Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010753
Article

Tech Science Press

Effect of Data Augmentation of Renal Lesion Image by Nine-layer Convolutional Neural Network in Kidney CT

Liying Wang[1], Zhiqiang Xu[2] and Shuihua Wang[3,4,5,*]

[1]School of Education Science, Nanjing Normal University, Nanjing, 210097, China
[2]Unit of Urology, Tongliao Hospital of Inner Mongolia, Tongliao, 028000, China
[3]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, 541004, China
[4]School of Mathematics and Actuarial Science, University of Leicester, Leicester, LE1 7RH, UK
[5]School of Architecture Building and Civil Engineering, Loughborough University, Loughborough, LE11 3TU, UK
*Corresponding Author: Shuihua Wang. Email: shuihuawang@ieee.org

Received: 26 March 2020; Accepted: 27 April 2020

Abstract: Artificial Intelligence (AI) becomes one hotspot in the field of the medical images analysis and provides rather promising solution. Although some research has been explored in smart diagnosis for the common diseases of urinary system, some problems remain unsolved completely A nine-layer Convolutional Neural Network (CNN) is proposed in this paper to classify the renal Computed Tomography (CT) images. Four group of comparative experiments prove the structure of this CNN is optimal and can achieve good performance with average accuracy about 92.07 ± 1.67%. Although our renal CT data is not very large, we do augment the training data by affine, translating, rotating and scaling geometric transformation and gamma, noise transformation in color space. Experimental results validate the Data Augmentation (DA) on training data can improve the performance of our proposed CNN compared to without DA with the average accuracy about 0.85%. This proposed algorithm gives a promising solution to help clinical doctors automatically recognize the abnormal images faster than manual judgment and more accurately than previous methods.

Keywords: Artificial intelligence; convolutional neural network; data augmentation; renal lesion; computed tomography image

1 Introduction

In general, the common diseases of urinary system of human includes calculi, infection, tumor, congenital dysplasia and trauma. CT as one main tool is applied to detect and diagnose most of them. With the development of the digitalized and intelligent medical diagnosis, AI becomes one hotspot in the field of the medical images analysis and provides rather promising solution. In an essence, AI could not replace human wisdom but assist in overcoming the disadvantage of human being energy limited and fallible. Although some research has been explored in smart diagnosis for the common diseases of urinary system, some problems remain unsolved completely.

Han et al. [1] has transferred the GoogleNet model to classify the renal cancer into three major subtypes in three-phrase enhanced CT images. They constructed three classifiers to implement the tri-class task. The accuracy of this algorithm on a dataset with 169 images could reached 85%.

Kuo et al. [2] investigated Resnet to predict eGFR and CKD status in the ultrasound images, the average accuracy on a dataset with 1446 images was nearly 85.6%.

Both of the deep learning application are facing same challenges, they didn't make any comparison with other algorithms, meanwhile, the performance of the test accuracy still need be improved in further.

Kumar et al. [3] proposed a two-layer perceptron with back propagation algorithm to diagnose the renal stone disease. Their dataset contained 1000 samples from real clinical data. They extract eight kinds of features to express the renal stone symptoms. The accuracy of this algorithm was 92%. It showed this neural network took 5% and 8% advantages over Learning Vector Quantization and Radial Basis Function respectively.

Mangayarkarasi et al. [4] adopted a PNN model to classify the renal ultrasound images into normal and abnormal categories. Their dataset only contained 24 normal and 53 abnormal images. Which are preprocessed by histogram equalization, mean filter and Gauss filter, segmentation of Region of Interest (ROI) operations. Then the PNN is trained by inputting the image attributes of mean, entropy and variation of one image. Though the overall average accuracy was 93.5%, the generalization of this method is hard to guarantee for possible existing overfitting on this small dataset.

These neural networks took specified features extracted by experience as the input data and obtained about 7% higher accuracy than typical deep learning models. This indicates that supervised machine learning models depend on both human knowledge and training data. Because the number of the training samples are often deficient, the performance of the classifier may be less generalization.

Since the typical CNN model has been applied successfully in massive discriminative tasks of various fields, such as medical diagnosis, it becomes one promising tool for researchers. Wang et al. [5] designed a seven-layer CNN to classify the renal lesion on the dataset with 614 CT images and got the state of art result of 90.36 ± 1.02% accuracy.

In the case of the training data acquired rather difficult, some methods such as transfer learning, data augmentation and so on could be applied to solve the problem of overfitting [6]. The effect of data augmentation techniques on image data depend on whether the data label is preserved after data warping and oversampling. For example, Wang [7] used rotation, gamma transformation and noise injection to augment CNN training dataset so as to achieve better performance in alcohol use disorder detection. Afterwards, this paper attempts to investigate the effect of data augmentation in our CNN model to distinguish the renal lesion.

The main contributions of this paper have three points. The first one is to improve the recognition accuracy for the renal lesion on CT dataset with CNN. The second one is to normalize the distribution of our dataset by color space transformation. The third one is to investigate the effect of the data augmentation on the class imbalance dataset.

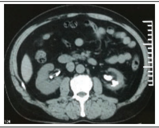
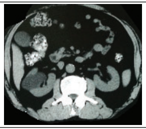
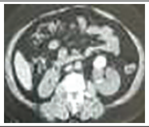
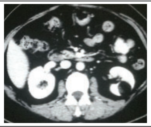
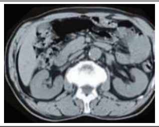
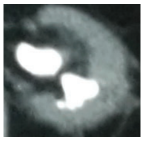
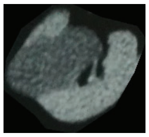
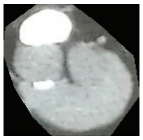
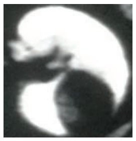
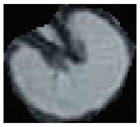
The remaining of this paper is organized as below. Section 2 describes the dataset of the related kidney CT images from the clinical patients. One CNN with data augmentation is constructed and explained in detail according to the classification target in Section 3. As following, the implementation parameters and results are tuned and illustrated, especially the effect of the data augmentation is discussed. Furthermore, more comparisons between the structures of CNN and other deep learning models are checked in Section 4. Final conclusion is drawn at the end of the paper (Section 5).

2 Dataset

This study got formal written consent approved by the Ethics Committee of Tongliao hospital of Inner Mongolia and all the subjects in the dataset kept credential formal written consent with Tongliao hospital. Our dataset consists of 614 kidney CT images, which is collected by clinical doctors through general or enhanced scans to diagnose the renal lesions with the device of Siemens SOMATOM Force CT in Tongliao hospital.

According to the clinical diagnosis by experienced doctors, these kidney CT images are identified as abnormal and normal classes. Moreover, there are four subtypes representing typical renal diseases in the abnormal class which are calculi, cysts or Hydronephrosis, calculi with cysts or Hydronephrosis and tumor. Then the samples in our dataset are illustrated in Tab. 1. To construct our CNN classifier, the human preprocessing only includes cropping out the Region of Interest (ROI) in the CT images. No any other preprocessing need to be done. In addition it should be worth mentioned the enhanced CT images are selected by doctors using the excretory urography which is taken from the excretory phase after injecting contrast agents about 15 minutes. For example, the sample of 1–4 tumor subtype was scanned by the enhanced CT, its tumor lesion was located in the dark part of the corresponding ROI. Comparatively, the rest samples in this table were scanned by the general CT, the lesion contrast of which varied significantly in intensity, shape and size.

Table 1: Kidney CT and ROI of our dataset

Category	Abnormal				Normal
Type	1-1	1-2	1-3	1-4	Healthy
CT					
ROI					

(Renal Lesion Type: 1-1 = calculi, 1-2 = cysts or hydronephrosis, 1-3 = calculi with cysts or hydronephrosis, 1-4 = tumor).

Totally our dataset involves 500 general CT images and 114 enhanced CT. These CT images cover two categories of normal and abnormal with total five renal lesion types. We define the abnormal category as the positive class and the normal category as the negative class.

As following, the size of our dataset presented in Tab. 2 is obviously not large and exists the case of class imbalance. The main reasons of this come from the difficulties for clinical doctors in their daily works to track diseases, mark images and integrate text records all together for us. Thus, two problems are prone to appear, overfitting caused by training on a small dataset and biasing to the majority class for prediction by training on a class imbalance dataset. To solve the problem of class imbalance, a common method is to increase the penalty cost of wrong prediction for minority class in the target function of the classifier model. In next section, data augmentation as a solution to alleviate these two problems is clarified in detail.

To evaluate the generalization of one classifier, it is best to use new data instead of training data to test. The holdout idea separates a part of dataset as test data and uses the remainder as training and validation data. In order to keep different classes in test data with same probability, same number of test data are hold out based on the class with less samples as below in Tab. 3.

Table 2: Distribution description of our dataset

Image Category	Renal Lesion Type	Sample Size
Normal	Healthy	191
Abnormal		423
	Calculi	145
	Tumor	80
	Cysts or Hydronephrosis	107
	Calculi with cysts or hydronephrosis	91
Total		614

Table 3: Test with holdout data

	Healthy	Abnormal
Test	41	41
Training	150	382
Total	191	423

3 Methodology

As we have seen, CNN as a classic deep learning model is one end to end network which is heavily explored for researchers. Our method comprises two parts as shown in Fig. 1. One part of this method is the data augmentation techniques applied to enlarge our kidney CT dataset. The other is one nine-layer CNN constructed to classify the renal lesion through training. Next, the related theories should be understood and interpreted at first for further reading.

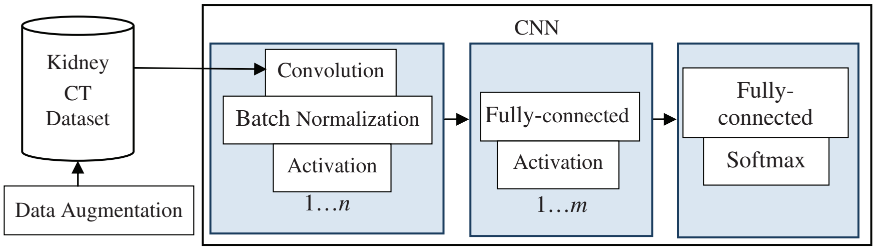


Figure 1: The structure of our proposed method

3.1 *CNN Structure*

According to the function of a layer, a CNN consists of a series of basic layers of complex convolution, activation [8,9], and fully-connected, as well additional layers of pooling, batch normalization, softmax. These layers are organized together in sequence and some of them repeat several times. Each layer as a module is composed of certain units [10–12]. Each unit transforms the input x to the output $f(x)$. Thus, the values from those units of the previous adjacent layer connecting to the unit of this layer compose of its input vector x_i. The corresponding output value of the unit is $f(x_i)$. Therefore, the operation f of the unit and the connection relationship determine the functions of this layer. Thus, the outputs of all units of the layer compose of a vector y as below.

$$\mathbf{y} = [\boldsymbol{y_i} = \boldsymbol{f}(\boldsymbol{x_i})] \tag{1}$$

As for the dimensions of these vectors of input and output, they rely on the number and the connection of the units locating in two adjacent modules. So, the number of the units are those external parameters need be chosen for users. These modules are assembled together carefully to implement a specific image recognition task. Next we explain the functions and the connections in those layers.

3.2 Convolution, Batch Normalization and ReLU Layer

The convolution operation can be expressed as $f(x) = wx + b$, where w is the vector of the convolution kernel weights [13–15], b is the bias for the output. When the convolution operates on an image $I = [Height, \ Width, Channel]$, the size of the kernel filter need be designed initially as $K = [height, width]$. Then the output with this filter in a specific position of the image I is

$$f(I, p, q) = \sum\nolimits_{i=1}^{height} \sum\nolimits_{j=1}^{width} w(i,j) I(p+i, q+j) + b$$
$$0 < p < Height, \ \ 0 < q < Width$$

The connections among the units for a convolution kernel are sparse [16–18], so as to extract these important features in local region. At the same time, w and b are just internal parameters learned from the training data. Then, If the kernel filter is operated with the sliding the window one by one, a new image is output as a feature map. It is obvious that the output size O_1 is

$$O_1 = [Height - height + 1, \ \ Width - width + 1] \tag{3}$$

which is smaller than the input size. Therefore, to these pixels in the border of the input image, the kernel filter needs an additional padding around the four borders so that the output size is same to the input size. When the padding size is P = [up, down, left, right], the output size O_2 is

$$O_2 = [Height - height + up + down + 1, \ \ Width - width + left + right + 1] \tag{4}$$

When the kernel filter slides more than one pixel, another a pair of parameters are set as $stride = [stride\_height, stride\_width]$. Thus, the output size O_3 of the feature map is

$$O_3 = \left[\frac{(Height - height + up + down)}{stride\_height} + 1, \frac{(Width - width + left + right)}{stride\_width} + 1 \right] \tag{5}$$

When set multiple kernel filters as $K = [height, width, kernel\_number]$ which are the external parameters about the size and the number of the kernel filters, the number of the output of the convolution layer is exactly the number of the kernel filters.

From the above explanation, we find out the kernel filter operates the input image iteratively for all pixels. In order to overcome the tediousness, parallel computation in batches applies $B = [Height, Width, Channel, Batch]$, the tensor way to implement the acceleration.

However, the distribution of the batches in the training dataset may vary greatly, which affects the stability of the internal parameters learned. To solve it, Ioffe, Szegedy [19] proposed Batch Normalization (BN) operation before the activation function to reduce the shift of internal covariate [20].

$$BN_{\gamma,\beta}(x_i) = \gamma_i \left[(x_i - \mu_B) / \sqrt{\sigma_B^2 + \epsilon} \right] + \beta_i \tag{6}$$

$$\mu_B = \sum\nolimits_{i=1}^{m} x_i / m \tag{7}$$

$$\sigma_B^2 = \sum_{i=1}^{m} (x_i - \mu_B)^2 / m \tag{8}$$

As far as the activation layer, it is to simulate the response output only when surpass a certain threshold. Common applied functions include $ReLU(x) = \max(0, x), sigmoid(x) = 1/(1 + e^{-z})$ and so on.

3.3 Fully-Connected and Softmax Layer

For a fully-connected layer, the function is same to that of the convolution layer. Differently, each unit of the fully-connected layer collects the information from all units of the previous layer as own input [21–24]. Thus, the connections in the fully-connected layer are rather dense. The number of corresponding weights for the connections is $N = [n \times m]$, much larger compared to the convolution layer. Meanwhile the value of these weights and biases do not be shared each other because the output of each unit represents a value in a definite category [25–28].

If we need get the relative value among different categories, the softmax function realizes this transformation based on the Bayes probability model.

$$\text{softmax}(x) = [e^{x_i} / \sum_i e^{x_i}] \text{ Such that } \sum_i e^{x_i} = 1 \text{ and } e^{x_i} > 0 \tag{9}$$

Typically, x_i with larger relative scores yields exponentially larger probabilities.

3.4 Data Augmentation

Deep Learning relies on big data to avoid overfitting. In the case of the limited data, artificially inflating datasets namely data augmentation achieves the benefit of big data in the limited data domain. Many data augmentation techniques have been proposed for constructing better datasets which can generally be classified as either a data warping or oversampling technique [29].

For data warping techniques, transformations in geometric and color space are two common forms of it. On one hand, geometric transformations encompass translation, rotation, scale, flipping, cropping. On the other hand, color transformations contain color filter, noise injection [30], histogram change, kernel filters, mixing images, random erasing and so on. All of them target to cover the more general data distribution to shorten the difference between training data and test data. However, the disadvantages of these methods include additional memory and time costs computationally. Meanwhile, the error rate drop from some methods such as mixing images is very difficult to explain from a human view.

Data augmentation prevents overfitting by modifying limited datasets to possess the characteristics of big data. It performs best under the assumption that the training and test dataset are both extracted from the same distribution. Otherwise, these methods will very unlikely be useful.

Data augmentation also alleviates class imbalance harm because they prefer the models to majority class predictions and render accuracy as a deceitful performance metric. Data augmentation falls under a data-level solution to it. Many different strategies for implementation are used. A naive and easy solution would be a simple random oversampling with small geometric and color space operations with different class ratios for majority and minority class [31].

However, oversampling could also cause overfitting more prevalent post-sampling on the minority class [32]. So more intelligent strategy on oversampling methods to increase the minority class size while preserving the extrinsic distribution, such as adversarial training, neural style transfer, GANs, and meta-learning schemes is a promising area for future work.

In regards to our samples dataset, because the sample size is small and the class size is imbalanced, data augmentation will be applied to overcome the overfitting and data bias problems. Four geometric transformations and two color transformations are used together to our dataset. The detailed values of

Table 4: Transformation description of our data augmentation

Transformation Type	Name	Value range	Step
Geometric	Affine transformation	[−0.2, 0.2]	Random
	Rotation	[−30, 30]	2
	Scale	[0.7, 1.3]	0.02
	Shift	[−50, 50]	Random
Color	Gamma enhance	[0.4, 1.6]	0.04
	Noise	Sigma = 0.01	Random

these transformations are described as Tab. 4 below, where affine transformation applies two dimensional shear operations, noise type chooses Gaussian white noise with zero mean and variance of 0.01. As for gamma enhance, because gamma represents the degree of adjusting brightness, less than 0.4 will make the new image too bright, while greater than 1.6 will make it too dark.

For each original training sample image, 30 new images are generated by one transformation with the same size to the input of our proposed CNN.

As shown in the following Fig. 2, one original CT image with the renal lesion type of calculi with cysts is taken as one example. As a result, 180 new images are generated by these six transformations with corresponding value ranges and steps. Only six new images of each transformation are exhibited in Figs. 2b–2g, whose indices are 1, 6, 11, 16, 21, 27 in the corresponding 30 new images.

3.5 Implementation

The CNN program is developed in Matlab 2019a. Its training and test stages are all run on a laptop with the operating system of Windows-10, NVidia GeForce GTX 1050 with 5 multiprocessors, and CPU clock rate of 2.2 GHz. To evaluate our CNN's performance, six indicators are used to get the average and overall values from multiple viewpoints. They are sensitivity (recall positive category), specificity (recall negative category), and precision of the positive category, accuracy of all categories, F1 and MCC. MCC gives a correlation coefficient between observation and prediction, whose value ranges between −1 to 1 and means disagreement to a perfect prediction.

4 Experiments, Results and Discussion

4.1 Training Configuration

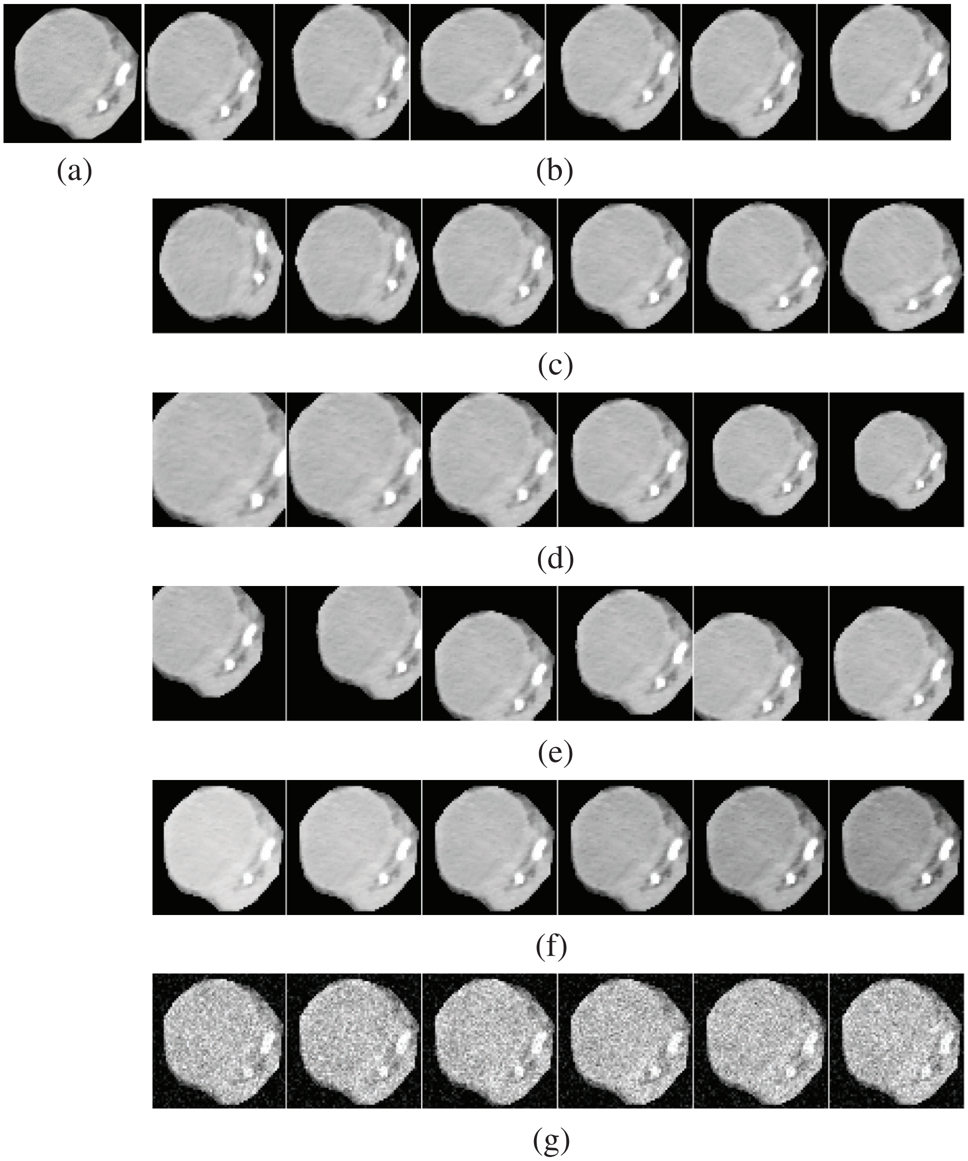
The CNN training algorithm minimizes the loss function [33] with least mean squared error and L_2 regularization item shown as Eq. (10), where there are m training samples and n optimized parameters, r is the L_2 Regularization coefficient set as 0.005.

$$w^* = \underset{w}{\operatorname{argmin}}[L = \frac{1}{m}\sum\nolimits_{i=1}^{m} (f(w,x_i) - y_i)^2 + \mathrm{r}\sum\nolimits_{j=1}^{n} w_j^2] \tag{10}$$

The weights are updated by the optimizer of stochastic gradient descent with momentum method [34] which averages previous gradients together to obtain smoother search path. It is given as Eqs. (11) and (12).

$$w_j = w_j + \mathrm{v}_k \tag{11}$$

$$\mathrm{v}_k = \mathrm{m}\mathrm{v}_{k-1} - \varepsilon\nabla L,\ \mathrm{v}_0 = 0 \tag{12}$$



(a) (b) (c) (d) (e) (f) (g)

Figure 2: Examples of six transformations of our data augmentation. (a) Original (b) Affine transform (c) Rotation (d) Scale (e) Shift (f) Gamma enhance and (g) Noise

where m is the momentum coefficient set as 0.9, ∇L is the gradient of the objective function at one iteration stage, ε is the learning rate which defines how much degree to update the internal parameters in each iteration. The parameters we assign in the software are given as following. The mini-batch size is 128, the maximum epoch is 30. The initial learning rate is 0.001 with decreasing by a factor of 0.1 in step of every 10 epochs.

4.2 Network Configuration

We construct a nine-layer deep CNN to classify our renal CT dataset. The structure of this CNN is shown in Fig. 3. The parameter values of each layer of CNN are described in Tab. 5. The input data are the preprocessed kidney CT images with size 72 × 72 and 3 channels. One whole convolutional layer is composed of the convolution operation directly followed by a BN and a ReLU stage. The output size of one convolution layer is calculated as Eq. (5).

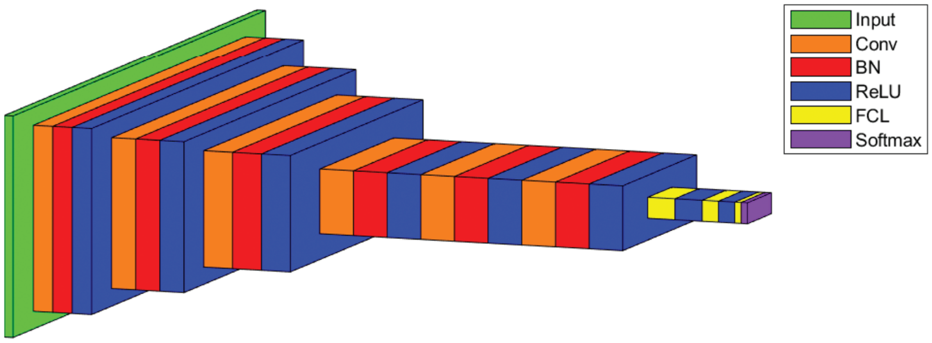


Figure 3: Feature maps of each layer of our proposed network

Table 5: The Structure and parameters of our 9-layer CNN

Layer	Purpose	Kernel size, number	Stride	Padding	Weight	Bias	Activation
	Input						72 × 72 × 3
1	Conv_1	3 × 3, 16	2 × 2	[0 1 0 1]	3 × 3 × 3 × 16	1 × 1 × 16	36 × 36 × 16
2	Conv_2	3 × 3, 32	2 × 2	[0 1 0 1]	3 × 3 × 16 × 32	1 × 1 × 32	18 × 18 × 32
3	Conv_3	3 × 3, 64	2 × 2	[0 1 0 1]	3 × 3 × 32 × 64	1 × 1 × 64	9 × 9 × 64
4	Conv_4	3 × 3, 128	3 × 3	[0 0 0 0]	3 × 3 × 64 × 128	1 × 1 × 128	3 × 3 × 128
5	Conv_5	3 × 3, 128	1 × 1	[1 1 1 1]	3 × 3 × 128 × 128	1 × 1 × 128	3 × 3 × 128
6	Conv_6	3 × 3, 128	1 × 1	[1 1 1 1]	3 × 3 × 128 × 128	1 × 1 × 128	3 × 3 × 128
7	FCL(50)				50 × 1152	50 × 1	1 × 1 × 50
8	FCL(10)				50 × 10	10 × 1	1 × 1 × 10
9	FCL(2)				2 × 10	2 × 1	1 × 1 × 2

Given the input size of Conv_1 layer is [72, 72, 3], the kernel size is [3, 3, 16], the stride size is [2, 2] and the padding is [0, 1, 0, 1], the output width is (72 – 3 + 0 + 1)/2 + 1 = 36. Therefore, the output size of Conv_1 is [36, 36, 16]. Our CNN has 7 such convolution layers to extract multi feature maps. At the end of the CNN pipelines, three fully connected layers (FCL) are added. Two FCLs with ReLU activation are used to output values. FCL(50) means there are 50 neurons in this FCL. Another FCL with softmax activation FCL(2) outputs the probability of the image binary classification.

4.3 Performance of Proposed CNN

We train the proposed CNN ten times and achieve the prediction results on the test dataset. Tab. 6 shows the performance of ten runtimes evaluated by these 6 indicators which are sensitivity, specificity, precision, accuracy, F1 and MCC. Each row gives the performance of one runtime. Finally the mean and the standard deviation of ten runtimes are exhibited at the last row. It indicates our CNN classifier performs rather well and steadily because the average values of the front five indicators are all above 91.98% and the standard deviation of them are less than 2.40%.

4.4 Result of Data Augmentation

Here we investigate the effect of data augmentation of renal lesion image by using our nine-layer CNN in Kidney CT dataset. Besides the above experiments, we run CNN training ten times on original dataset

Table 6: Statistical analysis of 10 runs of our method

Runtime	Sen (%)	Spe (%)	Pre (%)	Acc (%)	F1 (%)	MCC (%)
1	95.12	95.24	95.45	95.18	95.23	90.47
2	92.62	90.24	90.45	91.43	91.51	82.91
3	90.24	90.24	90.24	90.24	90.24	80.48
4	95.12	92.74	92.86	93.90	93.96	87.86
5	87.74	92.74	92.46	90.24	89.97	80.68
6	92.62	95.12	94.99	93.87	93.77	87.79
7	90.24	90.24	90.24	90.24	90.24	80.48
8	95.12	87.86	88.72	91.49	91.80	83.21
9	92.74	92.74	92.74	92.68	92.68	85.48
10	90.24	92.62	92.50	91.43	91.34	82.91
Mean ± SD	92.18 ± 2.40	91.98 ± 2.21	92.06 ± 2.05	92.07 ± 1.67	92.07 ± 1.70	84.22 ± 3.34

without data augmentation. The corresponding performance results are listed in Tab. 7. The averages of the front five indicators are 91.46%, 90.99%, 91.13%, 91.22% and 91.21% respectively. From the comparison of Tab. 8, the corresponding averages of the ten-runtime experiments with DA are 0.72%, 0.99%, 0.93%, 0.85%, 0.86% and 1.63% higher than those without DA. It indicates DA can improve the classification performance through enlarging the training data. On the other hand, data augmentation of each image in original dataset takes nearly 2 seconds. Because image transformation is forward and the dataset is small, so the time cost is quite short. At the same time, the augmented images are stored in hard disk to save RAM memory. Therefore, the spatial cost of data augmentation is acceptable relatively to the enormous hardware capacity. Therefore, the effect of data augmentation performs well.

Table 7: Statistical analysis of 10 runs of without DA

Runtime	Sen (%)	Spe (%)	Pre (%)	Acc (%)	F1 (%)	MCC (%)
1	87.86	92.74	92.37	90.24	90.00	80.69
2	85.36	92.62	92.11	88.99	88.59	78.21
3	90.24	92.74	92.50	91.46	91.34	82.98
4	95.12	87.74	88.72	91.46	91.80	83.15
5	90.24	92.74	92.61	91.49	91.39	83.03
6	95.12	90.24	90.69	92.68	92.85	85.46
7	87.74	92.74	92.46	90.24	89.97	80.68
8	95.12	87.86	88.64	91.46	91.75	83.17
9	95.12	90.24	90.69	92.68	92.85	85.46
10	92.74	90.24	90.48	91.49	91.58	83.03
Mean ± SD	91.46 ± 3.50	90.99 ± 1.92	91.13 ± 1.45	91.22 ± 1.07	91.21 ± 1.27	82.59 ± 2.10

Table 8: Comparison of using DA and not using DA

	Sen (%)	Spe (%)	Pre (%)	Acc (%)	F1 (%)	MCC (%)
Without DA	91.46 ± 3.50	90.99 ± 1.92	91.13 ± 1.45	91.22 ± 1.07	91.21 ± 1.27	82.59 ± 2.10
With DA (Ours)	92.18 ± 2.40	91.98 ± 2.21	92.06 ± 2.05	92.07 ± 1.67	92.07 ± 1.70	84.22 ± 3.34

4.5 Optimal Structure of Convolutional Layers

The convolutional layers do multiple feature extraction in a deep neural network. When fixing three FCLs as Tab. 5, we check how many convolutional layers the CNN should have so as to obtain the best performance. The number of convolutional layers is adjusted from small value to large value. Tab. 9 shows the experimental results of five CNNs with 3 to 7 convolutional layers. All the last convolutional layer has the same parameters setting to the Conv_6 in Tab. 5. It proves that the CNN with 6 convolutional layers is the optimal structure since the performance does not improve any more according to those six indicators.

Table 9: Optimal structure of Convolutional layers

	Sen (%)	Spe (%)	Pre (%)	Acc (%)	F1 (%)	MCC (%)
3Conv + 3FCL	90.73 ± 2.87	90.48 ± 2.30	90.66 ± 1.88	90.61 ± 0.57	90.61 ± 0.73	81.37 ± 1.16
4Conv + 3FCL	90.74 ± 2.64	91.01 ± 1.55	91.07 ± 1.28	90.86 ± 1.11	90.84 ± 1.22	81.85 ± 2.18
5Conv + 3FCL	91.69 ± 1.19	91.73 ± 2.23	91.77 ± 2.09	91.70 ± 1.30	91.69 ± 1.24	83.48 ± 2.61
6Conv + 3FCL	**92.18 ± 2.40**	**91.98 ± 2.21**	**92.06 ± 2.05**	**92.07 ± 1.67**	**92.07 ± 1.70**	**84.22 ± 3.34**
7Conv + 3FCL	91.95 ± 2.17	91.96 ± 1.90	92.05 ± 1.66	91.96 ± 0.81	91.94 ± 0.86	84.02 ± 1.62

4.6 Optimal Number of FCL

Here we fix six convolutional layers, and tuned the number of FCL layers carefully from small to large value. The experiments change the number of FCL from 2 to 5. The results are shown in Tab. 10. The input to the first FCL is the output of the sixth convolutional layer which has 3 × 3 × 128 = 1152 dimensions.

When the number of FCL layers is set as 2, FCL(50) and FCL(2) are applied in sequence. When the number of FCL layers is set as 3, then FCL(50), FCL(10) and FCL(2) are applied in sequence. When the number of FCL layers is set as 4, then FCL(50), FCL(25), FCL(10) and FCL(2) are applied in sequence. When the number of FCL layers is set as 5, then FCL(50), FCL(25), FCL(10), FCL(5) and FCL(2) are applied in sequence. It indicates that the CNN with 3 FCLs performs the best.

Table 10: Optimal number of FCL

	Sen (%)	Spe (%)	Pre (%)	Acc (%)	F1 (%)	MCC (%)
6Conv + 2FCL	91.21 ± 2.22	91.70 ± 1.62	91.74 ± 1.45	91.46 ± 0.94	91.42 ± 1.00	83.01 ± 1.89
6Conv + 3FCL	**92.18 ± 2.40**	**91.98 ± 2.21**	**92.06 ± 2.05**	**92.07 ± 1.67**	**92.07 ± 1.70**	**84.22 ± 3.34**
6Conv + 4FCL	91.46 ± 1.99	91.20 ± 2.92	91.38 ± 2.57	91.34 ± 1.49	91.36 ± 1.42	82.78 ± 2.96
6Conv + 5FCL	91.48 ± 2.2	91.49 ± 2.71	91.60 ± 2.26	91.48 ± 0.77	91.47 ± 0.73	83.07 ± 1.50

4.7 Comparison to State-of-the-Art Algorithms

To validate the advantages of our method over the previous methods, we compare with the related works to classify kidney images. One is PNN model used in paper [4] with the selected features, which include mean, entropy and standard deviation of ultrasound images. The other is our previous 7-layer deep CNN [5]. The results given in Tab. 11 show the average overall accuracy of our nine-layer CNN is 1.71% higher than the 7-layer CNN and over 26% higher than PNN. In fact our proposed 9-layer CNN without data augmentation achieves average accuracy of 91.22 ± 1.07% which performs 0.86% better than the previous 7-layer CNN.

Table 11: Comparison with previous algorithms

Method	Features	Accuracy (%)	Training Time(s)	Test Time(s)
PNN [4]	Selected image features as paper [4] mean+ standard deviation+ entropy	64.51 ± 2.70	1.4	0.2
CNN [5] (5Conv + 2FCL)	Feature maps in 5 convolution layers	90.36 ± 1.02	720	8
DA-CNN (Ours) (6Conv + 3FCL)	Feature maps in 6 convolution layers	92.07 ± 1.67	180 × 811.86	10.37

We also compare the time costs of these methods. The training times are listed in Tab. 11. The nine-layer CNN proposed in this paper takes 811.86 seconds for ten times training on the original dataset, which is called the original training time. When it runs on the augmented training dataset which is 180 times of the original training dataset, it takes about 180 times of the original training time. So on average one training on DA dataset costs about 4 hours. It evidently is more time-consuming than previous methods. Nevertheless, the trade-off is valuable to get a more accurate classification model in the training stage. While in the test stage, the test time of our DA-CNN is comparable because it takes 10.37 seconds on 82 test samples. Therefore the result means only less than 0.13 second is used to identify whether one image has the renal lesion. All in all, it is evident that the new method is faster than manual judgment to get more accurate prediction.

4.8 Discussion

In our deep learning algorithm, the number of convolutional kernels increases with the layers piling up, while the size of them keep same. This is the key point that CNN extracts a large number of local features to replace predefined limited features which are used to differentiate categories of samples. At the fully connected layers, the number of the nodes in one FCL decreases with the layers piling up. This realizes the function of gathering different features layer by layer to summarize the categories.

Meanwhile, the effect of data augmentation is positive to train a more accurate model. After enlarging the training dataset by DA, the learning model converges after certain epochs. So maximum epoch can be set as 10 so as to rationally reduce the training time.

From the above four groups of comparative experiments, we get the optimal structure of the nine-layer CNN. In general, the number of training samples, convolutional layers and fully connected layers could affect the performance of our CNN algorithm to some extent with moderate time cost.

5 Conclusion

In this paper, a nine-layer convolutional neural network is proposed to classify the renal CT images. Four groups of comparison experiments prove the structure of this CNN is optimal and can achieve good performance with average accuracy about 92.07 ± 1.67%. Although our renal CT data is not very large, we do augment the training data by affine, translating, rotating and scaling geometric transformation and gamma, noise transformation in color space. Experimental results validate the Data Augmentation (DA) on training data can improve the performance of our proposed CNN compared to without DA with the average accuracy about 0.85%.

Despite all of them, some works need be done in future. (i) The optimal structure of convolutional and fully connected layers have been verified in our method, but pooling layers are not considered. Further comparison about the pooling effect can be discussed. (ii) We compared with some related works, but more deep neural network algorithms should be covered to find out the best result. (iii) Our dataset includes CT images collected from different sources of the general and enhanced CT devices. Whether different brightness affect the classification performance may be investigated.

Currently the radiography is applying AI to implement the medical image recognition in clinical practice. Our algorithm is validated to be faster than manual judgment and more accurately than previous methods. With the amount of the images increasing in daily check, the limited manual diagnosis is becoming more laborious and time-consuming. Therefore, this kind of automatic identification of abnormal images may be a promising alternative prejudge approach to help clinical radiologists and doctors reduce their workload. Future works also need focus on generalization and interpretability of deep learning method.

Funding Statement: This study was supported by National Educational Science Plan Foundation "in 13th Five-Year" (DIA170375), China. Guangxi Key Laboratory of Trusted Software (kx201901); British Heart Foundation Accelerator Award, UK.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Han, S., Hwang, S. I., Lee, H. J. (2019). The classification of renal cancer in 3-phase CT images using a deep learning method. *Journal of Digital Imaging, 32(4),* 638–643. DOI 10.1007/s10278-019-00230-2.
2. Kuo, C. C., Chang, C. M., Liu, K. T., Lin, W. K., Chiang, H. Y. et al. (2019). Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning. *NPJ Digital Medicine, 2(1),* 1–9. DOI 10.1038/s41746-019-0104-2.
3. Kumar, K., Abhishek, B. (2012). Artificial neural networks for diagnosis of kidney stones disease. *International Journal of Information Technology & Computer Science*. DOI 10.5815/ijitcs.2012.07.03.
4. Mangayarkarasi, T., Jamal, D. N. (2017). PNN-based analysis system to classify renal pathologies in Kidney Ultrasound Images. *2nd International Conference on Computing and Communications Technologies, IEEE, Chennai, India,* 123–126.
5. Wang, L. Y., Xu, Z. Q., Zhang, Y. D. (2020). Renal lesion classification in kidney CT images by seven-layer convolution neural network. *Journal of Medical Imaging and Health Informatics, 10,* 1–9. DOI 10.1166/jmihi.2020.3217.
6. Shorten, C., Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6(1),* 60. DOI 10.1186/s40537-019-0197-0.
7. Wang, S. H., Lv, Y. D., Sui, Y., Liu, S., Wang, S. J. et al. (2018). Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling. *Journal of Medical Systems, 42(1),* 2. DOI 10.1007/s10916-017-0845-x.

8. Jiang, X., Chang, L., Zhang, Y. D. (2020). Classification of Alzheimer's disease via eight-layer convolutional neural network with batch normalization and dropout techniques. *Journal of Medical Imaging and Health Informatics, 10(5),* 1040–1048. DOI 10.1166/jmihi.2020.3001.
9. Wang, S. H., Muhammad, K., Hong, J., Sangaiah, A. K., Zhang, Y. D. (2020). Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. *Neural Computing and Applications, 32(3),* 665–680. DOI 10.1007/s00521-018-3924-0.
10. Liu, S., Zhao, C., An, Y., Li, P., Zhao, J. et al. (2019). Diffusion tensor imaging denoising based on riemannian geometric framework and sparse Bayesian learning. *Journal of Medical Imaging and Health Informatics, 9(9),* 1993–2003. DOI 10.1166/jmihi.2019.2832.
11. Zhang, Y. D., Govindaraj, V. V., Tang, C., Zhu, W., Sun, J. (2019). High performance multiple sclerosis classification by data augmentation and AlexNet transfer learning model. *Journal of Medical Imaging and Health Informatics, 9(9),* 2012–2021. DOI 10.1166/jmihi.2019.2692.
12. Jiang, X., Zhang, Y. D. (2019). Chinese sign language fingerspelling via six-layer convolutional neural network with leaky rectified linear units for therapy and rehabilitation. *Journal of Medical Imaging and Health Informatics, 9(9),* 2031–2090. DOI 10.1166/jmihi.2019.2804.
13. Bera, S., Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing, 41 (7),* 2664–2683. DOI 10.1080/01431161.2019.1694725.
14. Yu, X., Zeng, N., Liu, S., Zhang, Y. D. (2019). Utilization of DenseNet201 for diagnosis of breast abnormality. *Machine Vision and Applications, 30(7–8),* 1135–1144. DOI 10.1007/s00138-019-01042-8.
15. Mao, X., Ding, L., Zhang, Y., Zhan, R., Li, S. (2019). Knowledge-aided 2-D autofocus for spotlight SAR filtered backprojection imagery. *IEEE Transactions on Geoscience and Remote Sensing, 57(11),* 9041–9058. DOI 10.1109/TGRS.2019.2924221.
16. Yamamura, H., Putri, E. U., Kawakami, T., Suzuki, A., Ariesyady, H. D. et al. (2020). Dosage optimization of polyaluminum chloride by the application of convolutional neural network to the floc images captured in jar tests. *Separation and Purification Technology, 237,* 116467. DOI 10.1016/j.seppur.2019.116467.
17. Hong, J., Cheng, H., Zhang, Y. D., Liu, J. (2019). Detecting cerebral microbleeds with transfer learning. *Machine Vision and Applications, 30(7–8),* 1123–1133. DOI 10.1007/s00138-019-01029-5.
18. Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S. et al. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications, 78(3),* 3613–3632. DOI 10.1007/s11042-017-5243-3.
19. Ioffe, S., Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, *ACM, Lille, France,* 448–456.
20. Furusho, Y., Ikeda, K. (2020). Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives. *APSIPA Transactions on Signal and Information Processing, 9,* 2924. DOI 10.1017/ATSIP.2020.7.
21. Basha, S. S., Dubey, S. R., Pulabaigari, V., Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing, 378,* 112–119. DOI 10.1016/j.neucom.2019.10.008.
22. Lu, S., Lu, Z., Zhang, Y. D. (2019). Pathological brain detection based on AlexNet and transfer learning. *Journal of Computational Science, 30,* 41–47. DOI 10.1016/j.jocs.2018.11.008.
23. Yang, J., Jiang, Q., Wang, L., Liu, S., Zhang, Y. D. et al. (2019). An adaptive encoding learning for artificial bee colony algorithms. *Journal of Computational Science, 30,* 11–27. DOI 10.1016/j.jocs.2018.11.001.
24. Zhang, Y. D., Sun, J. (2018). Preliminary study on angiosperm genus classification by weight decay and combination of most abundant color index with fractional Fourier entropy. *Multimedia Tools and Applications, 77(17),* 22671–22688. DOI 10.1007/s11042-017-5146-3.

25. Janke, J., Castelli, M., Popovič, A. (2019). Analysis of the proficiency of fully connected neural networks in the process of classifying digital images: benchmark of different classification algorithms on high-level image features from convolutional layers. *Expert Systems with Applications, 135,* 12–38. DOI 10.1016/j.eswa.2019.05.058.

26. Zhang, Y. D., Zhao, G., Sun, J., Wu, X., Wang, Z. H. et al. (2018). Smart pathological brain detection by synthetic minority oversampling technique, extreme learning machine, and Jaya algorithm. *Multimedia Tools and Applications, 77(17),* 22629–22648. DOI 10.1007/s11042-017-5023-0.

27. Zhang, Y. D., Pan, C., Sun, J., Tang, C. (2018). Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *Journal of Computational Science, 28,* 1–10. DOI 10.1016/j.jocs.2018.07.003.

28. Zhang, Y. D., Pan, C., Chen, X., Wang, F. (2018). Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. *Journal of Computational Science, 27,* 57–68. DOI 10.1016/j.jocs.2018.05.005.

29. Sezer, A., Sezer, H. B. (2020). Deep convolutional neural network-based automatic classification of neonatal hip ultrasound images: a novel data augmentation approach with speckle noise reduction. *Ultrasound in Medicine & Biology, 46(3),* 735–749. DOI 10.1016/j.ultrasmedbio.2019.09.018.

30. Tada, Y., Hagiwara, Y., Tanaka, H., Taniguchi, T. (2020). Robust understanding of robot-directed speech commands using sequence to sequence with noise injection. *Frontiers in Robotics and AI, 6,* 144. DOI 10.3389/frobt.2019.00144.

31. Shahinfar, S., Al-Mamun, H. A., Park, B., Kim, S., Gondro, C. (2020). Prediction of marbling score and carcass traits in Korean Hanwoo beef cattle using machine learning methods and synthetic minority oversampling technique. *Meat Science, 161,* 107997. DOI 10.1016/j.meatsci.2019.107997.

32. Yeom, S., Giacomelli, I., Menaged, A., Fredrikson, M., Jha, S. (2020). Overfitting, robustness, and malicious algorithms: a study of potential causes of privacy risk in machine learning. *Journal of Computer Security, 28 (1),* 35–70. DOI 10.3233/JCS-191362.

33. Lyaqini, S., Quafafou, M., Nachaoui, M., Chakib, A. (2020). Supervised learning as an inverse problem based on non-smooth loss function. *Knowledge and Information Systems, 16(10),* 1063. DOI 10.1007/s10115-020-01439-2.

34. Chaudhuri, A. (2019). The minimization of empirical risk through stochastic gradient descent with momentum algorithms. *Computer Science Online Conference, Cham: Springer*, 168–181. DOI 10.1007/978-3-030-19810-7_17, ISBN: 978-3-030-19810-7.