Using Generative Adversarial Networks and Parameter Optimization of Convolutional Neural Networks for Lung Tumor Classification

Chun-Hui Lin 1, Cheng-Jian Lin 2,3,*, Yu-Chi Li 2 and Shyh-Hau Wang 1,4

1 Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan; charlinehalu@gmail.com (C.-H.L.); shyhhau@mail.ncku.edu.tw (S.-H.W.)
2 Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan; nancy19960222@gmail.com
3 College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan
4 Intelligent Manufacturing Research Center, National Cheng Kung University, Tainan 701, Taiwan
* Correspondence: cjlin@ncut.edu.tw

Abstract: Cancer is the leading cause of death worldwide. Lung cancer, especially, caused the most death in 2018 according to the World Health Organization. Early diagnosis and treatment can considerably reduce mortality. To provide an efficient diagnosis, deep learning is overtaking conventional machine learning techniques and is increasingly being used in computer-aided design systems. However, a sparse medical data set and network parameter tuning process cause network training difficulty and cost longer experimental time. In the present study, the generative adversarial network was proposed to generate computed tomography images of lung tumors for alleviating the problem of sparse data. Furthermore, a parameter optimization method was proposed not only to improve the accuracy of lung tumor classification, but also reduce the experimental time. The experimental results revealed that the average accuracy can reach 99.86% after image augmentation and parameter optimization.

Keywords: lung cancer; convolutional neural network; generative adversarial network; image augmentation; parameter optimization

1. Introduction

According to a report from the World Health Organization in 2018, there were about 9.6 million deaths from cancer globally, of which 1.76 million cases were attributed to lung cancers [1]. Studies have identified environmental factors and smoking as major causes of lung cancer [2]. Generally, chest X-ray, computed tomography (CT) and magnetic resonance imaging are modalities used to evaluate lung cancer [3,4]. The chest X-ray is the first test in diagnosing lung cancer. It indicates abnormal formations in the lungs. Compared to a chest X-ray, a CT scan can show a more detailed view of the lungs and can also show the exact shape, size, and location of formations. A CT scan is therefore a major diagnostic tool for the assessment of lung cancer. To reduce the workload of analyzing CT images manually and to avoid subjective interpretations, machine learning techniques are applied to computer-aided design systems for objectively auxiliary diagnosis. Lately, due to the rapid growth of deep learning, convolutional neural networks (CNNs) not only show good performance in image classification and object detection tasks [5–7], but are also widely used in several applications, including smart homes, driverless cars, manufacturing robots, drones, and chat robots. The studies of CNNs have been continually innovating and improving.

In 1998, LeCun et al. proposed LeNet-5 [8], a simple CNN for handwritten digits classification. The LeNet-5 comprises a feature extraction part convolutional layers and pooling layers and a classification part fully connected layer. Subsequently, in 2012, Krizhevsky et al.
proposed AlexNet [9] and won the ImageNet Large Scale Visual Recognition Competition. AlexNet replaces Sigmoid and Tanh activation function with a rectified linear unit. It also introduces Dropout and max pooling which are different from LeNet-5. In 2014, Szegedy et al. proposed GoogLeNet [10] and proposed Inception module which uses three different size of convolutional kernels simultaneously for extracting more features in one layer. In the same year, Simonyan et al. proposed VGGNet model [11]. VGGNet adopts $3 \times 3$ stacked convolutional layers and deepening the depth of the network as well as the number of input and output channel of layers. In 2015, He et al. proposed ResNet [12] and introduced a residual block alleviating the degradation problem of the deep network. Certainly, more and more architectures are innovated. However, unbalanced or sparse data sets and network parameter settings for training are two major problems faced by deep learning.

Unbalanced data, especially medical images, is one of the most challenging tasks in deep learning [13–17]. Typical data augmentation methods include translation, rotation, flipping, and zooming [18,19]. However, those geometric transformations might not be able to provide sufficient data diversity. In 2014, generative adversarial networks (GANs) [20] were proposed to tackle the problem of sparse data. This model consists of two networks: one generator network and one discriminator network. The generator network aims to generate plausible fake images. On the other hand, the discriminator network distinguishes real data from the data created by the generator or real data and acts as a classifier. In 2015, deep convolutional GANs (DCGANs), a direct extension of GANs, were proposed [21] which replaced the original convolutional layers by transposed convolutional layers. Later, many studies discussed the complementary data process techniques in medical applications. Perez et al. [22] investigated the impact of 13 data augmentation scenarios, such as traditional color and geometric transforms, elastic transforms, random erasing, and lesion mixing method for melanoma classification. The results confirmed that data augmentation can lead to more performance gains than obtaining new images. Madani et al. [23] implemented GANs for producing chest X-ray images to augment a dataset and showed higher accuracy for normal vs abnormal classification in chest X-rays.

In the other hand, selecting a better network parameter combination is another time-consuming task. Several experiments are required for determining the optimum parameter combination. To reduce the time cost, many studies have been proposed for network parameter optimization methods. Real et al. [24] introduced genetic algorithm into CNN architecture and achieved high accuracy in both CIFAR-10 and CIFAR-100 data sets. An autonomous and continuous learning algorithm proposed by Ma et al. [25] could automatically generate deep convolutional neural network (DCNN) architectures by partition DCNN into multiple stacked meta convolutional blocks and fully connected blocks then used genetic evolutionary operations to evolve a population of DCNN architectures. Although those methods showed high accuracy, they are still time consuming. The Taguchi method proposed by Dr. Genichi Taguchi has been widely applied as a design method [26–28]. It is not only straightforward and easy to implement in many engineering situations but also able to narrow down the scope of a research project quickly.

In the present study, the main contributions are to alleviate the problem of sparse medical images and to use a parameter optimizer to select an optimal network parameter combination in fewer experiments based on the state-of-art CNNs for providing an accurate and a general applicable lung tumor classification. Firstly, GAN was introduced to augment CT images in order to increase the data diversity for improving the accuracy of CNNs and AlexNet architecture was chosen as the backbone classification network with a parameter optimizer which is capable to select a better parameter combination in fewer experiments for achieving the goals of the present study. The rest of this paper is organized as follows. Section 2 describes a data augmentation method to increase lung tumor CT images. Section 3 reviews CNN architecture and introduces the network parameter optimizer. The experimental results and discussions are detailed in Section 4. Section 5 draws the conclusion.
2. Data Augmentation Using GANs

Training a multilayer CNNs using a limited number of data sets results in overfitting. To avoid overfitting data augmentation is a technique used to increase the amount of data in training processing. Typical data augmentation techniques include cropping, flipping, rotation, and translation, yet those methods are lacking data diversity. GAN, an innovation network proposed in 2014, was applied to generate new data automatically. It trains the generator and discriminator networks simultaneously. The former generates new images, and the latter learns to distinguish the fake images from the input of real and generated data. The two networks continually generate and discriminate tasks and constantly update the parameters of the network. Finally, the training process terminates when the generator network deceives the discriminator network. Figure 1 shows the flowchart of GANs.

![Figure 1. Flowchart of GANs.](image)

In the present study, DCGAN was applied for data augmentation which the discriminator network uses convolutional stride for downsampling and the generator network uses transposed convolution for upsampling. The details of DCGAN are described as follows.

2.1. Generator Network

The generator network learns characteristics from real images. First, a $1 \times 1 \times 100$ noise array is converted into a $7 \times 7 \times 128$ array by reshape and projection layers. The deconvolution (DC) layer, batch normalization, and ReLU activation function are performed to obtain a $64 \times 64 \times 3$ image. Figure 2 displays the flowchart of the generator network. Table 1 lists the details of the generator network parameters.

![Figure 2. Flowchart of the generator network.](image)

**Table 1.** Details of the generator network architecture.

| Layer | Filter Size | Number of Filters | Stride |
|-------|-------------|-------------------|--------|
| DC 1  | 5 × 5       | 512               | 1      |
| DC 2  | 5 × 5       | 256               | 2      |
| DC 3  | 5 × 5       | 128               | 2      |
| DC 4  | 5 × 5       | 64                | 2      |
| DC 5  | 5 × 5       | 3                 | 2      |
2.2. Discriminator Network

The discriminator network determines whether the input image is a generated image or a real image. The network takes a $64 \times 64 \times 3$ image as an input and the output is a scalar prediction score using a series of convolutional layers with batch normalization and leaky ReLU activation function. The dropout value is set to 0.5. Leaky ReLU shown in Figure 3 allocates all negative values to a nonzero slope. Figure 4 displays the flowchart of the discriminator network. Table 2 provides the details of the discriminator network parameters.

![Leaky ReLU Activation Function](image1)

Figure 3. Leaky ReLU activation function.

![Flowchart of the Discriminator Network](image2)

Figure 4. Flowchart of the discriminator network.

Table 2. Details of the discriminator network architecture.

| Layer | Filter Size | Number of Filters | Stride | Padding |
|-------|-------------|-------------------|--------|---------|
| C 1   | $5 \times 5$ | 64                | 2      | 1       |
| C 2   | $5 \times 5$ | 128               | 2      | 1       |
| C 3   | $5 \times 5$ | 256               | 2      | 1       |
| C 4   | $5 \times 5$ | 512               | 2      | 1       |
| C 5   | $5 \times 5$ | 1                 | 1      | 0       |

3. CNN Architecture and Parameter Optimizer

This section reviews a CNN architecture and describes how CNN parameters can be adjusted by using the parameter optimizer. Figure 5 illustrates the flowchart of parameter optimization process.
3. CNN Architecture and Parameter Optimizer

This section reviews a CNN architecture and describes how CNN parameters can be adjusted by using the parameter optimizer. Figure 5 illustrates the flowchart of parameter optimization process.

Figure 5. Parameters optimization process using Taguchi method.

3.1. CNNs

CNNs are the most commonly modalities used for image recognition and usually consist of three parts: convolutional, pooling, and fully connected (FC) layers. The convolutional and pooling layers are the most crucial parts for extracting global and local features.

3.1.1. Convolutional Layer

The convolutional layer (C) contains several kernels which are used to extract features from images. Each convolutional layer is covered by kernels with various weight combinations. The kernel performs convolution operations through a sliding approach to generate feature maps. Then, the inner product between the input kernels at each spatial position is calculated. Finally, the output of the convolutional layer is obtained by stacking the feature maps of all kernels in the depth direction.

3.1.2. Pooling Layer

The objective of using a pooling layer (Pool) is to reduce the size of feature maps without losing important feature information and reduce subsequent operations. Pooling can be performed using several methods, including average and max pooling. Average pooling calculates the average value within the selected patch from the feature map. Contrarily, max pooling calculates the maximum value within the selected patch from the feature map. In addition, padding (P) is seldom applied in the pooling layer. Also, the pooling layer does not generate trainable variables.

3.1.3. Activation Function

In neural networks, each neuron is connected to other neurons in order to passing the signal from an input layer to an output layer in one direction. The activation layer relates to the forward propagation of the signal through the network. The purpose of the activation function is to substitute the nonlinear function into the output of the neuron to solve complex nonlinear problems. Sigmoid, tanh, and ReLU are common activation functions, with ReLU being among the most widely used. ReLU, as expressed in Equation (1), is also used as an activation function for addressing the vanishing gradient problem and it can reduce the degree of overfitting, as displayed in Figure 6.

\[
ReLU(a) = \max(0, a),
\] (1)
3.1.4. Fully Connected Layer

The fully connected (FC) layer is functioned as a classifier. The FC layer converts the two-dimensional feature map output by the convolution layer into a one-dimensional vector. The final probability of each label is obtained using Softmax.

LeNet-5 and AlexNet contain fewer layers and simple architecture compared with other deeper CNNs. Among them, AlexNet has not only been presented good performance in many applications, but also allows color images as input, such as computed tomography images. Therefore, with data augmentation and parameter optimizer implementation, AlexNet might be a suitable network architecture used in this study. AlexNet consists of five convolutional layers, three pooling layers, three FC layers, and Softmax with 1000 outputs. The aim of this study was to classify lung CT images into benign or malignant tumors. Thus, the transfer learning technique was applied to change the last FC layer to two outputs. The AlexNet architecture is illustrated in Figure 7 and Table 3 lists the details of the AlexNet.

Table 3. Details of the AlexNet architecture.

| Layer | Feature Map Size | Kernel Size | Stride | Activation Function |
|-------|------------------|-------------|--------|---------------------|
| Layer 1 | C 55 × 55 × 96 | 11 | 4 | ReLU |
| Pool | | 27 × 27 × 96 | 3 | 2 | ReLU |
| Layer 2 | C 27 × 27 × 96 | 5 | 1 | ReLU |
| Pool | | 13 × 13 × 256 | 3 | 2 | ReLU |
| Layer 3 | C 13 × 13 × 384 | 3 | 1 | ReLU |
| Layer 4 | C 13 × 13 × 384 | 3 | 1 | ReLU |
| Layer 5 | C 13 × 13 × 256 | 3 | 1 | ReLU |
| Pool | | 6 × 6 × 256 | 3 | 2 | ReLU |
| Layer 6 | FC 9216 | - | - | ReLU |
| Layer 7 | FC 4096 | - | - | ReLU |
| Layer 8 | FC 4096 | - | - | ReLU |
| Output | - | 2 | - | Softmax |

Figure 6. ReLU activation function.
Selecting an optimal network parameter combination is a time-consuming task. In this study, the objective is to investigate the performance of CNNs using parameter optimization. The Taguchi method is a low-cost, high-efficiency quality engineering method that emphasizes improving product quality through design experiments. Therefore, the Taguchi method was applied for the parameter optimization of CNNs.

First, the objective function is defined. Then, the factors and levels that affect the objective function are selected. The orthogonal array and the signal-to-noise ratio (S/N ratio) are the two main indicators in the Taguchi method. The orthogonal array is used to determine the number of times the experiment needs and allocate experimental factors into an orthogonal array. Additionally, the S/N ratio is used to verify whether the CNN parameters are the optimal parameter combination. Finally, according to the experimental results the optimal key factors and levels are decided. Although the cost-effectiveness of the experiment is an issue, the optimal combination of factors and levels can be found. Figure 8 displays the flowchart of the Taguchi method.

Figure 7. AlexNet architecture.

3.2. Parameter Optimization

Figure 8. Flowchart of the Taguchi method.
• **First step:**

Understand the task to be completed. Here, the CNN parameters, including kernel size (KS), stride (S), and padding (P), were tasks that needed to be optimized in order to achieve higher accuracy in fewer experiments.

• **Second step:**

Select factors and levels. In AlexNet, the first convolutional layer involves global feature extraction, and the fifth convolutional layer involves local feature extraction of the input image. Therefore, KS, S, and P of the first and fifth convolutional layers were adjusted by Taguchi method. The factors are: kernel size (C1-KS), stride (C1-S), and padding (C1-P) of the first convolutional layer, kernel size (C5-KS), stride (C5-S), and padding (C5-P) of the fifth convolutional layer. The levels are assigned according to the parameters commonly used in the state-of-art CNNs as shown in Table 4.

**Table 4.** Details of factors and levels.

| Factor | Level 1 | Level 2 | Level 3 |
|--------|---------|---------|---------|
| C1     | KS      | 13      | 11      | 9       |
|        | S       | 4       | 3       | 2       |
|        | P       | 2       | 1       | -       |
| C5     | KS      | 7       | 5       | 3       |
|        | S       | 3       | 2       | 1       |
|        | P       | 2       | 1       | 0       |

• **Third step:**

Choose an appropriate orthogonal array. The orthogonal array provides statistical information with fewer experiments. After the factors and levels selection, the appropriate orthogonal array should be chosen based on the factors and the levels. In this study, C1-P had two levels, and C1-KS, C1-S, C5-KS, C5-S, and C5-P had three levels. The total degree of freedom in the experiment is 11, therefore, the $L_{18}$ orthogonal array is selected. Initially, the selected factors and levels required 486 ($3 \times 3 \times 2 \times 3 \times 3 \times 3$) experiments, while using the orthogonal array the scope of experiments was reduced to only 18 experiments.

• **Fourth step:**

Fill in the $L_{18}$ orthogonal array with the factors and levels designed in Table 4. The complete $L_{18}$ orthogonal array is presented in Table 5.

**Table 5.** $L_{18}$ Orthogonal array.

|      | C1-KS | C1-S | C1-P | C5-KS | C5-S | C5-P |
|------|-------|------|------|-------|------|------|
| 1    | 9     | 4    | 2    | 5     | 3    | 0    |
| 2    | 9     | 3    | 2    | 3     | 2    | 2    |
| 3    | 9     | 2    | 2    | 7     | 1    | 1    |
| 4    | 13    | 4    | 2    | 7     | 3    | 2    |
| 5    | 13    | 3    | 2    | 5     | 2    | 1    |
| 6    | 13    | 2    | 2    | 3     | 1    | 0    |
| 7    | 11    | 4    | 2    | 7     | 2    | 1    |
| 8    | 11    | 3    | 2    | 5     | 1    | 0    |
| 9    | 11    | 2    | 2    | 3     | 3    | 2    |
| 10   | 11    | 4    | 1    | 5     | 1    | 2    |
| 11   | 11    | 3    | 1    | 3     | 3    | 1    |
| 12   | 11    | 2    | 1    | 7     | 2    | 0    |
| 13   | 13    | 4    | 1    | 3     | 1    | 1    |
| 14   | 13    | 3    | 1    | 7     | 3    | 0    |
| 15   | 13    | 2    | 1    | 5     | 2    | 2    |
| 16   | 9     | 4    | 1    | 3     | 2    | 0    |
| 17   | 9     | 3    | 1    | 7     | 1    | 2    |
| 18   | 9     | 2    | 1    | 5     | 3    | 1    |
Fifth step:
Perform 18 experiments based on the $L_{18}$ orthogonal array. In this study, each experiment was tested five times to get an overall accuracy.

Sixth step:
Calculate the S/N ratio and analyze the experimental data.

Seventh step:
Accurate classification of lung tumor images is the purpose of this study. Hence, a higher S/N ratio indicates that the parameter combination is optimal and is able to provide superior performance.

Eighth step:
Finally, use the acquired optimal parameter combination to train AlexNet again to verify that the optimal parameter combination is able to improve the accuracy of this network.

4. Experimental Results

All experiments were implemented using MATLAB software on a personal computer (Intel Xeon processor E3-1225 v5; processor speed, 3.30 GHz; GTX 1080 graphics processor unit).

4.1. SPIE-AAPM Lung CT Challenge Data Set

The SPIE-AAPM Lung CT Challenge data set [29] was first presented at the Medical Imaging conference in 2015 and supported by the American Society of Medical Physics (AAPM) and the National Cancer Institute. It contains 22,489 lung CT images, with 11,407 images of malignant tumors and 11,082 images of benign tumors. The size of each image is $512 \times 512$ pixels. Figure 9 displays the CT images of malignant and benign tumors.

![Figure 9. CT images of (a) malignant and (b) benign tumors.](image)

4.2. Experiment 1: Data Augmentation

All lung tumor images were classified using AlexNet and the training parameters of AlexNet and GAN listed in Table 6 were chosen by the user experiences based on MATLAB official default settings. In order to avoid producing confusing images, malignant and benign tumor images were generated separately. Figure 10 displays the generated images.
Table 6. Training parameters of AlexNet and GAN.

|                  | AlexNet           | GAN               |
|------------------|-------------------|-------------------|
| Training Epochs  | 2                 | 100               |
| Mini Batch Size  | 15                | 100               |
| Optimizer Type   | SGDM              | Learning Rate     |
| Base Learning Rate | 0.0003           | Generator Network | 0.0002            |
|                  |                   | Discriminator Network | 0.0001     |

Figure 10. Generated (a) malignant and (b) benign tumor images.

The generated images were mixed into the original image data set for lung tumor identification. Thereby, 70% of mixed images were training data and 30% of mixed images were testing data, as presented in Figure 11.

Figure 11. Flowchart of classifying mixed data set.

Figure 12 displays the number of the mixed images and the accuracy of experiments. The accuracy improves when the number of images is increased. Table 7 lists the number of mixed images and Table 8 presents the accuracy, specificity, and sensitivity of lung tumor classification. With data augmentation, both accuracy and sensitivity improved from 97.48% to 98.42% and from 95.10% to 99.40%, respectively.
4.3. Experiment 2: Verification of Generated Image

To verify the plausible generated images, 30% of the original images were reserved as the validation data at the beginning. The remaining 70% of the original images were mixed with generated images as training data. The flowchart of verification process is presented in Figure 13.

![Flowchart of generated images verification process](image_url)

**Figure 13.** Flowchart of generated images verification process.

**Table 7.** Number of mixed images.

| Size of Data Set   | Original | Double | Triple | Quadruple |
|--------------------|----------|--------|--------|-----------|
| Total image        | 22,489   | 44,978 | 67,467 | 89,956    |
| Benign image       | 11,082   | 22,164 | 33,246 | 44,328    |
| Malignant image    | 11,407   | 22,814 | 34,221 | 45,628    |

**Table 8.** Accuracy, specificity, and sensitivity of mixed data set.

| Size of Data Set   | Original | Double | Triple | Quadruple |
|--------------------|----------|--------|--------|-----------|
| Accuracy           | 97.48%   | 97.71% | 98.39% | 98.42%    |
| Sensitivity        | 95.10%   | 95.50% | 97.30% | 99.40%    |
| Specificity        | 99.80%   | 99.90% | 97.40% | 99.50%    |

**Figure 12.** Number of mixed images and accuracy of experiments.

**Figure 14.** Displays the number of images and the accuracy of verification. The number of images after augmentation is presented in Table 9, and Table 10 displays the accuracy, specificity, and sensitivity of verification. The experimental results reveal that the...
Figure 14 displays the number of images and the accuracy of verification. The number of images after augmentation is presented in Tables 9 and 10 displays the accuracy, specificity, and sensitivity of verification. The experimental results reveal that the accuracy and sensitivity improved after the original data set was augmented. The highest accuracy rate reached 99.60%, and the highest sensitivity was 99.80% in other words, the generated images are able to be trusted to solve the problem of sparse medical images.

![Image showing the number of images and the accuracy of verification](image)

**Figure 14.** Number of images and the accuracy of verification process.

**Table 9.** Number of images after augmentation.

| Size of Data Set    | Original | Double | Triple | Quadruple |
|--------------------|----------|--------|--------|-----------|
| Total image        | 15,742   | 31,484 | 47,226 | 62,968    |
| Benign image       | 7757     | 15,514 | 23,271 | 31,028    |
| Malignant image    | 7985     | 15,970 | 23,955 | 31,940    |

**Table 10.** Accuracy, specificity, and sensitivity of generated image verification.

| Size of Data Set    | Original | Double | Triple | Quadruple |
|--------------------|----------|--------|--------|-----------|
| Accuracy           | 96.87%   | 99.18% | 99.60% | 99.60%    |
| Sensitivity        | 94.30%   | 98.80% | 99.30% | 99.80%    |
| Specificity        | 99.30%   | 99.60% | 99.90% | 99.40%    |

From Figures 12 and 14, it can be noticed that increasing the size of data into quadruple does not show significant accuracy improvement. The reason might be that the images diversity is sufficient for the network to learn the features of lung tumors when increasing the size of data into triple. Moreover, the accuracy in experiment 2 reached 99.6% which is higher than that of in experiment 1 might be the reason that the goal of conducing experiment 2 was to verify the generated images which contain noises in order to train the network more diversity. Therefore, the generated images are more appropriate to help network extracting different features but not for testing. Moreover, the sensitivity is another important index evaluation for the network, especially in medical applications. From those experiments, after data augmentation, the highest sensitivity achieved is 99.8%.

### 4.4. Using Parameter Optimization in Experiment 1

In parameter optimization, 18 parameter combinations were conducted through the orthogonal array, and each experiment was repeated five times. The training parameters of AlexNet are presented in Table 11. Table 12 lists the five observations and the S/N ratio.
Table 11. Training parameters of AlexNet.

| Training Epochs | Mini Batch Size | Optimizer Type | Base Learning Rate |
|-----------------|-----------------|----------------|--------------------|
| 2               | 15              | SGDM           | 0.0003             |

Table 12. Five times observations and the S/N ratio.

| Observations | Accuracy (%) | Average Accuracy (%) | S/N Ratio |
|--------------|--------------|-----------------------|-----------|
| 1            | 99.94        | 98.61                 | 99.82     | 99.65 | -0.0308 |
| 2            | 95.97        | 96.66                 | 94.95     | 98.8  | 96.72  | -0.2923 |
| 3            | 99.96        | 98.04                 | 94.95     | 99.97 | 99.93  | 0.0469  |
| 4            | 99.24        | 94.84                 | 91.12     | 98.16 | 97.45  | 0.3522  |
| 5            | 99.99        | 99.96                 | 100       | 99.93 | 99.99  | -0.0023 |
| 6            | 99.96        | 99.99                 | 100       | 99.91 | 99.97  | -0.0024 |
| 7            | 97.38        | 97.18                 | 98.41     | 94.31 | 99.1   | 0.2437  |
| 8            | 99.33        | 97.2                  | 99.85     | 99.61 | 99.18  | 0.0855  |
| 9            | 99.64        | 91.57                 | 99.93     | 99.26 | 98.32  | 0.2125  |
| 10           | 99.97        | 100                   | 99.63     | 99.97 | 100    | 0.0075  |
| 11           | 85.58        | 83.66                 | 82.75     | 82.14 | 88.45  | 84.52   | -1.4705 |
| 12           | 100          | 99.85                 | 99.96     | 100   | 99.96  | 0.0033  |
| 13           | 100          | 99.97                 | 100       | 100   | 99.99  | -0.0005 |
| 14           | 95.27        | 98.03                 | 91.71     | 88.75 | 94.1   | 93.57   | -0.5920 |
| 15           | 69.98        | 76.42                 | 84.03     | 75.27 | 85.83  | 78.31   | -2.1974 |
| 16           | 97.6         | 99.75                 | 98.74     | 93.86 | 99.07  | 97.80   | -0.1990 |
| 17           | 99.75        | 99.61                 | 99.08     | 99.63 | 99.63  | 99.54   | -0.0401 |
| 18           | 100          | 99.98                 | 99.32     | 100   | 99.86  | -0.0123 |

According to the S/N ratio in Table 12, the optimal level based on each factor was analyzed and the significant factors were ranked. The results from 18 experiments were displayed in Table 13 and the best factors were the results mapping to Table 4. The best parameter combination for SPIE-AAPM data set classification is C1-KS, C1-S, C1-P, C5-KS, C5-S, C5-P.

Table 13. Results analysis from 18 experiments.

| Factor | Level 1 | Level 2 | Level 3 | Delta | Significance Rank | Best Levels | Best Factors |
|--------|---------|---------|---------|-------|-------------------|-------------|--------------|
| KS     | -0.3372 | -0.5267 | -0.1014 | 0.42528 | 3                 | 3           | 9            |
| S      | -0.1390 | -0.4138 | -0.4125 | 0.27484 | 6                 | 1           | 4            |
| P      | -0.1410 | -0.5025 | -       | 0.36157 | 4                 | 1           | 2            |
| KS     | -0.6642 | -0.1244 | -0.1767 | 0.53980 | 2                 | 2           | 5            |
| S      | -0.4529 | -0.4197 | -0.0926 | 0.36037 | 5                 | 3           | 1            |
| P      | -0.0561 | -0.1218 | -0.7873 | 0.731  | 1                 | 1           | 2            |

The highest accuracy using the best parameter combination achieved is 99.99% in this study. The accuracy is considerably higher than other networks, besides, the training time is less than those networks. The results are shown in Table 14.

Table 14. Comparison of the best parameter combination with other methods.

| Method                | Accuracy (%) | Sensitivity (%) | Specificity (%) | Training Time |
|-----------------------|--------------|-----------------|-----------------|---------------|
| AlexNet               | 97.48        | 95.1            | 99.8            | 3 min 50 s    |
| GoogLeNet             | 99.6         | 99.2            | 99.9            | 16 min 14 s   |
| VGG16                 | 99.57        | 99.2            | 99.9            | 19 min 59 s   |
| VGG19                 | 97.32        | 99.7            | 95              | 21 min 6 s    |
| ResNet18              | 99.75        | 99.3            | 99.8            | 9 min 17 s    |
| AlexNet with Taguchi method | 99.9       | 99.9            | 100             | 3 min 25 s    |
Table 15 displays the best factors based on different sizes of images. Table 16 presents a comparison of AlexNet using the Taguchi method with original AlexNet according to different data quantity. Table 16 reveals that the average accuracy improves from 97.48% to 99.49% after data augmentation and parameter optimization. In addition, the experimental results are graphically presented in Figure 15.

### Table 15. Best parameter combination of each quantity of generated images.

| Size of Data Set | Factor | C1 | S | P | KS | S | P |
|-----------------|--------|----|---|---|----|---|---|
| Original        |        | 9  | 4 | 2 | 5  | 1 | 2 |
| Double          |        | 9  | 4 | 2 | 3  | 1 | 1 |
| Treble          |        | 11 | 2 | 2 | 5  | 1 | 2 |
| Quadruple       |        | 13 | 3 | 2 | 5  | 1 | 2 |

### Table 16. Comparison of the best parameter combination with AlexNet.

| Methods | AlexNet | Best Parameter Combination |
|---------|---------|---------------------------|
| Size of Data Set | Original | Double | Triple | Quadruple | Original | Double | Triple | Quadruple |
| First (%)   | 98.74   | 97.71 | 98.94 | 98.88 | 99.13 | 99.23 | 99.18 | 99.78 |
| Second (%)  | 97.66   | 96.87 | 98.99 | 99.47 | 99.99 | 99.38 | 99.17 | 99.63 |
| Third (%)   | 97.36   | 98.1  | 98.83 | 99.41 | 99.23 | 99.39 | 99.63 | 98.84 |
| Fourth (%)  | 96.47   | 98.69 | 98.99 | 99.48 | 98.57 | 99.32 | 99.44 | 99.89 |
| Fifth (%)   | 97.18   | 97.24 | 98.4  | 99.11 | 98.21 | 98.95 | 98.96 | 99.35 |
| Average (%) | 97.48   | 97.72 | 98.83 | 99.28 | 99.03 | 99.25 | 99.28 | 99.49 |
| SD          | 0.828   | 0.714 | 0.249 | 0.265 | 0.681 | 0.181 | 0.261 | 0.420 |

Figure 15. Graph of the best parameter combination in comparison with AlexNet.

4.5. Using Parameter Optimization in Experiment 2

Table 17 presents the best factors according to each size of data set. Table 18 lists a comparison of the AlexNet with Taguchi method with original AlexNet. The accuracy increases from 97.10% to 99.86% when the data are augmented, and the parameter optimization is implemented.
Table 17. Best factors of each quantity of images in experiment 2.

| Size of Data Set | Factor | C1 | S | P | C5 | S | P |
|------------------|--------|----|---|---|----|---|---|
| Original         | KS     | 9  | 4 | 2 | 3  | 1 | 1 |
| Double           | KS     | 9  | 4 | 2 | 3  | 3 | 1 |
| Treble           | KS     | 13 | 4 | 2 | 3  | 1 | 1 |
| Quadruple        | KS     | 11 | 2 | 2 | 3  | 3 | 1 |

Table 18. Comparison of AlexNet with Taguchi method with original AlexNet.

| Methods         | AlexNet | Best Parameter Combination |
|-----------------|---------|---------------------------|
| Size of Data Set| Original| Double | Triple | Quadruple | Original| Double | Triple | Quadruple |
| First (%)       | 95.67   | 98.09  | 98.46  | 99.05     | 99.10  | 99.36  | 99.76  | 99.89     |
| Second (%)      | 98.07   | 99.51  | 99.9   | 99.35     | 99.69  | 99.35  | 99.68  | 99.93     |
| Third (%)       | 97.30   | 98.8   | 99.3   | 98.10     | 99.75  | 99.9   | 99.73  | 99.94     |
| Fourth (%)      | 97.84   | 97.48  | 97.35  | 99.88     | 99.53  | 99.87  | 99.86  | 99.80     |
| Fifth (%)       | 96.63   | 99.14  | 98.76  | 99.44     | 98.56  | 99.87  | 99.84  | 99.76     |
| Average (%)     | 97.10   | 98.8   | 98.67  | 99.16     | 99.326 | 99.67  | 99.774 | 99.86     |
| SD              | 0.974   | 0.818  | 0.849  | 0.665     | 0.498  | 0.288  | 0.075  | 0.080     |

Overall, from experiment one and two, considering the size of the data, the better augment size might be double or triple. The accuracy shows significant improvement in those sizes of data set. In addition, AlexNet with optimal parameter combination shows better accuracy and lower standard deviation, which is more stable than the original AlexNet. Although Taguchi method reduces the number of experiments, it still needs to execute multiple times. However, for medical applications, it is vital to have an accurate classification network.

5. Conclusions

An accurate lung tumor classification is a crucial role for early diagnosis. Computer aided design can considerably reduce clinicians’ workload. However, obtaining open access medical images is difficult. Therefore, the GAN was used to augment the data set to alleviate the data shortage problem. With data augmentation, the overall accuracy of the CNN improved by 2.73%. Moreover, tuning the parameters in CNN has become another issue to face nowadays. In this study, the Taguchi method was implemented for selecting optimal parameters through fewer experiments. The experimental results revealed that the accuracy of using the optimal parameter combination can reach 99.86%. The present study only discussed the lung tumor classification and the optimizer for CNNs only took three parameters in the first and fifth layers as consideration. Further research will entail clinical application and optimizer improvement, such as adjusting the parameters of each layer to obtain the best parameter combination or implementing the optimizer in different network architectures. The method can also be applied to other medical applications, such as breast, brain, and liver cancer classification.

Author Contributions: Conceptualization, C.-H.L. and C.-J.L.; Methodology, C.-J.L.; Software, C.-H.L. and Y.-C.L.; Data Curation, C.-J.L. and S.-H.W.; Writing–Original Draft Preparation, C.-H.L. and Y.-C.L.; Writing–Review & Editing, C.-J.L.; Supervision, C.-J.L. and S.-H.W.; Funding Acquisition, C.-J.L. and S.-H.W. 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 of the Republic of China, grant number MOST 109-2634-F-009-031.
Acknowledgments: The authors would like to thank the support of the Intelligent Manufacturing Research Center (iMRC) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

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

References
1. World Health Organization. A Report About Cancer. Available online: https://www.who.int/news-room/fact-sheets/detail/cancer (accessed on 12 September 2018).
2. Ferlay, J.; Soerjomataram, I.; Dikshit, R.; Eser, S.; Mathers, C.; Rebelo, M.; Parkin, D.M.; Forman, D.; Bray, F. Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012. Int. J. Cancer 2015, 136, E359–E386. [CrossRef] [PubMed]
3. Pedersen, J.H.; Ashraf, H.; Dirksen, A.; Bach, K.; Hansen, H.; Toennesen, P.; Thorsen, H.; Brodersen, J.; Skov, B.G.; Dossing, M.; et al. The Danish Randomized Lung Cancer CT Screening Trial—Overall Design and Results of the Prevalence Round. J. Thorac. Oncol. 2009, 4, 608–614. [CrossRef] [PubMed]
4. Kim, H.S.; Lee, K.S.; Ohno, Y.; van Beek, E.J.R.; Biederer, J. PET/CT versus MRI for diagnosis, staging, and follow-up of lung cancer. J. Magn. Reson. Imaging 2014, 42, 247–260. [CrossRef] [PubMed]
5. Ding, C.; Tao, D. Trunk-Branch Ensemble Convolutional Neural Networks for Video-based Face Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2017. [CrossRef] [PubMed]
6. Rasti, P.; Uiboupin, T.; Escalera, S.; Anbarjafari, G. Convolutional Neural Network Super Resolution for Face Recognition in Surveillance Monitoring. In Proceedings of the International Conference on Articulated Motion and Deformable Objects, Palma de Mallorca, Spain, 13–15 July 2016.
7. Levi, G.; Hassnner, T. Age and gender classification using convolutional neural networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015.
8. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. In Proceedings of the IEEE: Piscataway, NJ, USA, 1998; Volume 86, pp. 2278–2324. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems; Communications of the ACM: New York, NY, USA, 2012; Volume 1, pp. 1097–1105. [CrossRef]
10. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
11. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 2014, arXiv:1409.1556.
12. 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.
13. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. Med. Image Anal. 2017, 42, 60–88. [CrossRef] [PubMed]
14. Greenspan, H.; van Ginneken, B.; Summers, R.M. Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. IEEE Trans. Med. Imaging 2016, 35, 1153–1159. [CrossRef]
15. Tajbakhsh, N.; Shin, J.Y.; Gurudu, S.R.; Hurst, R.T.; Kendall, C.B.; Gotway, M.B.; Liang, J. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? IEEE Trans. Med. Imaging 2016, 35, 1299–1312. [CrossRef] [PubMed]
16. Shi, J.; Zhou, S.; Liu, X.; Zhang, Q.; Lu, M.; Wang, T. Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset. Neurocomputing 2016, 194, 87–94. [CrossRef]
17. Brosch, T.; Tam, R. Manifold Learning of Brain MRIs by Deep Learning. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention-MICCAI, Nagoya, Japan, 22–26 September 2013.
18. Shorten, C.; Khosghoostaar, T.M. A survey on Image Data Augmentation for Deep Learning. J. Big Data 2019, 6, 60. [CrossRef]
19. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic Data for Text Localisation in Natural Images. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
20. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. Adv. Neural Inf. Process. Syst. 2014, 27, 2672–2680, arXiv:1406.2661.
21. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv 2016, arXiv:1511.06434v2.
22. Perez, F.; Vasconcelos, C.; Avila, S.; Valle, E. Data Augmentation for Skin Lesion Analysis. In OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis; Stoyanov, D., Ed.; CARE 2018, CLIP 2018, OR 2.0 2018, ISIC 2018; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11041. [CrossRef]
23. Madani, A.; Moradi, M.; Kararygirs, A.; Syeda-Mahmood, T. Chest x-ray generation and data augmentation for cardiovascular abnormality classification. In Proceedings of the SPIE 10574, Medical Imaging 2018: Image Processing, 105741M, Houston, TX, USA, 2 March 2018. [CrossRef]
24. Real, E.; Moore, S.; Selle, A.; Saxena, S.; Suematsu, Y.L.; Tan, J.; Le, Q.; Kurakin, A. Large-Scale Evolution of Image Classifiers. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
25. Ma, B.; Li, X.; Xia, Y.; Zhang, Y. Autonomous deep learning: A genetic DCNN designer for image classification. *Neurocomputing* 2020, 379, 152–161. [CrossRef]

26. Yang, W.H.; Tarng, Y.S. Design optimization of cutting parameters for turning operations based on the Taguchi method. *J. Mater. Process. Technol.* 1998, 84, 122–129. [CrossRef]

27. Ballantyne, K.N.; van Oorschot, R.A.; Mitchell, R.J. Reduce optimisation time and effort: Taguchi experimental design methods. *Forensic Sci. Int. Genet. Suppl. Ser.* 2008, 1, 7–8. [CrossRef]

28. Singh, K.; Sultan, I. Parameters Optimization for Sustainable Machining by using Taguchi Method. *Mater. Today Proc.* 2019, 18, 4217–4226. [CrossRef]

29. Armato, S.G., III; Hadjiiski, L.; Tourassi, G.D.; Drukker, K.; Giger, M.L.; Li, F.; Redmond, G.; Farahani, K.; Kirby, J.S.; Clarke, L.P. SPIE-AAPM-NCI Lung Nodule Classification Challenge Dataset. *Cancer Imaging Arch.* 2015. [CrossRef]