Union-net: lightweight deep neural network model suitable for small data sets

Jingyi Zhou¹ · Qingfang He² · Guang Cheng² · Zhiying Lin²

Accepted: 16 November 2022 / Published online: 25 November 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Traditional deep learning models prefer large data sets, and in reality small data sets are easier to obtain. It is more practical to build models suitable for small data sets. Based on CNN, this paper proposes the concept of union convolution to build a deep learning model Union-net that is suitable for small data sets. The Union-net has small model size and superior performance. In this paper, the model is tested based on multiple commonly used data sets. The experimental results show that Union-net outperforms most models when dealing with small datasets, and Union-net outperforms other models when dealing with complex classification tasks or dealing with few-shot datasets. The codes for this paper have been uploaded to https://github.com/yeaso/union-net.

Keywords Small data sets · Deep learning · Pre-trained model · Shallow network · CNN model

1 Introduction
In recent years, the research of deep learning based on small data sets has been paid more and more attention. In order to solve the problem that deep learning models use small data sets to easily produce overfitting, a variety of improved methods have been proposed. For example, augmenting data samples with image rotation, distortion, and more. For samples that are difficult to cluster in small data sets, the literature [1] proposes the concept of weak features, weak feature sample extraction algorithms, adaptive adjustment, secondary training algorithms and other methods...
to improve model training in small data sets. In view of the small number of samples in the medical data set and the large difference in the number of samples in each category, the paper [2] used model fusion, transfer learning, feature fusion technology, etc., to achieve good results in solving the multi-classification of breast cancer. The paper [3] proposes L1 and L2 regularization methods. This method prevents model overfitting by affecting weight updates during model training. Large-scale models use L1 regularization to reduce model complexity to prevent model overfitting, and models with few parameters use L2 regularization to prevent model overfitting. Paper [4] proposed the dropout method. Dropout is an effective method to prevent overfitting, but it also suffers from information loss. The literature [5] used an early stopping method. The early stopping method is to stop training when overfitting occurs, but it may be misjudged and stop early. Paper [6] proposed methods to reduce the number of model parameters, enhance data, and generate new data with adversarial networks. This approach is somewhat effective in preventing overfitting, but at the risk of degrading model performance. Paper [7] used methods such as transfer learning and fine-tuning. This method is currently widely used in small data set projects and works well when used correctly. If the pre-trained model is commercialized, there is a copyright risk. Paper [8] proposed a 3D deep learning model AppendixNet. The model was pre-trained on a large number of YouTube videos called Kinetics, and then fine-tuned on a small data set of 438 CT scans labeled for appendicitis. Paper [8] believes that pre-training the 3D model on a large natural video data set and then applying it to a specific small data set can improve the performance of the model.

The above research is mainly based on the optimization of model hyperparameters, model pre-training, and fine-tuning techniques to improve the model’s adaptability to small data sets. Most of the current research on deep learning applied to small data sets is focused on transfer learning (model pre-training and fine-tuning). That is, pre-training tasks with big data (such as the ImageNet data set [9]), using the pre-trained model to apply to small data sets, to solve tasks similar to big data tasks on small data sets. At present, transfer learning has become a common method to realize deep learning based on small data sets [10, 11]. However, there are two major problems in practical applications when using the transfer learning method: First, the used big data may have copyright issues. For example, the samples in the big data set may come from the internet, and the license for use is not clear. There are also copyright laws implemented in many countries/regions that consider it illegal to use ImageNet pre-trained models for commercial applications. Secondly, pre-trained models based on generally large data sets (such as ImageNet) are not a panacea. If the transfer learning scenario is very different between the source domain and the target domain, the results of machine learning are not ideal [12].

As we all know, given a large amount of data, even a simple learning model can solve complex tasks by memorizing [13, 14]. The hallmark of true artificial intelligence is that it can extract features from limited data. Paper [15] proposed to replace the commonly used cross-entropy loss function with the cosine loss function in the CNN model. Using this method, the performance of the model is improved on a data set with only a small number of samples in each category. It achieves the improvement in specific experiments by adjusting conventional algorithms and does not
have general applicability. Paper [16] proposed Compact Convolutional Transformers (CCT) model. The model outperforms traditional models on small data sets.

The definition of a small data set is currently not clearly described, it is related to the completed tasks and the diversity of the data. According to the paper [17], in the psychiatric human research, samples less than 10,000 are "small" samples. Paper [8] uses 438 samples labeled with appendicitis as a small data set. Paper [15] uses about 20 to 100 samples per classification as a small data set. For image classification, a data set with less than 100 training samples per category is very small. Accordingly, we define a small data set as follows: the number of samples in the data set is relatively small, or although the number of samples in the data set is large, it cannot fully characterize the properties of things.

Classical models are prone to overfitting when trained with small data sets. There may be copyright issues with big data used in practical applications. For tasks with small data sets, transfer learning (pre-trained models) methods are mostly used in practice. Pre-trained models based on big data may also face copyright issues if used in commercial applications. Therefore, this paper proposes the concept of union convolution, and designs a lightweight deep network model Union-net suitable for small data sets.

Contributions of this paper: (i) Propose a CNN model Union-net adapted to small data sets. (ii) The rationality of the model is deduced and verified. (iii) Propose the concept of union convolution. (iv) Test the model with multiple distinctive data sets (Small size grayscale image sample data set, small size color image sample data set, fine-grained image classification data set, more samples per class and less classification, fewer samples per class and more classification, fewer samples per class and less classification, etc., data sets.).

The experimental results show that Union-net outperforms most models when dealing with small datasets, and Union-net outperforms other models when dealing with complex classification tasks or dealing with few-shot datasets.

2 Model

2.1 The proposed union-net model

The Union-net model is designed based on convolutional neural network (CNN) technology [19]. The design inspiration of the model comes from three classic models: Resnet [20], Densenet [21], and Xception [22]. The residual concept of Resnet is adopted. The idea of feature reuse in Densenet is adopted. The channel merging method of multiple convolutions of the same input of Xception is changed to the information merging method of multiple different convolutions of the same input.

2.1.1 Union-net model structure

The structure of the Union-net model is shown in Fig. 1. Due to the limited space, the batch standardization layer [23] and the activation function layer [24] are omitted in Fig. 1. For detailed configuration, please refer to the model.
structure diagram (The model structure diagram has been uploaded to https://github.com/yeaso/union-net).

As can be seen from Fig. 1, the model consists of an input layer, three Union modules, a $3 \times 3 \times 256$ convolutional layer, a global average pooling layer [25] and a SoftMax classification output layer [26]. The " + " sign in each union module indicates the fusion and addition of the output results of the four groups of network units in the module. The fusion method used by Union-net is different from the fusion method used by the Xception [22] and Inception [27] models. Xception and Inception use channel splicing (called concatenate) [27]. Therefore, the shape of the feature map output by the Xception and Inception models is changed (that is, the number of channels increases). Union-net uses the addition of the information of multiple features. After the addition, the output shape of the feature does not change, and the number of channels remains unchanged. The output of Union1 is used as the input of Union2 module. The output of the Union2 module is used as the input of the Union3 module. The output of the Union3 module is merged and added with the output of Union1 and Union2, and the result of the addition is used as the input of the final convolutional layer of the model.

In order to further observe the feature output of the Union module in the model, a visualization tool is used to easily observe the features of Union1, Union2, Union3, and the union output of the three nodes. Take the model trained in this paper based on the 17flower data set [28] as an example to output visual features, as shown in Fig. 2. It can be seen that from the low-end to the high-end of the model, the output feature map gradually changes from a detailed expression to an abstract expression.
2.1.2 Union-net features

Judging from the model structure in Fig. 1, Union-net has the following characteristics:

1. From the appearance, Union-net is a simple network with a simple structure and a shallow network structure. There are three modules containing multiple convolutions and one convolutional layer (Union1, Union2, Union3, and \((3 \times 3, 256)\) conv).

2. All convolution layers use \(3 \times 3\) small convolution kernels to reduce model parameters and computational complexity. This model has a "shallow network" structure and uses a batch standardization layer [23], so the overfitting problem is solved. The dropout [4] used by the deep neural network model is not used here to prevent overfitting, because Union-net faces fewer data samples. If dropout is used, the
acquired information is randomly discarded during the training process, which will affect the model’s ability to fully capture sample features.

3. Union-net uses Maxpooling layer in the first union module [29]. According to related theories [29], the Maxpooling can reduce the deviation of the estimated mean value caused by the parameter error of the convolutional layer, and retain more texture information. The Maxpooling can retain more effective features, reduce dimensions, reduce parameters, and remove noise. These advantages are very beneficial for extracting features and suppressing overfitting. Therefore, Union-net uses the Maxpooling layer in the Union1 module. After the initial input data of the model is processed by Union1’s internal convolution structure, the output features are pooled and filtered to provide more effective data for subsequent processing. In the final stage of the model, the Maxpooling layer is not used. The main consideration is that the model structure is shallow, and increasing the pooling layer has the risk of losing local information. Instead, the traditional fully connected layer (dense layer) is replaced by global average pooling. This method comes from references [30, 31]. The dense layer parameters of the traditional convolutional neural network are huge. Replace dense with global average pooling, which greatly reduces the amount of parameters. This is helpful to suppress the overfitting of the model when training on a small data set.

4. The power of unity. Figure 3 shows the Union module structure. There are four groups of independent parallel convolutional neural network structures in the module. They are: 3 × 3 convolution kernel completes one layer of convolution (denoted as a); 3 × 3 convolution kernel completes two-layer stacked convolution (denoted as b); 3 × 3 convolution kernel completes three-layer stacked convolution (denoted as c); The 3 × 3 convolution kernel completes four-layer stacked convolution (denoted as d). The input of each group of convolution is the same, but each group has different receptive fields for the input features. Group a is the receptive field of the 3 × 3 convolution kernel. Group b is equivalent to the receptive field of the 5 × 5 convolution kernel. Group c is equivalent to the receptive field of the 7 × 7 convolution kernel. Group d is equivalent to the receptive field

![Fig. 3 Union module structure](image-url)
of the $9 \times 9$ convolution kernel [32]. Although the groups a, b, c, and d in the union module are simple convolutional neural network structures, they observe and extract features from different fields of view on the input features (shapes). The feature information obtained by them is added and merged, and the merged information is sent to the next union module. In the union module, the four groups of neural network units work individually, and the outputs of each group are combined together. From the perspective of the entire model, the outputs of the three union modules are combined together. This is also the origin of the model name Union-net.

5. Compared with the classic deep learning model, the Union-net model has fewer parameters and less floating point operations (FLOPs). Table 1 shows the comparison between the classic network model and the Union-net model in terms of model parameters, model FLOPs. It can be seen from Table 1 that the parameters of Union-net is much smaller than most classic networks, and is similar to the lightweight networks MobileNetV2 [31] and EfficientNet-B0 [33], and the FLOPs value of Union-net is much smaller than other models. Therefore, the inference time of Union-net is much less compared to other models under the same working environment.

### 2.1.3 Union convolution

This paper believes that the Union-net model can be regarded as a simple network structure. By analyzing the input and output among Union-net modules, this view is reasonable. In Fig. 3, there are four groups of convolutional neural network structural units a, b, c, and d, and each group has the same input $X$. a is a neural network structural unit of a convolutional layer, and its output $Y_a$ is expressed by formula (1). b is a neural network structural unit of 2 convolutional layers, and its output $Y_b$ is expressed by formula (2). c is a neural network structural unit of 3 convolutional layers, and its output $Y_c$ is expressed by formula (3). d is a neural network structural unit of 4 convolutional layers, and its output $Y_d$ is expressed

| Model                        | Parameters | FLOPs |
|------------------------------|------------|-------|
| Union-net                    | 1.72 M     | 0.003B|
| VGG16 [32]                   | 138 M      | 15.5B |
| Xception [22]                | 23 M       | 8.4B  |
| ResNet50 [20]                | 26 M       | 3.9B  |
| InceptionResNetV2 [34]       | 56 M       | 13B   |
| DenseNet169 [21]             | 14 M       | 3.5B  |
| MobileNetV2                  | 3.4 M      | 0.6B  |
| EfficientNet-B0 [33]         | 5.3 M      | 0.39B |
| ViT-12/16 [16]               | 85.63 M    | 0.002B|
| CVT-7/4 [16]                 | 3.72 M     | 0.002B|
| CCT-7/3 [16]                 | 3.76 M     | 0.56B |
by formula (4). Here \( w \) is the weight corresponding to the input \( X \), \( b \) is the bias parameter, \( \sigma \) is the activation function.

\[
Y_a = \sigma_{a1}(w_{a1} \ast X + b_{a1})
\]

(1)

\[
Y_b = \sigma_{b2}(w_{b2} \ast (\sigma_{b1}(w_{b1} \ast X + b_{b1})) + b_{b2})
\]

(2)

\[
Y_c = \sigma_{c3}(w_{c3} \ast (\sigma_{c2}(w_{c2} \ast (\sigma_{c1}(w_{c1} \ast X + b_{c1})) + b_{c2})) + b_{c3})
\]

(3)

\[
Y_d = \sigma_{d4}(w_{d4} \ast (\sigma_{d3}(w_{d3} \ast (\sigma_{d2}(w_{d2} \ast (\sigma_{d1}(w_{d1} \ast X + b_{d1})) + b_{d2})) + b_{d3})) + b_{d4})
\]

(4)

Formula (5) is a general expression of formulas (1)-(4). Where \( X_{i-1} \) represents the input of the previous layer of convolution, \( i \) represents the four groups of convolutional neural networks a, b, c, and d, and \( j \) represents the convolutional layer corresponding to each group of convolutional neural networks.

\[
Y_j = \sigma_{ij}(W_{ij} \ast X_{i-1} + b_{ij})
\]

(5)

It can be seen that each union module has only one input and one output. The working process of the standard neural network convolutional layer: accepts an input, and obtains an output through convolution kernel processing. The four groups of convolutional neural network structural units of the union module perform "complex convolution processing" when the union module accepts one input. The processing results of each group are added together as the output of the union module. Therefore, this paper regards each union module as a union convolution unit, which is called union convolution. According to this idea, the model structure of Fig. 1 is simplified as shown in Fig. 4. Obviously, it appears that Union-net is a "shallow" and simple deep neural network.

![Fig. 4 Union-net model simplified with union convolution](image-url)
3 Data sets

We conducted image classification experiments using our model on the following data sets: CIFAR-10 (ten categories), CIFAR-100 (100 categories) [15], MNIST (ten categories) [35], Fashion-MNIST (ten categories) [36] and 17-Flowers (17 categories) [28].

CIFAR-10 contains color images of ten categories, the size of the images is $32 \times 32$, and there are 5000 training images and 1000 testing images for each class in the data set.

The CIFAR100 data set has 100 classes. Each class has 600 color images of size $32 \times 32$, of which 500 are used as training set and 100 are used as test set.

The 17-Flowers data set has 17 categories of flowers. There are 80 color pictures of different sizes of each category of flower. It is a data set with a small number of samples.

The first four data sets not only have a small number of training samples, but they are also small in resolution. Additionally, MNIST and Fashion-MNIST only contain a single channel, greatly reducing the information density. The images of each category of 17-Flowers have large pose and lighting variations, and images of different categories are also very similar. 17-Flowers belongs to the fine-grained classification task data set.

This paper tests Union-net on these diverse and distinctive datasets to verify the performance of the designed model.

4 Experimental settings

Test environment requirements for the Union-net model in this article: python3.6 + keras2.3.1 + opencv-python 4.5.5.62.

The main hyperparameters used in our experiments are:

- batch_size = 128.
- epochs = 100.

Optimizers use Nadam. Nadam parameter settings ($lr=0.01$, $beta_1=0.5$, $beta_2=0.999$, $epsilon=1e-08$, schedule_decay=0.004).

Learning rate adjustment strategy: monitor='val_accuracy', factor=0.5, patience=2, verbose=1, mode='min', epsilon=0.0001, cooldown=0, min_lr=0.

For other detailed configurations, see the uploaded code files. The full codes of the Union in this article have been published at https://github.com/yeaso/union-net.
5 Results and compare

The source programs used in these experiments are 5 python files and pre-trained model files (.h5 format) generated based on different data sets, which have been uploaded to github. Except that 17-Flowers needs to configure the data set, the model experiments based on the other four data sets only need to run the given code directly.

This paper evaluates Union-net on the CIFAR-10, CIFAR100 [28], MNIST, Fashion-MNIST and 17-Flowers [18] data sets. Based on the first four datasets, the paper [16] provides the results (testing accuracy) on the classic model, Vision Transformers (ViT) model, Compact Vision Transformers (CVT) model and Compact Convolutional Transformers (CCT) model. The results (testing accuracy) of Union-net are compared with these results (testing accuracy), as shown in Table 2.

From the experimental results, Union-net is better than most classical models in dealing with small data sets, and slightly worse than the CCT model. From Table 2, we find that Union-net outperforms most models when training on two relatively difficult data sets (CIFAR100, Fashion-MNIST). Furthermore, it achieves 86.58% accuracy in the fine-grained classification (17-Flowers) test.

6 Ablation experiment

In order to further verify whether the structural combination of the Union modules of the Union model is reasonable, based on the 17flower data set, split experiments and expansion experiments were performed on the union modules. As can be seen

| Model                  | CIFAR-10 | CIFAR-100 | Fashion-MNIST | MNIST | 17-Flowers |
|------------------------|----------|-----------|---------------|-------|------------|
| Union-net(ours)        | 91.79    | 74.69     | 96.88         | 98.67 | 86.58      |
| ResNet18 [16]          | 90.27    | 66.46     | 94.78         | 99.80 | –          |
| ResNet34 [16]          | 90.51    | 66.84     | 94.78         | 99.77 | –          |
| MobileNetV2/0.5 [16]   | 84.78    | 56.32     | 93.93         | 99.70 | –          |
| MobileNetV2/2.0 [16]   | 91.02    | 67.44     | 95.26         | 99.75 | –          |
| ResNet56 [16]          | 94.63    | 74.81     | 95.25         | 99.27 | –          |
| ResNet110 [16]         | 95.08    | 76.63     | 95.32         | 99.28 | –          |
| ViT12/16 [16]          | 83.04    | 57.97     | 93.61         | 99.63 | –          |
| ViT-Lite-7/16 [16]     | 78.45    | 52.87     | 93.24         | 99.68 | –          |
| ViT-Lite-7/8 [16]      | 89.10    | 67.27     | 94.49         | 99.69 | –          |
| ViT-Lite-7/4 [16]      | 93.57    | 73.94     | 95.16         | 99.77 | –          |
| CVT-7/8 [16]           | 89.79    | 70.11     | 94.50         | 99.70 | –          |
| CVT-7/4 [16]           | 94.01    | 76.49     | 95.32         | 99.76 | –          |
| CCT-2/3 x 2 [16]       | 89.75    | 66.93     | 94.08         | 99.70 | –          |
| CCT-7/3 x 2 [16]       | 95.04    | 77.72     | 95.16         | 99.76 | –          |
| CCT-7/3 x 1 [16]       | 96.53    | 80.92     | 95.56         | 99.82 | –          |
from the structure of the Union module in Fig. 3, the Union module is composed of four groups of convolutional neural network units a, b, c, and d, so it is divided into the following experimental groups:

\[ \text{U}_a, \text{U}_b, \text{U}_c, \text{U}_d, \text{U}_{ab}, \text{U}_{ac}, \text{U}_{ad}, \text{U}_{bc}, \text{U}_{bd}, \text{U}_{cd}, \text{U}_{abc}, \text{U}_{abd}, \text{U}_{acd}, \text{U}_{bcd}. \]

The expansion experiment is to expand the four groups of convolutional neural network units a, b, c, and d of the Union module into five groups of convolutional neural network units a, b, c, d, and e, where e is a 3×3 convolution kernel Five-layer stacked convolution (denoted as e), this experimental group is denoted as \( \text{U}_{abcde} \).

Note: \( \text{U}_a \) means that the Union module is composed of one group of convolutional neural network units a; \( \text{U}_{abc} \) means that the Union module is composed of three groups of convolutional neural network units a, b, and c. The names of other experimental groups indicate similar meanings.

The parameter settings are exactly the same as Sect. 4, and the above 15 experimental groups were tested with the same computer.

The model experiment results (testing accuracy) of 15 experimental groups are shown in Table 3.

In order to facilitate the observation of the effect of different combinations of union convolution modules on the model performance, the data (testing accuracy) in Table 3 are represented by a line graph, as shown in Fig. 5.

It can be seen from Fig. 5 that except for the \( \text{U}_{ac} \) and \( \text{U}_{ad} \) groups, the differences between the other split experimental groups are not very large. The performance of the two groups of \( \text{U}_{ac} \) and \( \text{U}_{ad} \) lags behind \( \text{U}_a, \text{U}_c, \text{U}_d \), indicating that \( \text{U}_a \) combined with \( \text{U}_c \) or with \( \text{U}_d \), their combined results lag behind \( \text{U}_a, \text{U}_c, \text{U}_d \). \( \text{U}_{ab} \) is better than \( \text{U}_a \) and \( \text{U}_b \). \( \text{U}_{bc} \) is better than \( \text{U}_b \) and \( \text{U}_c \). \( \text{U}_{bd} \) is better than \( \text{U}_b \) and \( \text{U}_d \). The combination of \( \text{U}_{cd} \) lags behind \( \text{U}_c \) and \( \text{U}_d \). From the results, without the participation of one-layer convolution a or two-layer convolution b, the effect of the combination is reduced. The results of \( \text{U}_{abc}, \text{U}_{abd}, \text{U}_{acd}, \) and \( \text{U}_{bcd} \) show that the combination of any three of a, b, c, and d becomes stable, and the combined result no longer lags behind the single \( \text{U}_a, \text{U}_b, \text{U}_c, \) and \( \text{U}_d \). \( \text{U}_{abcde} \) and Union-net test results are the best. Although \( \text{U}_{abcdef} \) has one more e than Union-net, the experimental results are not significantly improved compared to Union-net. The structure of \( \text{U}_{abcdef} \) model is much more complicated than Union-net, and the convolution calculation amount of each Union module of \( \text{U}_{abcdef} \) has changed from the original \( a+b+c+d+e=1+2+3+4=10 \) to \( a+b+c+d+e=1+2+3+4+5=15 \). Therefore, this paper chooses the \( \text{U}_{abcd} \) structure as the union convolution layer of Union-net. Of course, reasonable extensions to Union-net according to specific applications will also improve the performance of the model.

7 Further verification

In order to verify the applicability of the Union-net model, experiments on the model are based on MiniImageNet [37], and CUB200_2011 data set [38]. The method of making the training set is as follows: (i) MiniImageNet contains 60,000 color images in 100 categories, each of which has 600 samples. Randomly select ten categories from its 100 categories, each category has 100 training samples, and a total of three data sets.
### Table 3  Comparison table of split experiment, expanded experiment results (testing accuracy) and Union model results based on the 17flower data set

|    | U_a  | U_b  | U_c  | U_d  | U_{ab} | U_{ac} | U_{ad} | U_{bc} | U_{bd} | U_{cd} | U_{abc} | U_{abd} | U_{acd} | U_{bcd} | U_{abcde} | Union    |
|----|------|------|------|------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|-----------|-----------|
|    | 0.8015 | 0.8051 | 0.8125 | 0.8088 | 0.8235  | 0.7794  | 0.8199  | 0.8235  | 0.8051  | 0.8125  | 0.8125  | 0.8162  | 0.8235  | 0.8665  | 0.8658    |           |
are obtained, denoted as MIN (R1), MIN (R2), and MIN (R3). (ii) The CUB200_2011 data set has a total of 200 categories, each with about 60 samples, and a total of 11,788 pictures. It is a fine-grained classification data set about birds. Ten categories were randomly selected from its 200 categories, and a total of three data sets were obtained, denoted as CUB (R1), CUB (R2), and CUB (R3). The experimental equipment and parameter configuration are consistent with the aforementioned ablation experiment. The measured experimental results (testing accuracy) are shown in Table 4.

Judging from the test results in Table 4, the Union-net model performs well overall. Union-net also performs well in small datasets with few samples. The model needs further improvement in fine-grained classification. In particular, Union-net also performs well in small datasets with few samples. This also verifies that the model designed in this paper based on the union convolution module is reasonable and achieves the expected effect.

### 8 Conclusions and future works

This paper proposes the concept of Union convolution. The model based on Union convolution has the characteristics of simple structure, few parameters and strong feature extraction ability. The experimental results show that Union-net outperforms most models when dealing with small datasets, and Union-net outperforms other models when dealing with complex classification tasks or dealing with few-shot datasets. This result also verifies that the union convolutional layer (Union module) can process

| Data set | MIN (R1) | MIN (R2) | MIN (R3) | CUB (R1) | CUB (R2) | CUB (R3) |
|----------|----------|----------|----------|----------|----------|----------|
| Accuracy | 93.57    | 92.04    | 94.01    | 89.24    | 90.33    | 88.96    |
information better than the conventional convolutional layer. Union-net is based on CNN, and the fields which are suitable for CNN are theoretically suitable for Union-net. How to add the idea of union convolution in the fields touched by RNN and LSTM is the future research direction. The source codes of this paper has been released, and we hope peer experts can give us some suggestions and discover problems with the model to further improve the performance of the model.

**Author contributions** JZ contributed to software, validation, and data curation. QH contributed to writing—original draft preparation, software, writing—reviewing and editing, conceptualization, and methodology. GC contributed to supervision, funding acquisition, and project administration. ZL contributed to visualization and investigation.

**Funding** This work was supported by Beijing Natural Science Foundation (No.L191006) and Academic Research Projects of Beijing Union University (No.XP202021).

**Data availability** The codes of the program supporting the results of this study has been uploaded to github. The URL is https://github.com/yeaso/union-net.

**References**

1. He Q, Wang H, Cheng G (2021) Research on classification of breast cancer pathological tissues with adaptive small data set. Comput Sci 48(6A):67–73
2. Qingfang H, Guang C, Huimin J (2021) BCDnet: parallel heterogeneous eight-class classification model of breast pathology. PloS One 16(7):e0253764. https://doi.org/10.1371/journal.pone.0253764
3. Wu G, Hu H, Yu M (2020) Research on regularization methods in deep learning. Comput Sci Appl 10(6):10
4. Srivastava N et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
5. Jz A, Min ZA, and Sbl B (2022) Fully corrective gradient boosting with squared hinge: fast learning rates and early stopping
6. Yinan Y et al (2019) Research on small sample data generation technology based on generative adversarial network. Electr Power Constr 40(05):71–77
7. Afridi et al (2018) On automated source selection for transfer learning in convolutional neural networks. Pattern Recog J Pattern Recogn Soc
8. Rajpurkar P, Park A, Irvin J et al (2020) AppendiXNet: deep learning for diagnosis of appendicitis from a small data set of CT exams using video pretraining. Sci Rep 10:3958. https://doi.org/10.1038/s41598-020-61055-6
9. Deng J, Dong W, Socher R, Li L-J, Li K, and FeiFei L (2009) ImageNet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE. Vol 1, pp. 248–255
10. Chazhoor A et al (2022) Deep transfer learning benchmark for plastic waste classification. Intell Rob 2(1):1–19
11. Hu SX et al (2022) Pushing the limits of simple pipelines for few-shot learning: external data and fine-tuning make a difference. doi: https://doi.org/10.48550/arXiv.2204.07305
12. Yosinski J, Clune J, Bengio Y et al (2014) How transferable are features in deep neural networks?. In: international conference on neural information processing systems. MIT Press
13. Torralba A, Fergus R, Freeman WT (2008) 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans Pattern Anal Mach Intell (TPAMI). 30(11):1958–1970
14. Zhang C, Bengio S, Hardt M, Recht B, and Vinyals O (2017) Understanding deep learning requires rethinking generalization. In: International conference on learning representations (ICLR). 01
15. Barz B and Denzler J (2020) Deep learning on small data sets without pre-training using cosine loss. In: 2020 IEEE winter conference on applications of computer vision (WACV). Snowmass. CO. USA, pp. 1360–1369. doi: https://doi.org/10.1109/WACV45572.2020.9093286
16. Ali H, Steven W et al (2022) Escaping the big data paradigm with compact transformers doi: https://doi.org/10.48550/arXiv.2104.05704
17. Koppe G, Meyer-Lindenberg A, Durstewitz D (2020) Deep learning for small and big data in psychiatry. Neuropsychopharmacology 46:176–190
18. Nilsback ME, and Zisserman A (2008) Automated flower classification over a large number of classes. In: Sixth Indian conference on computer vision. Graphics & image processing. ICVGIP 2008. Bhubaneswar. India. IEEE. Vol 12, pp. 16–19
19. Zhou F, Jin L, and Dong J (2017) A survey of convolutional neural networks. Chin J Comput 06
20. He K et al (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision & pattern recognition. IEEE Comput Soc
21. Huang G, Liu Z, Laurens V et al (2016) Densely connected convolutional networks. IEEE Comput Soc
22. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR) IEEE
23. Ioffe S & Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. https://arxiv.org/abs/1502.03167
24. Chieng HH, Wahid N, Ong P (2020) Parametric flatten-T swish: an adaptive nonlinear activation function for deep learning. J Inf Commun Technol 20(1):21–39
25. Arkah ZM, and Alzubaidi LS (2020) Convolutional neural network with global average pooling for image classification. In: International conference on electrical, communication, electronics, instrumentation and computing (ICECEIC)
26. Wan L et al (2019) Summary of application of softmax classifier deep learning image classification method. Navig Control
27. Szegedy C et al (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). Boston, MA. USA. pp. 1–9
28. Nilsback ME, and Zisserman A (2008) Automated flower classification over a large number of classes. In: Sixth Indian conference on computer vision, graphics & image processing. ICVGIP 2008. Bhubaneswar. India. IEEE. vol 12, pp. 16–19.
29. Hamker FH (2018) Predictions of a model of spatial attention using sum and max-pooling functions. Neurocomputing 56:329–343
30. Lin M, Chen Q, and Yan S (2013) Network in network. Comput Sci
31. Sandler M, Howard A, Zhu M et al (2018) Inverted residuals and linear bottlenecks: mobile networks for classification detection and segmentation. In: IEEE conference on computer vision and pattern recognition (CVPR). pp. 4510–4520
32. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Sci
33. Tan M, and Quoc L (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR
34. Szegedy C, Ioffe S, Vanhoucke V et al (2016) Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: AAAI conference on artificial intelligence
35. LeCun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
36. Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image data set for benchmarking machine learning algorithms
37. Vinyals O, Blundell C, Lillicrap T et al (2016) Matching networks for one shot learning. NIPS. doi: https://doi.org/10.48550/arXiv.1606.04080
38. Wah C, Branson S, Welinder P et al (2011) The Caltech-UCSD Birds-200–2011 Data set. California Institute of Technology

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.