Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/47/20241274

Exploring the effect of depth and width of CNN models on binary classification of dogs and cats

Xu Li

Khoury College of Computer Science, Northeastern university, Boston, MA, 02115, United States

li.xu2@northeastern.edu

Abstract. Convolutional neural networks (CNNs), one of the trendiest study areas in recent years, have garnered a lot of interest. Their performance is particularly notable in speech recognition, target identification, and picture recognition. The performance of a convolutional neural network model is particularly crucial because convolutional neural networks deal with problems that demand enormous amounts of data and have strict standards for result correctness. The convolutional neural network model's depth and width are the first two parameters that are examined in this paper, which is built on the Tensorflow framework. Next, the impact of depth on the accuracy of model's prediction is examined. The model's accuracy and loss are then used as a benchmark to assess its quality; at the same time, the performance of the model can be discussed in relation to the impact of width by selecting a different number of neurons with various convolutional kernel sizes; Finally, A combinational convolutional, the experiments' result show that, with a classification accuracy of up to 95.4%, the combinatorial convolutional neural network outperforms a model with single depth or width.

Keywords: Image Classification, Convolutional Neural Networks, Model Architecture.

1. Introduction

The advancement of deep learning is a result of the emergence of an infinite number of neural network model architectures, which is a significant component of the field of artificial intelligence research. The convolutional neural network (CNN) is more and more frequently used for computer vision problems like image identification, identifying objects in photographs, and detecting faces. In computer vision issues like image identification, speech recognition, and object detection, CNN is frequently utilized, and its performance is more noticeable.

In 1980, Neocognitron is proposed, which made a great contribution to the development of artificial intelligence [1]. 1989 Yann LeCun proposed the first real CNN -LeNet. With the concept of CNN, scholars have been continuously engaged in research in this field, and CNN has achieved better recognition in several pattern recognition neighborhoods, especially in handwriting digit recognition. LeNet [2]. Since the notion of CNN was first proposed, researchers have been actively working in this area, and CNN has produced positive recognition results in a number of pattern recognition domains, particularly in handwritten digit identification [3]. However, at that time, CNN can only do image recognition on small-scale datasets, and the recognition effect is not good when applied to large-scale

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/47/20241274

datasets. Researchers didn't regain their enthusiasm about convolutional neural networks again until AlexNet model won the ImageNet 2012 challenge [4].

CNN's network structure is built similarly to a house using a convolution layer, an activation function layer, and a pooling layer. Although the convolution layer and the pooling layer are formed differently in CNN than they are in a typical neural network.

CNN performs quite well when used to address the issue of image data type. The process of CNN to compute the output result through Forward Propagation (Forward Propagation) is called feed forward process, in which the input layer inputs the data, and the first hidden layer computes the weights and bias of the parameters, which are then passed on to the second hidden layer, and so on and so forth, and finally output the computed result [5]. In other words, up to the last layer, the output result, the previous output is used as the input of the next layer. Forward propagation does not involve any changes to the network parameters [6]. The loss function is calculated using error back propagation (back propagation), and the weights of the parameters and the bias error are propagated up to the top layer from the bottom layer until all parameters are finely trained [7,8].

With the improvement of computer hardware and software performance, convolutional neural networks have entered a period of rapid development, and in the current situation, the performance of CNNs at different depths and widths has become the most popular research topic [9]. Since deeper network model architectures, which suggest better linear expressiveness, can learn more complex transformations and can thus fit more complex feature inputs, it is generally believed that their predictive expressiveness is superior to that of simple single-layer network models. Bengio and Lecun argued that most functions if they can be solved just fine with a deep structure. then it is impossible to solve them with a shallower model that is shallower and has the same structure [10]. The wider the width of the model (number of neurons vs. size of convolutional kernel), the richer the features will be learned in different directions as well as at different frequencies at each layer. Since a general width neural network does not have connections between multiple layers, the coupling between layers is low and the computational speed of the whole model is increased to a great extent [11]. By increasing the depth and the width of the CNN model, respectively, the performance is strengthened.

This paper takes the cat and dog image recognition dataset provided by Kaggle as an example, discusses how different depths and widths affect the model performance by analyzing the accuracy of CNN for cat and dog image recognition under different depth and width conditions; analyzes how to select a CNN model with appropriate depth and width for a specific problem; and constructs a new CNN model for cat and dog image recognition through the idea of integrated learning, and achieves a high prediction accuracy.

2. Method

2.1. Dataset

The dataset is the cat and dog image recognition dataset provided by Kaggle, which is a classical binary classification problem that aims to determine whether it is a cat or a dog based on the input image [12]. There are 25,000 color photos altogether in the collection, 12,500 of them are dogs and 12,500 are cats. the sizes of the images range, with resolutions ranging from150×150 to500×500 varying. The dataset is organized into two folders. Each folder keeps cat and dog photographs individually for each 12500 images, making a total of 25000 cat and dog images in the train folder and the test folder, respectively. The training, validation, and test sets in this dataset are all included in the same folder. In this study, the train_test_split function of sklearn was utilized twice: once to split the original dataset into a training set and a temporary set, and once to separate the temporary set into a test set and a validation set. The result was an 8:1:1 ratio between the aforementioned three sets. Additionally, the stratify parameter was also used. The input image size is changed to 150×150×3 (3 is the number of image channels) in order to increase training effectiveness, and the experiment is started as a result. The visualization results are displayed in Figure 1.

DOI: 10.54254/2755-2721/47/20241274



Figure 1. Representative examples of the dataset [12].

2.2. Model

The CNN's network structure is similar to that of a general neural network, with the exception that it has a convolution layer and a pooling layer. The CNN is made up of an activation function ReLU (or Sigmoid), an input layer, and a convolutional layer [13]. The pooling layer, the convolutional layer, and the activation function ReLU (or Sigmoid) layer are collectively referred to as the hidden layer. The structure of CNN is shown in Figure 2.

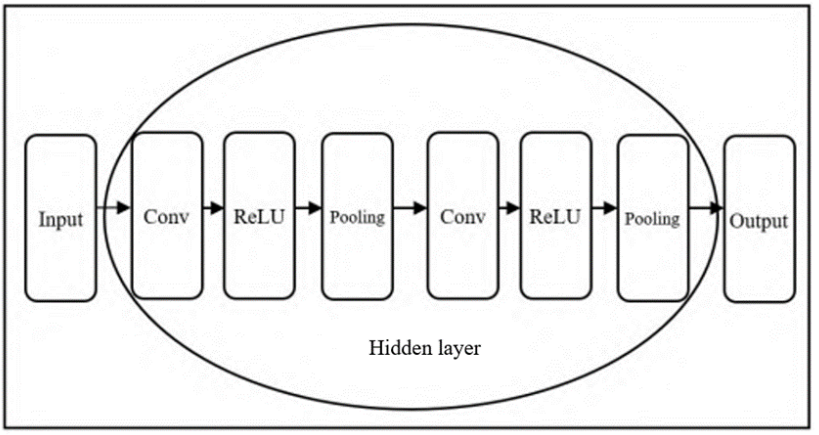

Figure 2. Architecture diagram of CNN (Figure Credits: Original).

2.2.1. Convolutional Layer

In a conventional neural network, features from the input image are extracted by connecting the input layer and hidden layer via a fully connected connection. This fully connected connection results in a slow calculation speed when dealing with large-size images because of the increased computational volume. However, the convolutional layer, as the most crucial component of CNN, each hidden unit can only connect a part of the input unit, which is skillfully solved by the convolutional operation.

Before understanding convolutional operations, let's clarify a few concepts.

Convolution Kernel: Each matching pixel in the output image is produced by weighting and averaging the pixels in a small area of the input image using a function called the convolution kernel. It is common to refer a convolution kernel as a filter. [14].

Stride: how much the convolution kernel moves.

Padding: Before the image is subjected to convolutional operations, a fixed value is filled around the input data, called padding.

The convolutional layer performs the convolutional operation, and the convolutional kernel is a two-dimensional matrix (it can be 1×1, 2×2, or 3×3, or even larger, the exact size can be chosen by you). For

DOI: 10.54254/2755-2721/47/20241274

the input data, the convolution operation slides the window of the convolution kernel at regular intervals and applies 0. As the blue 3×3 part shown in Figure 3, each element on the blue window of 3×3 is first multiplied and then summed with the element at the corresponding position on the convolution kernel of 3x3, and the result obtained is saved to the corresponding position on the output. Then, according to the step size (Figure 3 with 1 step size), the blue window is slid over the input data in the order from left to right and from top to bottom, and all the results of the convolution operation can be obtained.

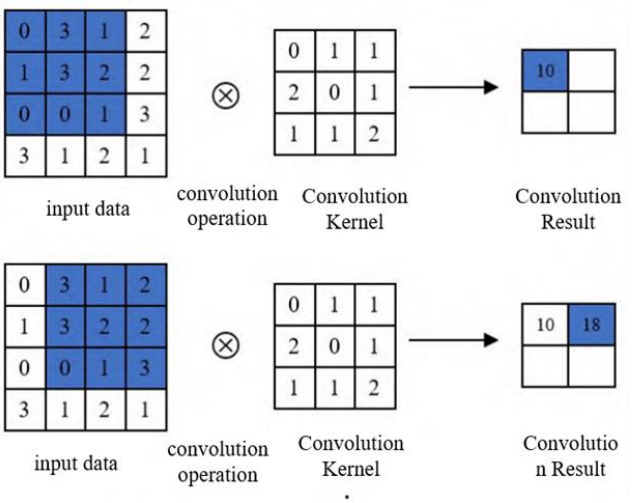

Figure 3. Convolutional operation (Figure Credits: Original).

Parameters are typically chosen before the beginning to train the model. The convolutional neural network's capacity for prediction increases with the number of convolutional kernels. The entire convolutional layer satisfies equation (1)

$$y_i^n = f\left(\sum_{j\in\mathbb{N}_i}\left(y_j^{\mu-1}\times k_{ji}^n\right)+b_i^n\right) \quad (1)$$

where y_i^n is used to indicate the value of the first layer (current layer) on the first layer (current layer): y is used to indicate the output of the convolution performed on the first neuron in the first layer (current layer): y_j^{n-1} represents the results of the convolution performed on the first neuron in the first "n-1" layer of the first$j$. Y denotes the input of the first neuron in the first layer (previous layer); k_{ji}n y represents the first neuron's input in the first layer (previous layer).the convolution kernel of n layers; b_i^n is the outcome from the first neuron in the$n$ layer (the previous layer).i the bias of the first neuron; The collection of feature maps is M_i.$f(*)$ denotes the nonlinear activation function, and the sigmoid and tanh functions are the two most commonly used nonlinear activation functions. ReLU was chosen as the non activation functio. The mathematical expression is shown in Equation (2).

$$h(x)=\begin{cases}x, x>0\\0, x\leqslant 0\end{cases} \quad (2)$$

The convolutional layer's main responsibilities are to extract pertinent features from the input image and to reduce the network parameters in order to maintain the network's sparsity and avoid overfitting.

2.2.2. Pooling Layer

The only modification pooling layer made is the reduction of the feature size. It does not change any of the feature map's essential details. In order to perform the specific operation, the feature map in the convolutional layer is typically divided into a number of small regions. Each small region's values are then chosen to replace all the values in this small block. The most common pooling techniques are Max pooling and Mean pooling. There are other pooling techniques as well. To represent the value of the

DOI: 10.54254/2755-2721/47/20241274

max pooling method, the maximum value of the picture region is selected. Taking Max pooling as an example, the order of operation of pooling is shown in Figure 4.

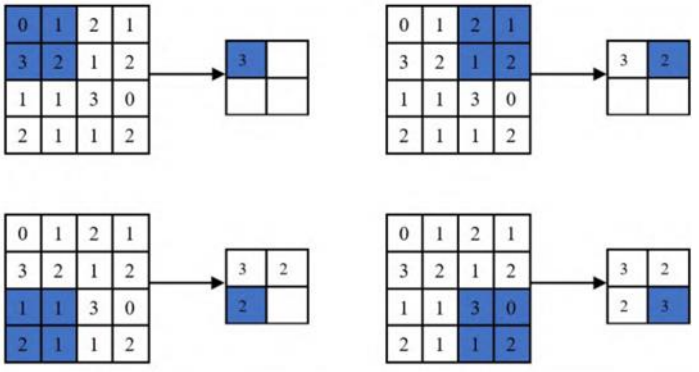

Figure 4. Pooling operation (Figure Credits: Original).

The pooling layer has three main features, first, the pooling layer differs significantly from convolutional layers by having 0 learning parameter of the laye, and only the average or maximum pooling operation is needed; second, each channel carries out an independent pooling operation, so the number of channels before and after the operation is fixed; third, the pooling layer is robust to deviations at small positions, when the value at some positions of the input data changes slightly When there is a small change in the value at some position of the input data, the result remains unchanged after the pooling layer operation.

2.2.3. Fully Connected Layer

Following the pooling of the convolution across multiple layers, the data is sent to a fully connected layer. The classification information from the convolutional and pooling layers is combined with the features from the fully connected layers. A layer cannot be fully connected unless every neuron in it connects with every neuron in the layer above it. This layer also uses the activation function, typically the ReLU activation function, to improve network performance. To get the result, the data is ultimately moved to the softmax for output.

In general, the common CNN structure ends here. The regularization (dropout) operation is carried out in the fully-connected layer to avoid the overfitting phenomenon because, when applied to real-world problems, overfitting will result in the majority of the prediction results, producing unsatisfactory prediction results. The dropout technique will be chosen to discard half of the neurons, in order to reduce the dependency between neurons and to weaken the synergy between the features and to make the learning between neurons more stable,

Evaluation Indicators

This time, the model is used to check the accuracy rate (Precision), check the full rate (Recall), F1 value, accuracy rate (ACC), 4 kinds of indexes to measure the effectiveness of the text classifier, as demonstrated in Table 1.

Table 1. Confusion matrix.

	Predicted to be a positive example	Predicted as a negative example
True as a positive example	TP (true example)	FN (false negative)
True as a negative example	FP (False Positive)	TN (true negative)

(1) Checking accuracy rate P is the proportion of samples predicted positively and correctly by the classifier to all samples predicted positively and is calculated as follows:

$$P = \frac{TP}{TP+FP} \tag{3}$$

DOI: 10.54254/2755-2721/47/20241274

(2) Checking the full rate R is the proportion of samples predicted positively and correctly by the classifier to all true positive samples and is calculated as:

$$R = \frac{TP}{TP+FN} \tag{4}$$

(3) The accuracy rate ACC is a metric for thoroughly assessing the model's results in classification; the higher the accuracy rate, the more effective the classifier is.

$$ACC = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{5}$$

(4) F_1 The value is a combination of$P$ and$R$ of a metric, and the general formula for calculating it.

$$F_\beta = \frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R} \tag{6}$$

At this point, the weighted harmonic mean of P and R serves as the foundation for F_β at this point. R's relative importance to P is indicated by the ratio $\beta > 1$. The formula degenerates to the predetermined standard F_1 value when it is typically taken as $\beta = 1$, as shown in the formula:

$$F_1 = \frac{2\times P\times R}{P+R} \tag{7}$$

3. Results

3.1. Experimental Setup

In this paper, experiments are carried out on cat and dog image dataset based on Tensorflow framework, and two CNNs are constructed, which are deep CNN (deep-net) and wide CNN (wide-net). A CNN with many hidden layers is referred to as a "deep-net," and the network architecture is shown in Figure 5. The number of neurons is 16, 32, 64, and 256, and the number of layers (including the input layer) is 4, 6, 8, and 10. A CNN with one hidden layer and lots of neurons is known as a "wide-net". The number of neurons is 32, 64, 128, 256, 512, 1024, 2048, and 4096, and the network architecture is shown in Figure 6. The selected activation function is ReLU, and the selected optimizer is Adam. Additionally, the fully connected layer now includes a dropout technique to prevent overfitting since it occurs between the training set and the validator. [14]. The dropout value is set to 0.5.

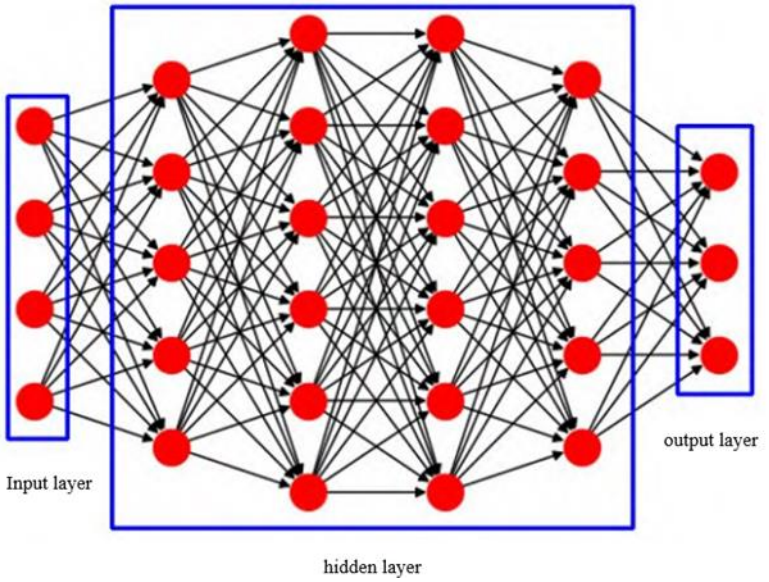

Figure 5. Architecture of Deep-net architecture (Figure Credits: original).

DOI: 10.54254/2755-2721/47/20241274

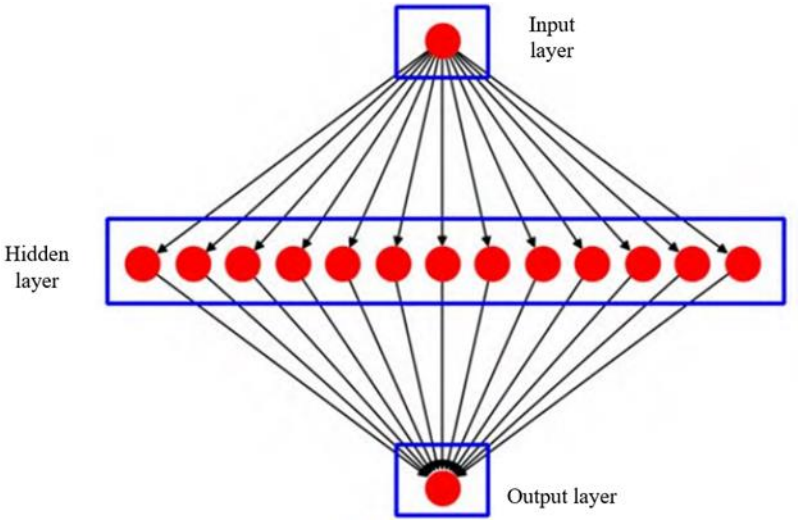

Figure 6. Architecture of Wide-net architecture (Figure Credits: Original).

3.2. Performance Comparison

The number of neurons used in the tests to determine how deep CNN affected the performance of the cat and dog image recognition models was 16, 32, 64, 128 and 256, respectively. The number of network model layers used in the tests was 4, 6, 8 and 10. The results of the experiment shows a rising and then a falling trend in the model's prediction accuracy. When the convolutional neural network has 6 layers and 256 neurons, the model's has its highest accuracy, and the performance is demonstrated in Table 3-1, where the prediction accuracy reaches 75.4%. Deep-net Accuracy denotes the accuracy of the deep CNN. Table 2 illustrates the association between the number of CN layers and prediction accuracy. Figure 7 showa the relationship between prediction accuracy.

When there are more than six network layers, the model's accuracy declines instead. Therefore, the deeper the CNN, the less accurate its predictions will be. The gradient loss problem will encountered by the CNN as the number of network layers rises, decreasing the CNN's ability to make accurate predictions. When solving different practical problems, the depth of the model should be optimized so that the prediction result of the model is optimal.

Two concepts serve as the foundation of this essay's investigation of how width affects the CNN model's performance. It first constructs a convolutional network with one hidden layer and 32, 64, 128, 256, 512, 1024, 2048, or 4096 neurons as the neuron numbers in the network's hidden layer. Analyze the impact on the model's performance.

Table 2. Accuracy of deep CNN models at different depths.

Number of layers	Deep-net accuracy/%
4	71.7
6	75.4
8	74.4
10	73.6

As Table 3 shows, the highest accuracy of 75.8% is achieved when there are 2048 neurons, and when there are more than 2048 neurons, the model accuracy significantly declines. Wide-net Accuracy shows the accuracy of a wide CNN. Table 3 illustrates the association between the number of neurons and model precision. Therefore, the performance of the model is related to the width of the CNN, not the wider the width the better, when the width increases, the model parameters increase, the longer the

DOI: 10.54254/2755-2721/47/20241274

training time, and easy to overfitting phenomenon. To build a good CNN model, the number of neurons is also an object to focus on.

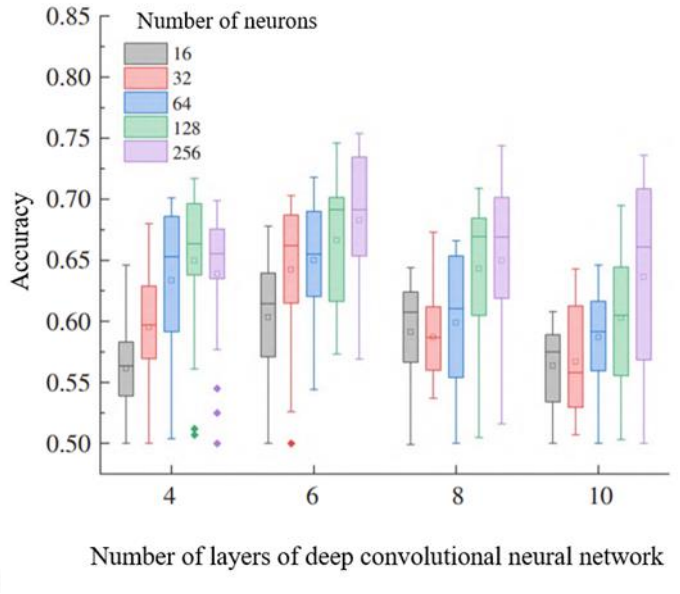

Figure 7. Performance of deep CNN models at different depths (Figure Credits: Original).

Table 3. Accuracy of width CNN model with different number of neurons.

Number of neurons	Wide-net accuracy/%
32	60.0
64	64.4
128	68.3
256	72.2
512	74.3
1024	74.9
2048	75.8
4096	74.4

Second, a convolutional network consisting of one hidden layer is also constructed, fixing the number of neurons to 32 and setting the convolutional kernel sizes to 1×1, 3×3, 5×5, 7×7, and 9×9 to analyze its impact on the model performance.

As Table 4 shows, the convolution kernel size of 5 × 5 yields the highest model prediction accuracy of 69%. Wide-net Accuracy indicates the accuracy of the width CNN. Figure 8 displays the relationship between convolutional kernel size and model precision as a plot. Similarly, choosing the right convolution kernel is a must for constructing an excellent model.

Table 4. Accuracy of width CNN model with different convolutional kernel sizes.

Convolution kernel size	Wide-net accuracy/%
1×1	64.1
3×3	68.7
5×5	69.0
7×7	68.9
9×9	67.4

DOI: 10.54254/2755-2721/47/20241274

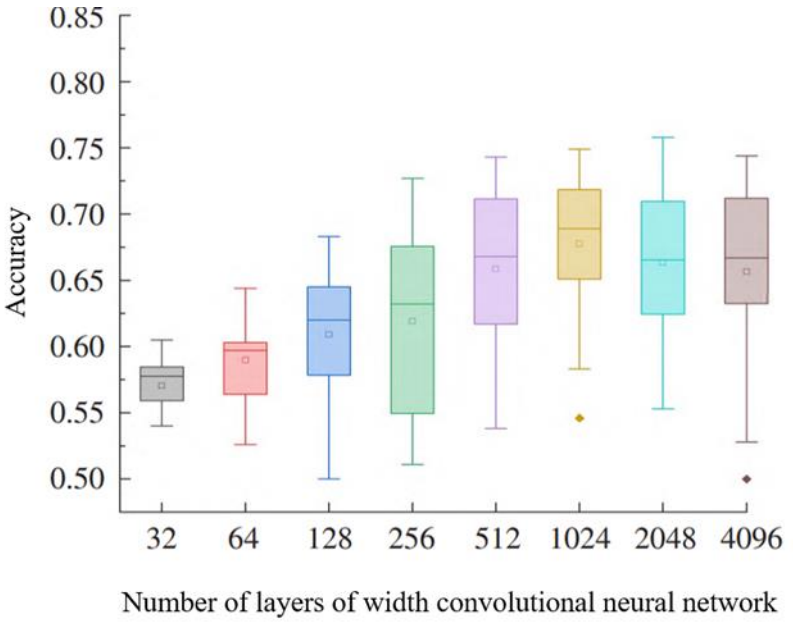

Figure 8. Performance of width CNN model with different number of neurons (Figure Credits: Original).

In summary, when considering the impact of CNN depth on model performance, it is inappropriate to pursue too deep depth, because it is not necessary that the deeper the CNN depth, the better the model performance will necessarily be, the depth is too deep, to take into account the problem of gradient loss, and in the face of different problems, the performance of the CNN will instead be worse. Similarly, when considering the effect of CNN width on model performance, not only the number of neurons should be considered, but also the appropriate convolutional kernel size should be selected.

In this paper, the convolutional layer count is set to 4, Tables 5 shows the number of neurons and convolutional kernel sizes, and six combined convolutional neural networks are proposed. These networks combine depth CNNs and width CNNs, and their accuracies are compared. It was found that the prediction accuracies of the combined convolutional CNNs were all higher than those of the depth or width CNNs alone, and observation of the accuracy and loss plots of DWNet5 and DWNet6, which had the highest prediction accuracies, on both the training and validation sets revealed that the combined CNNs had a better fitting ability and better model performance. The accuracy and loss plots of the trained CNNs are shown in Figures 9 below.

Table 5. Prediction accuracy of different combinations of CNN models.

Network name	Con1	Con2	Con3	Con4	Convolution kernel size	Accuracy(%)
DWNet1	16	32	64	64	3x3	80.6
DWNet2	16	32	64	64	5x5	81.2
DWNet3	32	64	128	128	3x3	81.5
DWNet4	32	64	128	128	5x5	82.0
DWNet5	64	128	256	256	3x3	90.0
DWNet6	64	128	256	256	5x5	95.4

DOI: 10.54254/2755-2721/47/20241274

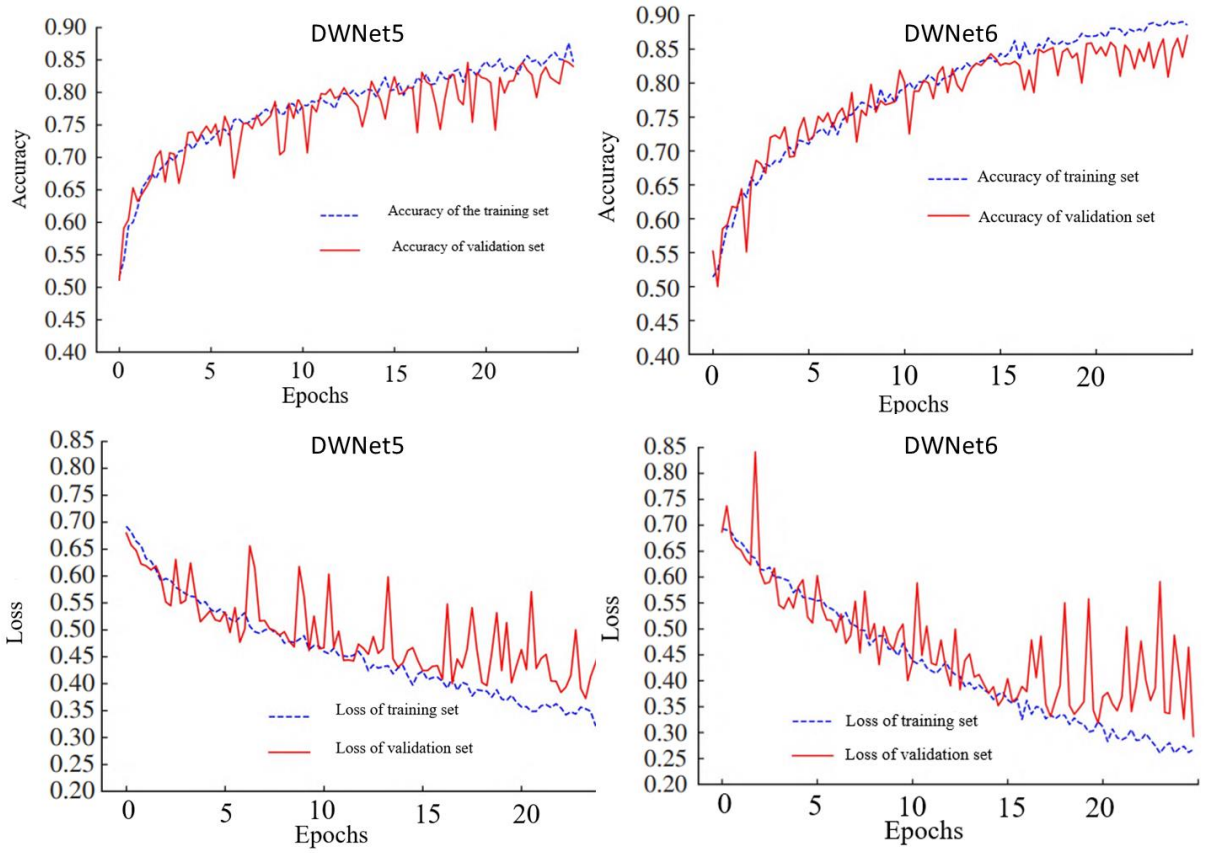

Figure 9. Accuracy and loss of DWNet5 and DWNet6 (Figure Credits: Original).

3.3. Visualization Result

The dataset is splinted visualized as shown in Figure 10 below. The fact that the dataset for this study has the same number of samples in each category as we can see suggests that the experimental dataset is balanced, which is helpful for enhancing the model's capacity for generalization. This balance aids in the model's ability to generalize. Additionally, each dataset also contains a sizable number of samples, which supports the model's capacity to learn and subsequently improves accuracy.

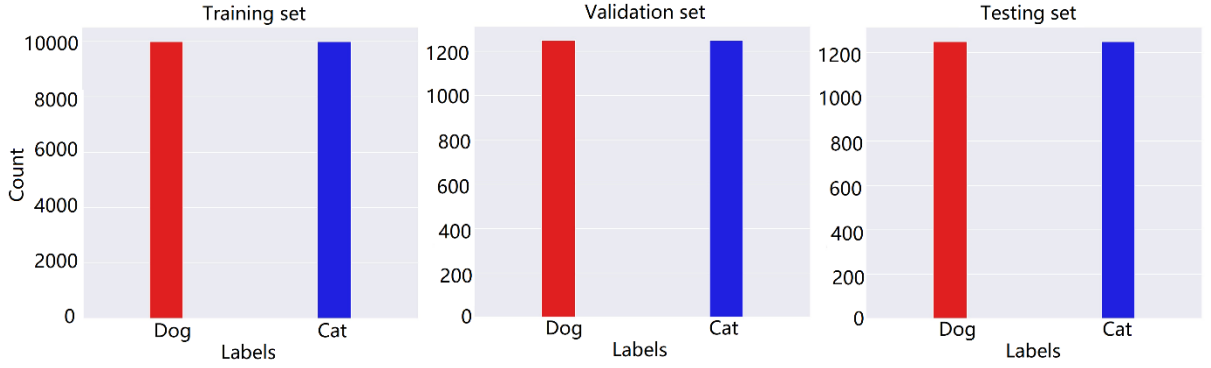

Figure 10. Training set test validation set visualization results (Figure Credits: Original).

This work evaluated the CNN model under this network structure of DWNet6, and finally used the confusion matrix to demonstrate the classification ability of this CNN model, which is shown in Figure

DOI: 10.54254/2755-2721/47/20241274

11 below. The confusion matrix demonstrates the DWNet6's accuracy reaches up to more than 95%, which is a very high level, indicating that the model has good prediction ability.

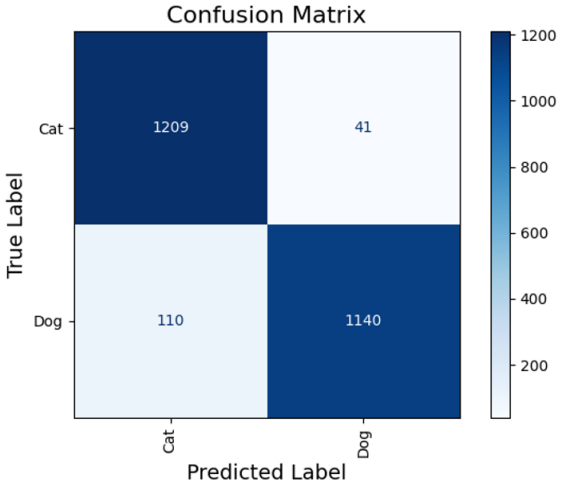

Figure 11. Visualization of the CNN model evaluation confusion matrix (Figure Credits: Original).

4. Discussion

4.1. Analysis of Model Strengths and Weaknesses

The CNN model designed in this paper achieves better results on the cat and dog image recognition task and has the following advantages over the classical CNN model: 1. To adapt to various feature inputs and classification requirements, the model's depth and width can be changed in accordance with various application scenarios and datasets. 2. The model uses special modules such as the BN module, the Inception module, and the Residual module to improve the model's convergence speed, stability, expressiveness, and adaptability. 3. In order to increase data diversity and robustness and reduce the likelihood of overfitting, the model makes use of data enhancement operations.

The CNN model designed in this paper also has some shortcomings that need to be further improved, mainly as follows: 1. the number of parameters of the model is still large, resulting in high computation and memory occupation of the model, which may affect the operation efficiency and deployment flexibility of the model. 2. the model negelects other information in the image, such as colors, shapes, textures, etc., which may result in the model being insensitive to some details or changes, and reduce the model's accuracy and generalization ability.

4.2. Prospects and Applications

The CNN model designed in this paper has some room for improvement and extension, and some new research directions and problems can be explored, such as: network structure optimization, which can try to use more advanced or innovative network structure design methods, such as Neural Architecture Search (NAS), Meta-Learning, Neural Network Distillation, etc. to improve the efficiency and effectiveness of the network structure. And this CNN model has some practical value in cat and dog image recognition task, which can be applied to some related fields and scenarios, such as: 1. Pet identification: It can identify the pet's breed, color, gender and other information according to the image, providing more convenience and services for pet owners. 2. Animal protection: it can recognize the species, number, state and other information of animals according to the image, providing more data and support for animal protection work. There are other application values due to the limited space in this paper will not be listed, in short, the CNN model has a considerable practical value and a broad application market.

DOI: 10.54254/2755-2721/47/20241274

5. Conclusion

The performance of deep CNN models is improved with deeper structure within a specific depth range. However, for some CNNs, the depth is too deep, and the gradient drop issue arises, resulting in poorer model performance and poorer prediction results; The parameters that determine the width of the CNN are the number of neurons and the size of the convolutional kernel, and the performance of the CNN model gets better as these parameters increase within a certain range. Beyond this range, the increase in width increases the computation required by the single-layer network, which increases the training and prediction time, decreasing the model's computational efficiency; however, due to the flexibility of the combined CNN, the model performance of the combined CNN is characterized by high prediction accuracy, low error, good fitting ability, and high generalization performance, and it performs better than single-layer CNNs.

For the actual problem, how to choose the appropriate depth and width of the CNN model and how to choose the combination of CNNs with good performance need to be tried manually, which is time-consuming and inefficient, and it is a great opportunity and challenge for future scholars in the field of CNN research.

References

[1] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4), 193-202.

[2] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

[3] Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. In 2014 13th international conference on control automation robotics & vision, 844-848.

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1-8.

[5] Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2017). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In 2017 IEEE international conference on intelligence and security informatics, 43-48.

[6] Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. In 2017 international conference on communication and signal processing, 0588-0592.

[7] Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. Frontiers in neuroscience, 10, 508.

[8] Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and dogs. In 2012 IEEE conference on computer vision and pattern recognition, 3498-3505.

[9] Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. Drug discovery today, 23(6), 1241-1250.

[10] Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. Large-scale kernel machines, 34(5), 1-41.

[11] Loukas, A. (2019). What graph neural networks cannot learn: depth vs width. arXiv preprint arXiv:1907.03199.

[12] Kaggle. dogs vs. cats. URL: https://www.kaggle.com/c/dogs-vs-cats/data. Last Accessed: 2023/09/27

[13] Gao, Y., Liu, W., & Lombardi, F. (2020). Design and implementation of an approximate softmax layer for deep neural networks. In 2020 IEEE international symposium on circuits and systems, 1-5.

[14] Mu, R., & Zeng, X. (2019). A review of deep learning research. KSII Transactions on Internet and Information Systems, 13(4), 1738-1764.