], as shown in Formula (2).

$$\mathrm{ReLU}(\mathrm{x}) = max(0, x) \tag{1}$$

$$\mathrm{Mish}(\mathrm{x}) = x \times tanh(softplus(x)) \tag{2}$$

2.3.3. Loss Function

The function of the loss function layer (loss layer) is to calculate the expected results predicted by the key features, to make differential corrections with the real results, and to combine the gradient drop optimization function to increase the convergence speed of network weight renewal. The most commonly used correction functions are loss functions, such as mean absolute error (MAE), mean square error (MSE) and cross-entropy [35]. This experiment used (cross-entropy) as shown in Formula (3).

$$\mathrm{f}(\mathrm{x};\mathrm{u}) = exp\left(-\sum_{j=1}^{N} \frac{xj}{uj}\right)\prod_{j=1}^{N} \frac{1}{uj} \tag{3}$$

Cross-entropy was used to determine the sample to be calculated, which consisted of x_j with u_j values, where x_j represents random weights, and u_j represents the weights, which are exponentially distributed and independent of each other.

2.3.4. Evaluating CNN Model Performance for Lung Disease Prediction from the Chest X-ray Film

This study utilized several common parameters to evaluate architectural deep learning examination performance using three convolutional neural networks. TP indicates true positive (True Positive) (i.e., predicted to suffer from lung disease and actually suffering from lung disease), while TN is true negative as predicted to suffer from lung disease and actually suffering from lung disease, while TN is true negative (True Negative) (i.e., the predicted absence of lung disease and no recorded presence of lung disease). FP is a false positive, which predicts the development of lung disease that is not actually present, while FN is a false negative which predicts no development of lung disease despite the real presence of lung disease, the Formulas (4)–(7) based on the work in [36–39].

1. Accuracy can be represented as the number of classified data sets divided by the total number of data test sets, as shown in Formula (4).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

2. The precision rate indicates the correct prediction of the number of categories divided by the total number of data falling into that category, as shown in Formula (5).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

3. The recall rate (Recall) correctly predicts the number of categories divided by the total number of data actually belonging to each category, as shown in Formula (6).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

4. The F1measure is used to balance the assessment of accuracy and recall rates and also to evaluate classification models, as shown in Formula (7).

$$\text{F1measure} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

3. Implementation Details

3.1. Computer Hardware and Software Setting.

In this experiment we created CNN models using computer hardware and software as the execution environment, as shown in Table 1 below.

Table 1. The experimental environment.

Computer Equipment	Application and System Software
Central Processing Unit: Intel I7-6700 [1], 3.40 gigahertz	Operating System: Windows 10 [2]
Random Access Memory: 32 gigabytes	Data Scikit-learn 0.21.2
Graphics Processing Unit (GPU): NVIDIA®GTX 1060 [3], 6 gigabytes	TensorFlow-GPU 1.13.1
Solid State Drive: 250 gigabytes	Python 3.7.3 and Keras-GPU 2.2.4

[1] Central Processing Unit Intel I7-6700 is product name of Intel Corporation, Santa Clara County, CA, USA; [2] Windows 10 is product name of Microsoft Corporation, Washington, DC, USA; [3] NVIDIA®GTX 1060 is product name of NVIDIA Corporation, Santa Clara County, CA, USA.

3.2. Dataset Setting

Due to the chest X-ray film using a wide range of sizes and a large size not being suitable for dataset training to create the CNN models, it is necessary to shrink the X-ray image to reduce time for training the CNN models. This process reproduces the model and converts the image to a matrix size of 224 × 224 × 3, which is the normal size for dataset training in CNNs. Although the X-ray image is black and white, the data are red, green, and blue using an RGB three-color system [40].

The technique of adding the number of the images in a dataset via the data augmentation technique is a long-standing technique that solves datasets of a small number. There are several lung shapes other than a normal lung. In fact, the chest X-ray image has a slightly distorted angle from the original images, though not over 90 degrees. This model cannot be applied effectively considering the actual performance of the original dataset [41,42]. This research has applied the data augmentation technique, which will rotate only some of the images. In order to apply the augmentation technique to the chest X-ray dataset and not to distort the original image for improving the dataset efficiency to use training the CNN models, we used the shuffle sampling technique in combination with the rotated images. The angle will be random for each image in the range. The duplicate images of the dataset reduce this problem. Using only the shuffle sampling technique reduces the duplicate images in the dataset [43,44]. Sometimes an error occurs, such as an image being taken with a tilted angle, the presence of a distorted lung shape in a normal image, or a patient with a scoliosis spine. In order to develop a modeling experiment that can manage an image better without having to take a tilted image, the scoliosis patients' spines are used for dataset training in the model. Due to the distorted X-ray image not being present in the dataset used in this research, it is possible to create a chest X-ray image that represents a distorted X-ray image by rotating it. The image is assigned an angle of −10 to 10 degrees randomly based on the dataset training [45,46]. Figure 8 illustrates comparative images of the lung shapes of the images created with the patient's scoliosis spines. Some parts of the lungs have similar shapes. The images on the left and right are the chest X-ray images generated by image rotation, while the middle side is the chest X-ray image of the patient with an abnormal spine [47,48].

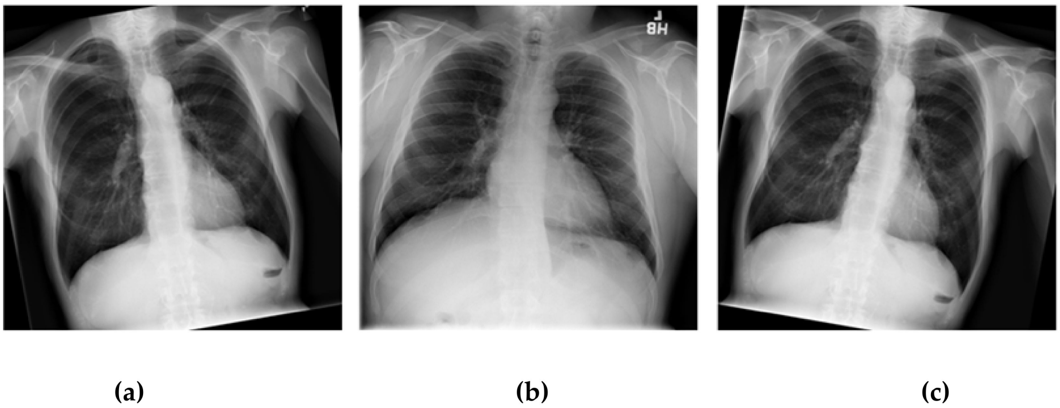


Figure 8. Examples of distorted chest X-ray images: (**a**) the patient's normal spine tilted −10 degree by rotating; (**b**) a patient's abnormal spine; (**c**) a patient's normal spine tilted 10 degree by rotating.

3.3. *Optimizer Setting*

The gradient descent method is presently the most famous optimizer method and is also the most commonly used method for optimizing a CNN. The latest machine learning libraries contain various algorithms for enhancing the gradient descent method, but these algorithms are not disclosed and are used as black box optimizers to develop the performance of a CNN. This experimental research used seven famous optimizer methods, the formulas of which are shown below.

1. Stochastic Gradient Descent (SGD) updates a high variance value impact into a loss function value with direct variation and different intensities. This is a good method because it easily and efficiently obtains the minimum value in the center of the field compared to the other algorithms [29]; the formula is shown below.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J\left(\theta; \chi^{i}; y^{i}\right) \tag{8}$$

For SGD, the determined learning rate is 0.1 (η), the input is χ^{i}, and the label is y^{i} for the training, the gradient for loss function uses $_{\theta}J$, and the validation dataset. θ is the cost function of the calculating gradient.

2. Adagrad is an algorithm that can optimize the learning rate for the parameter in a suitable range by increasing its updating for a smaller number of parameter values. However, little time is used to update the various numbers of the parameter [30]; the formula is shown below.

$$\theta\mathrm{t}, i = \theta t - 1, i - \frac{\eta}{\sqrt{Gt - 1, i + \varepsilon}} . gt - 1, i \tag{9}$$

Adagrad's learning rate is 0.1 (η). $G_{t-1,\,\mathrm{i}}$ is the gradient of the objective function (θ_t) for the calculating gradient at time step t, ε = le^{-08} and g_t is the current gradient.

3. Adadelta can constrain the collection of the calculation of gradient descent to resize the resulting weight value instead of collecting the W value from the previous update. The aim is to repeat the decaying learning rate of all previous gradients [49]; the formula is shown below.

$$\mathrm{E}\left[\mathrm{g}^2\right]\mathrm{t} = \gamma E\left[g^2\right]t - 1 + (1-\gamma)g^2 t.$$

The Adadelta learning rate is 0.1 (η); the solving fraction problem (*rho*, or γ) of the gradient is 0.9 at time step t, the diagonal matrix is g^2t, and the decaying average is $E[g^2]t$.

4. RMSprop is a method for collecting the cost value of the gradient descent that is used for learning by applying the gt rate, MeanSquaret and x represent the historical learning rate and solves the problem of Adagrad's radical reduction in learning rates [32]. The formula is shown below.

$$\mathrm{E}\left[\mathrm{g}^2\right]\mathrm{t} = 0.9E - \left[g^2\right]t - 1 + 0.1g^2 t.$$

The RMSprop learning rate is 0.001 (η). We used Hinton's input to set the solving fraction (*rho* or γ) of the gradient as 0.9 at time step t, the diagonal matrix is g^2t and the decaying average was set as $E[g^2]t$.

5. Adaptive Moment Estimation (Adam) is an optimizer that can adjust the learning rates for each parameter at a time. It can also solve the decay of the gradients in each subsequent step along with Adadelta and explain the origination of decaying $\mathrm{m_t}$, as well as gradients [50]. The formula is shown below.

$$\mathrm{m_t} = \beta_1 m_{t-1} + (1-\beta_1)gt \tag{10}$$

To create the vectors of Adam using m_t at time step t, we set β_1=0.9 according to the advice provided by the authors of Adam.

6. Adamax is a variant of Adam and provides a simpler range for the upper limit of the learning rate. This model reduces the unstable problems of the parameter values. The formula is shown below.

$$\mathrm{v_t} = max\left(\beta_2 \times vt - 1, \left|gt\right|\right) \tag{11}$$

The Adamax learning rate is 0.002 (η), based on the work in [50] using β_2 = 0.999, where $|gt|$ is the current gradient and v_t is the update rule scales of the gradient in Adamax.

7. Nadam is similar to Adam with Nesterov momentum. It has a stronger constraint in its learning rate and also has a more direct impact on the update of the gradient. The formula is shown below;

$$\theta t + 1 = \theta t - \frac{\eta}{\sqrt{\hat{u}_t + \varepsilon}}\left(\beta_1 \hat{a}_t + \frac{(1-\beta_1)gt}{1-\beta_1^t}\right) \tag{12}$$

The Nadam learning rate is 0.002 (η), based on the work in [51] using β_1 = 0.9, ε = le^{-08} and the objective function (θ_t), with $\hat{a}_t$ and $\hat{u}_t$, provide an updated rule for Nadam at time step t.

3.4. Parameter Setting for Training and Validation

This experiment uses three famous CNN models, with input size of 224 × 224 and seven optimizer methods for optimization of these CNN models. The iterations are 70 epochs, the training dataset size is 80% and the validation dataset size is 20% of the total number of the lung disease dataset;

the convolutional neural network output layer contains two classes that comprise the normal and lung disease status. The batch size is 20 images, the activation function is Mish, the loss function is Cross-entropy, the dropout technique for solving the overfitting problem is 0.5, based on [31,32]. Table 2 shows a list of the learning rates of each optimizer method. The parameter values of the learning rates based on the work in [49–53].

Table 2. The learning rate of each optimizer method for training the CNN models.

Optimization Algorithm	SGD	Adagrad	Adadelta	RMSprop	Adam	Adamax	Nadam
Learning rate	0.01	0.01	0.01	0.001	0.001	0.002	0.002

Stochastic gradient descent (SGD).

4. Experimental Results

Tables 3–5 illustrate the performance of the CNN models combined with the seven optimizers and Mish comparison with traditional CNN models. Table 3 shows the lung lesion detection performance of MobileNet with Mish compared to traditional MobileNet, which uses ReLU. The best results with MobileNet were obtained by using Nadam and Mish, with an accuracy rate of 93.28%, a precision rate is 93.24%, a recall rate of 93.46% and an F1 measure rate of 93.27%. For MobileNet using SGD with ReLU (traditional method), the accuracy rate was 74.48%, the precision rate was 75.93%, the recall rate was 73.48%, and the F1 measure rate was 73.52%.

Table 3. The comparative performance of MobileNet with Mish and ReLU using different optimizer methods.

Activation Function and Optimizer Methods	Accuracy %	Precision %	Recall %	F1 Measure %	Training Time
Nadam & ReLU	90.51	90.5	90.53	90.5	58 min 31 sec
Nadam & Mish	93.28	93.24	93.46	93.27	58 min 48 sec
Adamax & ReLU	87.17	87.31	86.9	87.04	58 min 05 sec
Adamax & Mish	90.44	90.42	90.64	90.43	58 min 18 sec
Adam & ReLU	86.91	87.19	86.56	86.75	58 min 27 sec
Adam & Mish	89.93	89.93	90.15	89.91	58 min 38 sec
Adadelta & ReLU	83.56	84.11	83.03	83.26	56 min 20 sec
Adadelta & Mish	87.6	87.54	87.73	87.57	56 min 34 sec
RMSprop & ReLU	83.04	83.68	82.47	82.71	55 min 47 sec
RMSprop & Mish	85.37	85.33	85.52	85.34	56 min 07 sec
Adagrad & ReLU	79.94	81.4	79.06	79.29	56 min 36 sec
Adagrad & Mish	80.89	80.8	80.94	80.84	57 min 03 sec
SGD & ReLU	74.48	75.93	73.48	73.52	59 min 27 sec
SGD & Mish	76.59	76.68	76.82	76.57	59 min 48 sec

Table 4. The comparative performance of ResNet-50 with Mish and ReLU using different optimizer methods.

Activation Function & Optimizer Methods	Accuracy %	Precision %	Recall %	F1 Measure %	Training Time
Nadam & ReLU	96.64	96.58	96.81	96.63	116 min 50 sec
Nadam & Mish	97.59	97.52	97.74	97.58	117 min 12 sec
Adamax & ReLU	94.23	94.2	94.43	94.22	111 min 21 sec
Adamax & Mish	95.09	95.07	95.31	95.08	111 min 42 sec
Adam & ReLU	93.45	93.42	93.65	93.44	113 min 39 sec
Adam & Mish	94.49	94.44	94.67	94.48	113 min 54 sec
Adadelta & ReLU	92.68	92.66	92.89	92.67	109 min 34 sec
Adadelta & Mish	93.54	93.51	93.74	93.53	109 min 39 sec
Rmsprop & ReLU	91.99	91.96	92.18	91.98	108 min 16 sec
Rmsprop & Mish	92.42	92.38	92.61	92.41	109 min 12 sec
Adagrad & ReLU	84.76	84.71	84.62	84.66	111 min 04 sec
Adagrad & Mish	87.17	87.09	87.12	87.11	111 min 34 sec
SGD & ReLU	79.43	79.34	79.24	79.28	117 min 42 sec
SGD & Mish	79.69	79.6	79.5	79.54	118 min 12 sec

Table 5. The comparative performance of DenseNet-121 with Mish and ReLU using different optimizer methods.

Activation Function& Optimizer Methods	Accuracy %	Precision %	Recall %	F1 Measure %	Training Time
Nadam & ReLU	98.62	98.56	98.69	98.61	123 min 10 sec
Nadam & Mish	98.88	98.83	98.91	98.87	123 min 29 sec
Adamax & ReLU	95.69	95.65	95.9	95.68	118 min 27 sec
Adamax & Mish	96.38	96.32	96.54	96.37	118 min 39 sec
Adam & ReLU	95.09	95.02	95.22	95.08	119 min 40 sec
Adam & Mish	95.43	95.37	95.59	95.42	120 min 17 sec
Adadelta & ReLU	91.82	91.75	91.83	91.78	116 min 22 sec
Adadelta & Mish	92.34	92.27	92.34	92.3	116 min 47 sec
RMSprop & ReLU	89.15	89.12	89.05	89.08	114 min 29 sec
RMSprop & Mish	91.65	91.59	91.62	91.6	115 min 09 sec
Adagrad & ReLU	85.19	85.29	84.92	85.04	117 min 48 sec
Adagrad & Mish	88.03	88.07	87.85	87.93	118 min 05 sec
SGD & ReLU	81.06	81.12	80.74	80.86	123 min 45 sec
SGD & Mish	82.53	82.69	82.15	82.31	123 min 57 sec

Figure 9 compares the efficiency between MobileNet and MobileNet combined with Mish and Nadam on the validation data, which comprises 20% of the 5810 images, or 1162 images. These images are split into two statuses, 538 images of a normal status and 624 images of a lung disease status for predicting lung lesions. The true class of confusion matrix for MobileNet combined with Mish and Nadam is shown in Figure 9b, which correctly predicted a normal status in 487 images and a lung disease status in 565 images; this model did not correctly predict lung lesions in 110 images. The MobileNet results are shown in Figure 9a, this model correctly predicted a normal status in 320 images and a lung disease status in 546 images.

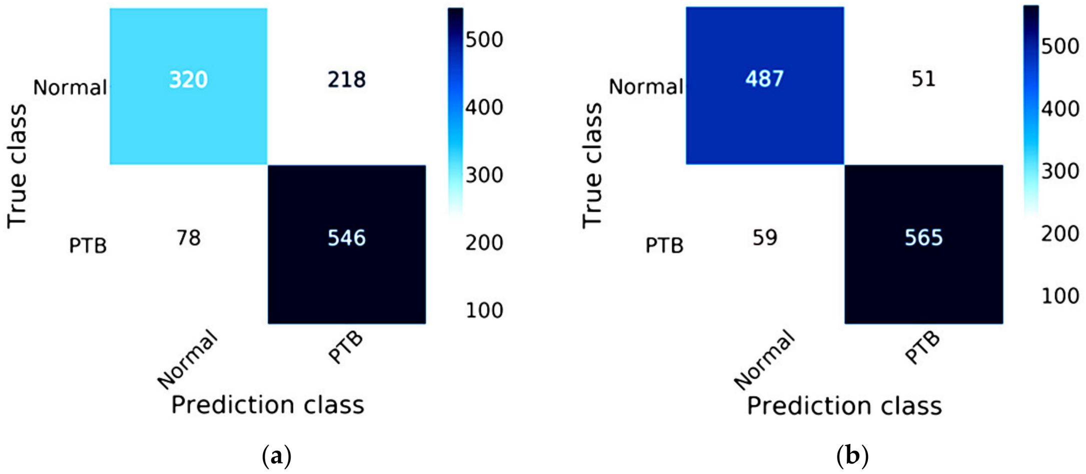


Figure 9. Confusion matrix of MobileNet: (**a**) the traditional model results of MobileNet; (**b**) the results of MobileNet combined with Mish and Nadam.

For ResNet-50, the best results were obtained by using Nadam and Mish, with an accuracy rate of 97.59%, a precision rate of 97.52%, a recall rate of 97.74%, and an F1 measure rate of 97.58%. For ResNet-50 using SGD with ReLU (the traditional method), the accuracy rate was 79.43%, the precision rate was 79.34%, the recall rate was 79.24%, and the F1 measure rate was 79.28%, as shown in Table 4.

The efficiency of ResNet-50 combined with Mish and Nadam is shown in Figure 10. The true class of the confusion matrix for ResNet-50 combined with Mish and Nadam is shown in Figure 10b. This model correctly predicted a normal status in 537 images and a lung disease status in 597 images of lung disease status and did not correctly predict lung lesions in 28 images. ResNet-50 is shown in Figure 10a; this model correctly predicted a normal status in 413 images and a lung disease status in 510 images.

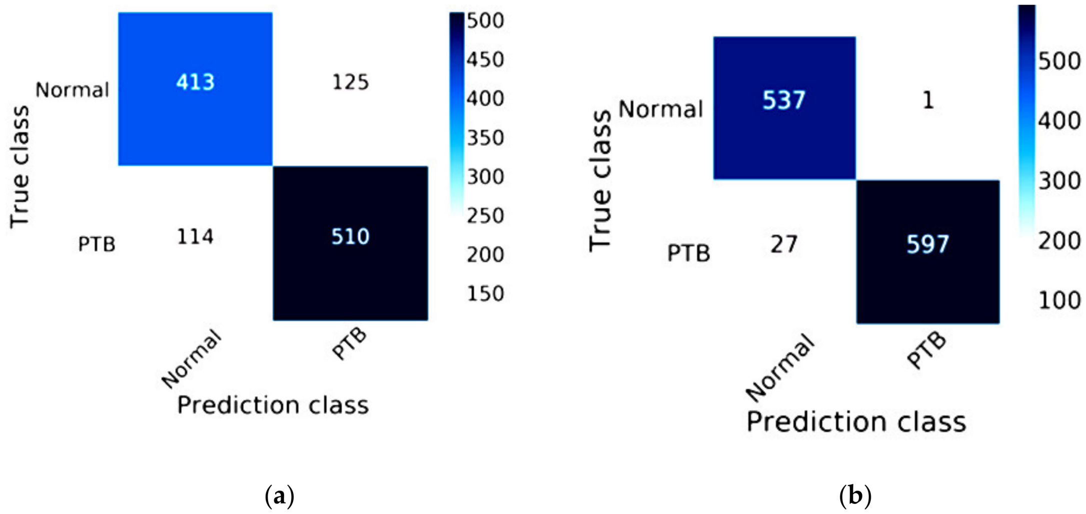
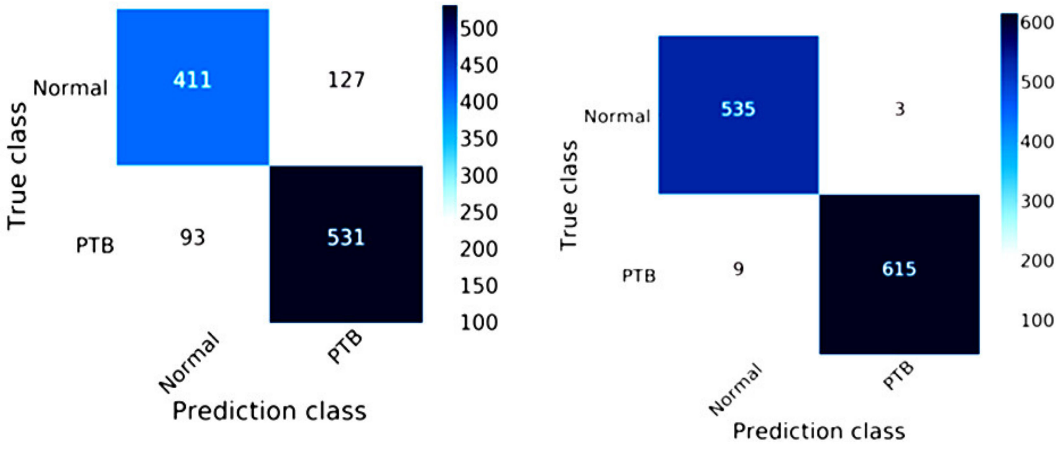


Figure 10. Confusion matrix of ResNet-50: (**a**) the traditional model results of ResNet-50; (**b**) the results of ResNet-50 combined with Mish and Nadam.

The efficiency of DenseNet-121 combined with Mish and Nadam is shown in Figure 11. The true class of the confusion matrix for DenseNet-121combined with Mish and Nadam is shown in Figure 11b. This model correctly predicted a normal status in 535 images and a lung disease status in 615 images and did not correctly predict lung lesions in 12 images. The DenseNet-121 results are shown in

Figure 11a; this model correctly predicted a normal status in 411 images and a lung disease status in 531 images.



(**a**) (**b**)

Figure 11. Confusion matrix of DenseNet-121: (**a**) the traditional method results of DenseNet-121; (**b**) the results of DenseNet-121combined with Mish and Nadam.

Table 5 shows the best performance for the detection of lung lesions using the optimizer method with the activation function, which can increase the potential of CNN models. In this research, using Nadam and Mish combined with DenseNet-121 predicted lung lesions with an accuracy rate of 98.88%. The precision rate was 98.83%, the recall rate was 98.91%, and the F1 measure rate was 98.87%. This is a higher accuracy rate than the prediction of lung lesions with the traditional method of DenseNet-121 which uses ReLU with SGD and offers an accuracy rate of 81.06%, a precision rate of 81.12%, a recall rate of 80.74%, and an F1measure rate of 80.86%, as shown below.

Table 6 describes the 580 testing images of the training and validation data, split into 232 images of a normal status and 348 images of a lung disease status from chest X-ray images. The traditional method of DenseNet-121 produced false predictions for 8 images of a normal status or 3.44% and 8 images of a lung disease status or 2.30%; the true predictions included 224 images of a normal status or 96.56% and 340 images of a lung disease status or 97.70%. DenseNet-121 combined with Mish with Nadam produced false predictions for 3 images of a normal status or 1.29% and 3 images of a lung disease status or 0.87%; true predictions were made for 229 images of a normal status or 98.71% and 345 images of a lung disease status or 99.13%.

Table 6. Comparative performance of model testing using DenseNet-121 combined with Mish and Nadam and the traditional method of DenseNet-121.

Models	Optimizers	Classes	Images	True Prediction	False Prediction
DenseNet-121 (Traditional method)	SGD	Normal Disease Overall	232 (100%) 348 (100%) 580 (100%)	224 (96.56%) 340 (97.70%) 564 (97.25%)	8 (3.44%) 8 (2.30%) 16 (2.75%)
DenseNet-121 and Mish	Nadam	Normal Disease Overall	232 (100%) 348 (100%) 580 (100%)	229 (98.71%) 345 (99.13%) 574 (98.97%)	3 (1.29%) 3 (0.87%) 6 (1.03%)

5. Discussion

The ability to determine important dataset features based on the values of each optimizer parameters vital to improve the time consumption and accuracy of image classification. In addition, data augmentation techniques can increase the potential, of image classification. The parameters used to fine-tune the performance of each optimizer method [54] are shown in Table 7.

Table 7. Parameters of each optimizer method.

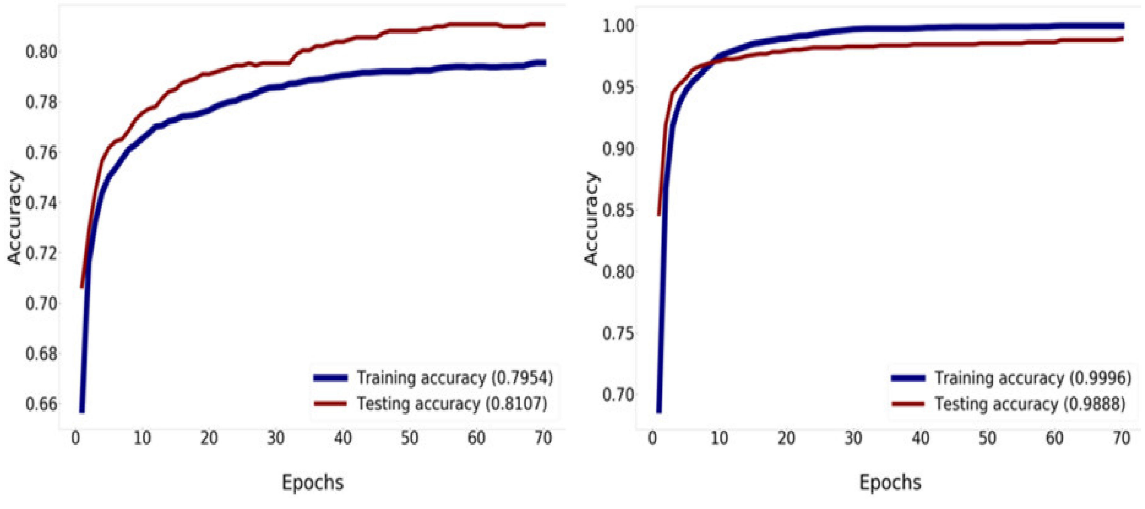
Optimizer Methods	Parameters
Nadam	Learning rate, beta_1, beta_2, epsilon, schedules-decay
Adamax	Learning rate, beta_1, beta_2, epsilon, decay
Adam	Learning rate, beta_1, beta_2, epsilon, decay
Adadelta	Learning rate, rho, epsilon, decay
RMSprop	Learning rate, rho, epsilon, decay
Adagrad	Learning rate, epsilon, decay
SGD	Learning rate, momentum, decay, nesterov

For training CNN models, MobileNet is a small model that requires little time for image classification; the highest accuracy of this experiment was 93.28%. This model is suitable for mobile computing device that require low power consumption for processing [45,46]. ResNet-50 is a popular classification model for predicting images; the highest accuracy in this experiment was 97.59%, which involved the problem solving of degradation by using identity mapping and residual mapping [26,27]. In this experiment, the best result of classification was 98.88%; this was accomplished by using DenseNet-121, which makes using of a bottleneck layer along with a construction for transition combinations and, using DenseBlock, offers a factor that indicates compression not exceeding a value of 1 [28,29]. There are limitations to this research. For example, our computer hardware features lower performance than the recommended requirements; thus, the application software for this experiment could not be used. Modern computer hardware has extremely high performance and can be used for large-scale image analysis.

Tables 3–5 compare the performance of the classification models using different optimization methods. The most accurate of the three CNN models was DenseNet-121 combined with Nadam and Mish, which provided an accuracy of 98.88%; the second-highest accuracy was ResNet-50 combined with Nadam and Mish, which provided an accuracy of 97.59%; and the third-highest accuracy was MobileNet combined with Nadam and Mish, which provided an accuracy of 93.28%. For a comparison of the time consumption of the classification models, the lowest time was found for MobileNet combined with RMSprop and ReLU, which provided a time consumption of 55 min 47 sec; the second-lowest time was ResNet-50 combined with RMSprop and ReLU, which provided a time consumption of 111 min 4 sec; and the third-lowest time was found for DenseNet-121 combined with RMSprop and ReLU, which provided a time consumption of 118 min 27 sec.

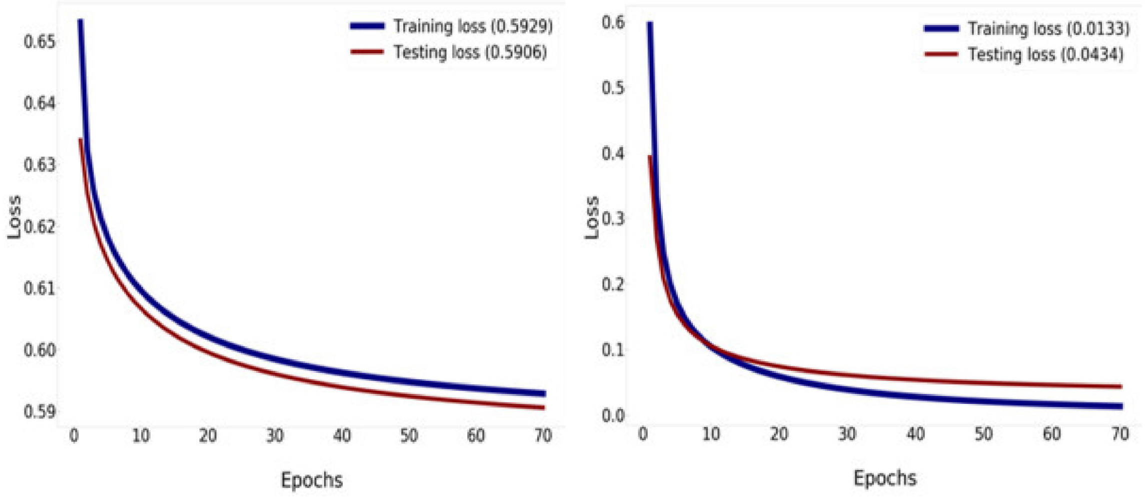
In order to speed up the training of the network, in this research we determined the batch size parameter based on the number of parameters used by each CNN model and the floating-point number of the activation function. Batch Size is the number of samples selected for a training session. A larger batch size will increase the learning speed of the model. Batch size directly affects the use of GPU memory. If the available GPU memory is not large, it is better to set the value smaller [55].

The Mish activation function is a new deep learning activation function that has a final accuracy better than Swish (+0.494%) and ReLU (+1.671%). In this work, Mish was superior to ReLU at high significance levels ($p < 0.0001$). The Mish function also guarantees the smoothness of each point. Mish has no upper bound but does, have a lower bound. Moreover, its smooth and non-monotonic properties all improve the results [34]. Figures 12 and 13 illustrate tests of the validation accuracy.



(**a**) (**b**)

Figure 12. Training and validation accuracy history: (**a**) training and validation accuracy of DenseNet-121; (**b**) training and validation accuracy of DenseNet-121 combined with Mish and Nadam.



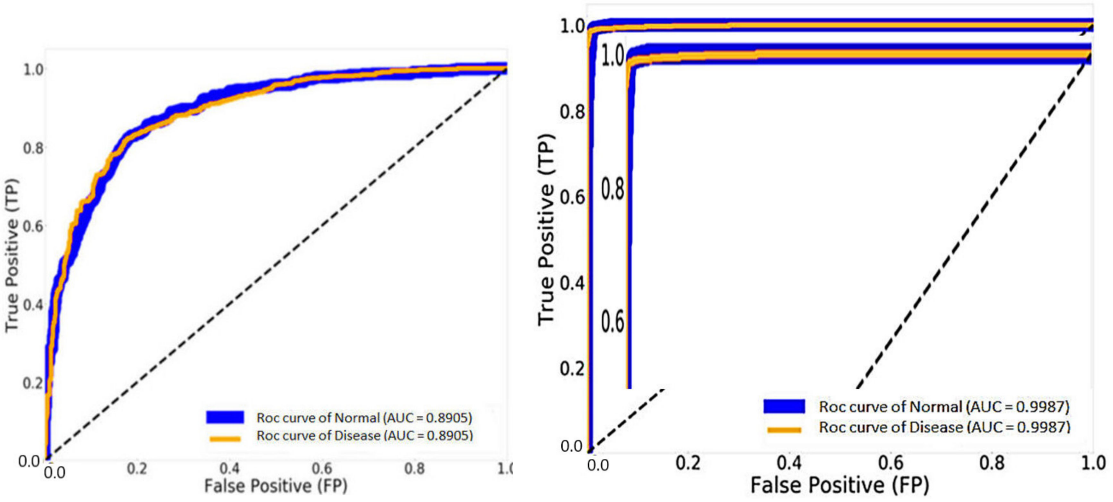
(**a**) (**b**)

Figure 13. Training and validation loss history: (**a**) training and validation loss of DenseNet-121; (**b**) training and validation loss of DenseNet-121 combined with Mish and Nadam.

Figure 12 compares the result of the training and validation accuracy between the traditional DenseNet-121 and DenseNet-121 combined with Mish and Nadam, which can increase the efficiency of accuracy up to 99.96% for training and 98.88% for validation, as shown in Figure 12b. DenseNet-121 can increase the efficiency of accuracy up to 79.54% for training and 81.07% for validation, as shown in Figure 12a. This research determined that 70 epochs are needed for training and validation history [56,57].

Figure 13 compares the results of the training and validation loss between the traditional DenseNet-121 and DenseNet-121 combined with Mish and Nadam, which can reduce the loss down to 0.0133% for training and down to 0.0434% for validation, as shown in Figure 13b. DenseNet-121 can reduce loss down to 0.5929% for training and 0.5906% for validation, as shown in Figure 13a.

Figure 14 shows the result of the AUC and ROC curves produced by the FP and TP rates, which evaluate the performance of our CNN models. Figure 14a illustrates 89.05% of the AUC with the traditional DenseNet-121 and 99.87% of the AUC with DenseNet-121 combined with Mish and Nadam, as shown in Figure 14b. This experiment can improve the efficiency of traditional CNN models by changing hyperparameters using Mish and seven optimizer methods while adjusting the suitable values for each optimizer parameter to determine the best result.



(**a**) (**b**)

Figure 14. The results of the area under the curve (AUC) and receiver operating characteristics (ROC) curve: (**a**) DenseNet-121 and (**b**) DenseNet-121 combined with Mish and Nadam.

6. Conclusions

This research was focused on applying modeling to detect traces of lung infection via a deep learning approach using the DenseNet-121 network, which was compared to other network models, such as ResNet-50 and MobileNet. The purpose of this research was to determine the efficiency of the three most well-known CNN models, MobileNet, Resnet-50 and Densenet-121, and to improve the efficiency of these CNN models by using Mish with seven optimizer methods to predict lung disease, as well as to compare the efficiency between traditional CNN models and CNN models using Mish with seven optimizer methods to predict lung disease.

The materials and methodology of this research was divided into four parts; the first part involved preparing the data method, which consisted of data augmentation techniques. Chest X-ray images featuring scoliosis of the spine in patients with abnormal lung shapes may look like lung disease symptoms and result in an erroneous diagnosis. Using the rotation technique in the data preprocessing stage for lung shape images can resolve this problem. Some areas of the chest X-ray image of a lung disease may look like a normal lung if the image processing technique is not suitable, which will lead to an incorrect diagnosis. Therefore, the suitable selection of an image processing technique to correctly classify lung disease is paramount. Using a preprocessing data technique can help create dataset training for the CNN model and increase efficiency. Related research has used a number of datasets with 5810 images for dataset training; this increased the processing time needed to create the CNN model.

The second part involved an activation function (Mish), transfer learning and dropout techniques to improve the efficiency of CNN models. Dropout was used to solve the overfitting problem and transfer learning was used to improve the efficacy of time consumption and the accuracy of image classification. The third part used a loss function (Cross-entropy) and seven optimizer methods consisting of Nadam, Adamax, Adam, Adadelta, RMSprop, Adagrad and SGD to determine the best

CNN model performance to predict lung disease. The fourth part involved evaluating CNN models performance for lung disease prediction from the chest X-ray film.

Optimization is useful for model training; and involves the batch size, activation function and optimizer. Optimization is used to adjust the weight of the connected lines in a neural network. These seven optimizer methods can determine if the weight parameter needs to adjust the learning rate of the CNN model. Research on activation functions remains on going, and ReLU still dominates the activation functions used for deep learning; however, this research was changed by the introduction of Mish. This activation function determines the scale of the output variable value from the input variable value and also guarantees the smoothness of each point. Mish can receive a single scalar to change the network parameters without entering any scalar. Mish takes inspiration from Swish's self-gating property, where scalar input is provided to the gate. Self-gating is able to replace activation functions like ReLU without changing the network parameters. Mish has no upper bound but does have a lower bound; further, its smooth and non-monotonic properties all improve the results [49]. Weights emphasize the importance of the input variable value that is used to determine the weight value of the input variable with the connected neuron before transferring the input variable value to the activation function. Weights can be changed by model training for the most accurate model [58,59].

There are limitations to this research. For example, our computer hardware features lower performance than the recommended requirements; thus, the application software for this experiment could not be used. Modern computer hardware has extremely high performance and can be used for large-scale image analysis. The creation of an efficient CNN model based on a number of images and preprocessing data techniques significantly improved the model's efficiency. Some CNN structures are suitable parameters for dataset training to reduce time of training the CNN models and increase accuracy. Creating a CNN model using the rotation technique allowed us to customize rotation of the images by −10 to 10 degrees and train the dataset created by the CNN models [47,48].

Table 6 summarizes the performance of the model testing for the detection of lung lesions with a validation accuracy rate of 97.25%, for model testing performed using the traditional DenseNet-121 model. The validation accuracy rate was 98.88% and 98.97% for the model testing performed using DenseNet-121 combined with Mish and Nadam; this model gives the best performance for the detection of lung lesions and is better than the traditional DenseNet-121 model. Our research results improved the optimization of CNN models in the areas of each optimizer parameter, such as learning rate and activation function, which improved the performance efficiency of the CNN model to predict the lung disease from chest X-ray images. The results for time consumption under the different optimization methods showed the lowest time for MobileNet with RMSprop, at 55 min 47 sec; the second-lowest time was accomplished by MobileNet with Adadelta, at 56 min 20 sec, and the third-lowest time was found for MobileNet with Adagrad, at 56 min 36 sec. The contributions of using CNN models to predict lung lesions from chest X-ray images include assisting the doctor in reducing diagnostic time for detection and minimizing the errors in detecting lung lesions from chest X-ray images by choosing a suitable CNN structure for the chest X-ray dataset. With many numerous chest X-ray images, CNN models can be used to recognize image features. lung disease images can also be distorted by a scoliosis spinal condition.

For future studies, researchers should use deep learning to classify more sophisticated images. There are three patterns that can help develop this direction: the education of art and culture through the classification of artifacts; the development of agricultural business and economics through an evaluation of soil quality for planting economic crops with an analysis of plant leaf diseases; assisting in medical 3D organ simulation; and in the agricultural and food industries through the detection of cancer cells in humans.

Author Contributions: Conceptualization, K.S. and C.-F.T.; data curation, C.-E.T. and P.W.; formal analysis, K.S.; funding acquisition, C.-F.T.; investigation, N.J. and C.-F.T.; methodology, K.S. and C.-F.T.; project administration, C.-F.T.; resources, K.S.; software, K.S.; supervision, C.-F.T.; validation, C.-E.T. and P.W.; visualization, K.S.; writing—original draft, K.S.; writing—review & editing, K.S. and C.-F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Republic of China, Taiwan, Grant Numbers MOST-107-2637-E-020-006, MOST-108-2637-E-020-003, MOST-108-2321-B-020-003 and MOST-107-2321-B-020-005.

Acknowledgments: The authors would like to express their sincere gratitude to the anonymous reviewers for their useful comments and suggestions for improving the quality of this paper; we thank the Department of Tropical Agriculture and International Cooperation, the Department of Management Information Systems, the National Pingtung University of Science and Technology, Taiwan and the Ministry of Science and Technology, Republic of China, Taiwan, for supporting this research.

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

References

1. Maduskar, P.; Philipsen, R.H.M.M.; Melendez, J.; Scholten, E.; Chanda, D.; Ayles, H.; Sánchez, C.I.; Ginneken, B.V. Automatic detection of pleural effusion in chest radiographs. *Med. Image Anal.* **2016**, *28*, 22–32. [CrossRef] [PubMed]
2. Sivaramakrishnan, R.; Antani, S.; Candemir, S.; Xue, Z.; Abuya, J.; Kohli, M.; Alderson, P.; Thoma, G. Comparing deep learning models for population screening using chest radiography. In Proceedings of the Spie Medical Imaging 2018: Computer-Aided Diagnosis, Houston, TX, USA, 12–15 February 2018; pp. 1–11.
3. Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep learning applications in medical image analysis. *IEEE Access* **2018**, *6*, 9375–9389. [CrossRef]
4. Hwang, S.; Kim, H.-E.; Jeong, J.; Kim, H.-J.; Inc, L. A novel approach for tuberculosis screening based on deep convolutional neural networks. In Proceedings of the Spie Medical Imaging 2016: Computer-Aided Diagnosis, San Diego, CA, USA, 28 February–2 March 2016; pp. 1–8.
5. Hogeweg, L.; Clara, I.S.; Maduskar, P.; Philipsen, R.; Story, A.; Dawson, R.; Theron, G.; Dheda, K.; Peters-Bax, L.; Ginneken, B.V. Automatic detection of tuberculosis in chest radiographs using a combination of textural, focal, and shape abnormality analysis. *IEEE Trans. Med. Imaging* **2015**, *34*, 1–19. [CrossRef] [PubMed]
6. Stirenko, S.; Kochura, Y.; Alienin, O.; Rokovyi, O.; Gordienko, Y. Chest X-ray analysis of tuberculosis by deep learning with segmentation and augmentation. In Proceedings of the 2018 IEEE 38th International Conference on Electronics and Nanotechnology, Kiev, Ukraine, 24–26 April 2018; pp. 422–428.
7. Doi, K.; MacMahon, H.; Giger, M.L.; Hoffmann, K.R. *Computer-Aided Diagnosis and Its Potential Impact on Diagnostic Radiology*, 1st ed.; Elsevier: Chicago, USA, 1999; pp. 11–20.
8. Mazurowski, M.A.; Buda, M.; Saha, A.; Bashir, M.R. Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI. *J. Magn. Reson. Imaging* **2019**, *49*, 939–954. [CrossRef] [PubMed]
9. Chi, J.; Walia, E.; Babyn, P.; Wang, J.; Groot, G.; Eramian, M. Thyroid nodule classification in ultrasound images by fine-tuning deep convolutional neural network. *J. Digit. Imaging* **2017**, *30*, 477–486. [CrossRef] [PubMed]
10. Abtahi, M.; Amiri, A.M.; Byrd, D.; Mankodiya, K. Hand motion detection in fNIRS neuroimaging data. *Healthcare* **2017**, *5*, 20. [CrossRef] [PubMed]
11. Erickson, B.J.; Korfiatis, P.; Kline, T.L.; Akkus, Z.; Philbrick, K.; Weston, A.D. Deep learning in radiology: Does one size fit all? *J. Am. Coll. Radiol.* **2018**, *15*, 1–12. [CrossRef]
12. Taylor, A.; Sen, M. Assessment of the impact of deformable registration of diagnostic MRI to planning CT on GTV delineation for radiotherapy for oropharyngeal carcinoma in routine clinical practice. *Healthcare* **2018**, *6*, 135. [CrossRef]
13. Suzuki, A.; Aoki, M.; Miyagawa, K.M.; Takaya, H.; Kotani, Y.; Nakai, H.; Matsumura, N. Differential Diagnosis of Uterine Leiomyoma and Uterine Sarcoma Using Magnetic Resonance Images. *Healthcare* **2019**, *7*, 158. [CrossRef]
14. McBee, M.P.; Awan, O.A.; Colucci, A.T.; Ghobadi, C.W.; Kadom, N.; Kansagra, A.P.; Tridandapani, S.; Auffermann, W.F. Deep learning in radiology. *Acad. Radiol.* **2018**, *25*, 1472–1480. [CrossRef]
15. Faust, O.; Hagiwara, Y.; Hong, T.J.; Lih, O.S.; Acharya, U.R. Deep learning for healthcare applications based on physiological signals: A review. *Comput. Methods Programs Biomed.* **2018**, *161*, 1–13. [CrossRef] [PubMed]

16. Milletari, F.; Ahmadi, S.; Kroll, C.; Plate, A.; Rozanski, V.; Maiostre, J.; Levin, J.; Dietrich, O.; Ertl-Wagner, B.; Bötzel, K.; et al. Hough-CNN: Deep learning for segmentation of deep brain regions in MRI and ultrasound. *Comput. Vis. Image Underst.* **2017**, *164*, 92–102. [CrossRef]
17. Lee, S.; Lee, Y.H. Improving emergency department efficiency by reinforcement learning. *Healthcare* **2020**, *8*, 77. [CrossRef] [PubMed]
18. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. *arXiv* **2017**, arXiv:1707.07012.
19. Silva, F.L.D.; Costa, A.H.R. A survey on transfer learning for multiagent reinforcement learning systems. *J. Artif. Intell. Res.* **2019**, *64*, 645–703. [CrossRef]
20. Wang, C.; Elazab, A.; Jia, F.; Wu, J.; Hu, Q. Automated chest screening based on a hybrid model of transfer learning and convolutional sparse denoising autoencoder. *BioMed Eng. Online* **2018**, *17*, 1–19. [CrossRef]
21. Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; Summers, R.M. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *arXiv* **2017**, arXiv:1705.02315v5.
22. Wang, A.G.H.W.; Weyand, M.Z.T.; Andreetto, B.C.M.; Adam, D.K.H. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861v1.
23. Gao, Z.; Zhang, Y.; Li, Y. Extracting features from infrared images using convolutional neural networks and transfer learning. *Infrared Phys. Technol.* **2020**, *105*, 103237. [CrossRef]
24. Muñoz-martínez, F.; Abellán, J.L.; Acacio, M.E. InsideNet: A tool for characterizing convolutional neural networks. *Future Gener. Comput. Syst.* **2019**, *100*, 298–315. [CrossRef]
25. Francis, L.M.; Sreenath, N. Live detection of text in the natural environment using convolutional neural network. *Future Gener. Comput. Syst.* **2019**, *98*, 444–455. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Liu, S.; Tian, G.; Xu, Y. InsideNet: A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter. *Neurocomputing* **2019**, *338*, 191–206. [CrossRef]
28. Liu, Y.; Zhang, X.; Cai, G.; Chen, Y.; Yun, Z.; Feng, Q.; Yang, W. Automatic delineation of ribs and clavicles in chest radiographs using fully convolutional DenseNets. *Comput. Methods Programs Biomed.* **2019**, *180*, 105014. [CrossRef] [PubMed]
29. Liu, Q.; Xiang, X.; Qin, J.; Tan, Y.; Tan, J.; Luo, Y. Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. *Knowl. Based Syst.* **2020**, *192*, 105375. [CrossRef]
30. Soullard, Y.; Tranouez, P.; Chatelain, C.; Nicolas, S.; Paquet, T. Multi-scale gated fully convolutional DenseNets for semantic labeling of historical newspaper images. *Pattern Recognit. Lett.* **2020**, *131*, 435–441. [CrossRef]
31. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
32. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2020**, arXiv:0580v1.
33. Grimstad, B.; Andersson, H. ReLU networks as surrogate models in mixed-integer linear programs. *Comput. Chem. Eng.* **2019**, *131*, 106580. [CrossRef]
34. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681.
35. Fan, L.; Zhang, T.; Zhao, X.; Wang, H.; Zheng, M. Deep topology network: A framework based on feedback adjustment learning rate for image classification. *Adv. Eng. Inform.* **2019**, *42*, 100935. [CrossRef]
36. Peng, Y.; Kondo, N.; Fujiura, T.; Suzuki, T.; Yoshioka, H.; Itoyama, E. Classification of multiple cattle behavior patterns using a recurrent neural network with long short-term memory and inertial measurement units. *Comput. Electron. Agric.* **2019**, *157*, 247–253. [CrossRef]
37. DeLancey, E.R.; Simms, J.F.; Mahdianpari, M.; Brisco, B.; Mahoney, C.; Kariyeva, J. Comparing deep learning and shallow learning for large-scale wetland classification in alberta, Canada. *Remote Sens.* **2019**, *12*, 2. [CrossRef]
38. Karargyris, A.; Siegelman, J.; Tzortzis, D.; Jaeger, S.; Candemir, S.; Xue, Z.; Santosh, K.C.; Vajda, S.; Antani, S.; Folio, L.; et al. Combination of texture and shape features to detect pulmonary abnormalities in digital chest X-rays. *Int. J. Comput. Assist. Radiol. Surg.* **2016**, *11*, 99–106. [CrossRef]

39. Mishra, S.; Rastogi, Y.P.; Jabin, S.; Kaur, P.; Amir, M.; Khatun, S. A deep learning ensemble for function prediction of hypothetical proteins from pathogenic bacterial species. *Comput. Biol. Chem.* **2019**, *83*, 107147. [CrossRef] [PubMed]
40. Liu, D.; Jiang, Y.; Pei, M.; Liu, S. Emotional image color transfer via deep learning. *Pattern Recognit. Lett.* **2018**, *110*, 16–22. [CrossRef]
41. Arena, P.; Basile, A.; Bucolo, M.; Fortuna, L. Image processing for medical diagnosis using CNN. *Nucl. Instrum. Methods Phys. Res.* **2003**, *497*, 174–178. [CrossRef]
42. Zhuang, X.; Mastorakis, N.E. Matching of images with rotation transformation based on the virtual electromagnetic interaction. *Wseas Trans. Comput.* **2015**, *14*, 679–697.
43. Sajn, L.; Kukar, M. Image processing and machine learning for fully automated probabilistic evaluation of medical images. *Comput. Methods Programs Biomed.* **2011**, *104*, 75–86. [CrossRef]
44. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant disease detection and classification by deep learning. *Plants* **2019**, *8*, 468. [CrossRef]
45. Kamilaris, A.; Prenafeta-boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
47. Kim, H.; Kim, H.S.; Moon, E.S.; Yoon, C.-S.; Chung, T.-S.; Song, H.-T.; Suh, J.-S.; Lee, Y.H.; Kim, S. Scoliosis imaging: What radiologists. *Radio Graph.* **2010**, *30*, 1823–1842.
48. Yang, C.; Li, Y.; Zhao, Y.; Zhu, X. Adult degenerative scoliosis: Can cobb angle on a supine posteroanterior radiograph be used to predict the cobb angle in a standing position? *Medicine* **2016**, *95*, e2732. [CrossRef] [PubMed]
49. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
50. Qayyum, A.; Muhammad, S.; Awais, M.; Majid, M. Medical image retrieval using deep convolutional neural network. *Neurocomputing* **2017**, *266*, 8–20. [CrossRef]
51. Kc, K.; Yin, Z.; Wu, M.; Wu, Z. Depthwise separable convolution architectures for plant disease classification. *Comput. Electron. Agric.* **2019**, *165*, 1–6. [CrossRef]
52. Zhao, H.; Liu, F.; Liang, H.Z.Z. Research on a learning rate with energy index in deep learning. *Neural Netw.* **2019**, *110*, 225–231. [CrossRef]
53. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2017**, arXiv:1609.04747.
54. Becherer, N.; Pecarina, J.; Nykl, S.; Hopkinson, K. Improving optimization of convolutional neural networks through parameter fine-tuning. *Neural Comput. Appl.* **2019**, *31*, 3469–3479. [CrossRef]
55. Hosseini, H.; Xiao, B.; Jaiswal, M.; Poovendran, R. On the limitation of convolutional neural networks in recognizing negative images. *arXiv* **2017**, arXiv:1703.06857.
56. Dolz, J.; Desrosiers, C.; Ayed, I.B. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage* **2018**, *170*, 456–470. [CrossRef] [PubMed]
57. Tummala, M. Image classification using convolutional neural networks. *Int. J. Sci. Res. Publ.* **2019**, *9*, 382–385. [CrossRef]
58. Chen, Y.; Lin, Z.; Zhao, X.; Member, S.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
59. Lopes, U.K.; Valiati, J.F. Pre-trained convolutional neural networks as feature extractors for tuberculosis detection. *Comput. Biol. Med.* **2017**, *89*, 135–143. [CrossRef]