One-dimensional DCNN Feature Selective Transformation with LSTM-RDN for Image Classification

Chaorong Li, Yuanyuan Huang, Wei Huang, Fengqing Qin

Abstract—Feature selection and transformation are the important techniques in machine learning field. A good feature selection / transformation method will greatly improve the performance of classification algorithm. In this work, we proposed a simple but efficient image classification method which is based on two-stage processing strategy. In the first processing stage, the one-dimensional features are obtained from image by transfer learning with the pre-trained Deep Convolutional Neural Networks (DCNN). These one-dimensional DCNN features still have the shortcomings of information redundancy and weak distinguishing ability. Therefore, it is necessary to use feature transformation to continue to obtain more distinguishable features. We propose a feature learning and selective transformation network based on Long Short-Term Memory (LSTM) combing ReLU and Dropout layers (called LSTM-RDN) to further process DCNN one-dimensional features. . The verification experiments were conducted on three public object image datasets(Cifar10, Cifar100 and Fashion-MNIST), three fine-grained image datasets(CUB200-2011, Stanford-Cars, FGVC-Aircraft) and a COVID-19 dataset. In the experiments, we used several backbone network models, including AlexNet, VGG16, ResNet18, ResNet101, InceptionV2 and EfficientNet-b0. Experimental results have shown that through feature selective transformation, the recognition accuracy of these DCNN models can significantly exceed the classification accuracies of the state-of-the-art methods.

Index Terms—Feature Selective Transformation, LSTM, Image Classification, COVID-19, Deep Convolutional Neural Network.

I. INTRODUCTION

Machine vision is playing an increasingly important role in human production and life, and it has been driving the development of artificial intelligence (AI). The image classification task in machine vision is the basis of target detection, semantic

segmentation and scene understanding. Its importance is self-evident, but the current machine classification technology is still at a low level.

Filters and descriptors[1] are two basic techniques for extracting image features. The former mainly uses linear transformation of pixels in a local area of the image to obtain more prominent image information. These filters include median filters, wavelet filters, etc.; descriptors mainly perform more complex calculations on the pixels within the local area of the image, and these calculations include the description of the local image geometric information or statistical information, such as LBP[2] and SIFT[3] descriptors. Due to the complexity of the image, the local features of the aforementioned hand-crafted descriptor or filter are mainly determined by the pixels in the local area of the feature point, and its performance is very limited. In recent years, research has shifted from handcrafted descriptors and filters to descriptors and filters based on machine learning. The learning-based descriptors and filters collect a local area or patch with a pixel as the center, and then learn the features through machine learning. Dictionary learning[4],[5], and neural network learning[6] are two types of learning-based descriptors.

Filters and descriptors focus on the local features of the image, and then use statistical techniques such as histograms to get the overall features, while the method based on Deep Convolutional Neural Network (DCNN) directly extracts the overall features from the whole image. The advantage of this direct overall feature extraction method is that it can implement an end-to-end learning process, and it is more convenient to extract the image features and has better performance. AlexNet[7] is a classic shallow and single-branch convolutional neural network. It defeated traditional machine learning methods and won the championship on the ImageNet dataset in 2012. As application requirements increase, this shallow and single-branch neural network is increasingly unable to meet actual requirements. Since then, various effective network models have been designed to overcome the difficulties encountered in image classification. At present, the neural network structure is developing in the direction of more network branches (horizontal) and deeper network layers (vertical). The vertical development of the network focuses on using deeper layers to obtain better results. The original AlexNet only had more than 20 layers, to the dozens of layers in the later VGG model[8], and to the thousands of layers in the current NasNet[9]. The horizontal development of the network is not as fast as the deepening; it mainly increases

This work is supported by Project of Sichuan Science and Technology Program (No.2019YFS0068), Key projects of Sichuan Provincial Department of Education (No.18ZA0538), Projects of Yibin science and Technology Bureau(No.2017ZSF009-9), Research projects of Yibin University(Nos.2020YY02,2021YY06). (Corresponding author: Chaorong Li)

C.R. Li is with the School of Computer, Chongqing University of Technology(e-mail: lichaorong@cqu.edu.cn).

Y.Y. Huang is with the Department of cyberspace engineering , chengdu university of information technology, Chengdu 610225, China (e-mail: iyy-huang@hotmail.com).

W. Huang is with The Clinical Hospital of Chengdu Brain Science Institute, MOE Key Lab for Neuroinformation, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: lembert1990@163.com).

F. Qin is with the Department of Computer Science and Information Engineering, Yibin University, Yibin 644000, China (e-mail: qinfengqing@163.com).

Manuscript received xxxx xx, 2020; revised xxxx xx, 2020.

the branches of the network to extract a wider range of image features; the models have more branches including the residual structure of the ResNet[10], WideResNet[11] and the branch structure of the Inception networks[12], [13], [14].

The DCNN image classification performance has reached a bottleneck, and it is very difficult to improve the recognition performance. Recently, new visual models appeared such as Transformer and MPL-based model. Transformer[15] used in the natural language processing field, has been introduced into the field of machine vision and received great attention. ResMLP[16] is a simple network model and achieves an unexpectedly high performance on ImageNet classification benchmarks. ResMLP has an architecture built entirely upon multi-layer perceptrons embedded into the residual network.

Super-deep (DCNN) and multi-branched network structure make the model more demanding on the machine hardware. It usually takes several days or even dozens of weeks to train a large model; but the performance improvement is indeed insignificant, at least not achieving our expectation. Is this kind of super-deep structure necessary? Researchers consider using Transformer or Attention mechanism[17] to avoid the disadvantages of super-deep network model. Different from this measure, we use two-stage processing strategy to overcome the problem of huge parameters, so as to improve the performance of network model. The proposed method only needs the one-dimensional features of the DCNN, such as ResNet18, and then uses the feature selective transformation techniques and fine-tunes the model parameters, to obtain promising results which are much higher than the recognition results of other super-deep DCNN and Transformer models.

Our experiments show that there is no significant performance improvement between the shallow neural network and the super-deep neural network after feature selective transformation at most of cases, which is also one of the main contributions of this article. In addition, because there is correlation between DCNN features, we regard the one-dimensional DCNN output features as vector sequences, and use Short-Term Memory (LSTM) to perform feature transformation to make the features have better distinguishing capabilities; at the same time, we also added ReLU and Dropout functions in the network to perform feature selective transformation for improving the performance.

II. RELATED WORK

A typical classification task of machine learning is to predict the labels through sample images. The first step of classification task is to perform the feature extraction which is to convert the sample image into a set of features with obvious physical meaning such as geometric and texture features, or statistical significance(see Fig.1). If the samples have few features, we will consider adding more features. In reality, there are often too many features, and some of them need to be reduced. Feature transformation and selection plays an important role in improving the performance of models. Feature transformation and selection is to find the most effective features (in-variance of similar samples, discrimination of different samples, robustness to noise) from the original features. They not only

Fig. 1. Feature extraction, feature transformation and feature selection.

reduce overfitting, the runtime and improve the generalization ability of the model, but also enables the model to obtain better interpret-ability and enhance the understanding content of the images.

Typical feature extraction algorithms include Gabor transformation[18], HOG[19] and LBP descriptors, fully connected network[20] and DCNN which is used to extract the abstract features of image. Feature transformation methods are relatively extensive, and involve most of the machine learning and pattern recognition approaches. Some feature extraction methods can also be used as feature transformation like BP network and DCNN. In addition, Recurrent Neural Network (RNN)[21] is also commonly used for image feature transformation.

Feature selection refers to selecting N significant features from the existing M features to optimize the specific indicators of the system. Principal component analysis (PCA)[22] and linear discriminant analysis (LDA)[23] and Deep Forest (DF)[24] are the effective methods for feature selection.

It should be emphasized that the three technologies of feature selection, extraction, and transformation are sometimes difficult to distinguish. In many cases, the same model (such as DCNN) can be used for feature extraction and feature transformation; PCA and LDA also use feature selection operation when performing feature transformation. For the sake of distinction, this paper calls the method of finding the optimal features by using two techniques at the same time as feature selective transformation.

In recent years, LSTM network have emerged for image classification and other processing. It is a time cyclic neural network, which is specially designed to solve the long-term dependency problem of general RNN (Recurrent Neural Network). RNN has a chain form of repeated neural network modules. In a standard RNN, this repeated structural module is a very simple, such as a Tanh layer, while the repeating module of LSTM is more complex and much more effective. The basic cell of LSTM is shown in Fig.2. Since LSTM is a kind of RNN, its output is affected by the current input x_t , the previous input h_{t-1} (hidden units), and the influence of the cell state C_{t-1} . Overall, LSTM consists of three gates. Three gates (forget gate, input gates and output gate) combine these inputs and control the output. The forget gate is expressed as

follows

Fig. 2. Internal structure of LSTM cell.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

There are two important parts in the input gates which will update the cell state, as follows.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C} &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \quad (2)$$

The new cell state C_t is expressed as

$$C_t = f_t * C_{t-1} + i_t * \tilde{C} \quad (3)$$

The final output h_t (or y_t) is the result of multiplying the output o_t of the output gate and the current cell state C_t

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (4)$$

In image analysis, LSTM can be regarded as a powerful tool to enhance DCNN. Although DCNN has achieved great success, its performance is still difficult to meet the requirements in practical applications. Recently, there have been many examples of DCNN combined with LSTM. For example, [25] proposes handwriting recognition based on Convolutional-LSTM Network; [26] proposes Tree-Structured Regional CNN-LSTM model for dimensional sentiment analysis. In the hybrid combination, LSTM is mostly placed at the end of the network or used as the intermediate processing unit of the network.

III. LSTM-BASED SELECTIVE FEATURE TRANSFORMATION NETWORK

The process of the proposed one-dimensional DCNN feature selection approach based on LSTM is shown in Fig.3. The method uses a pre-trained network model as the backbone model, such as ResNet, and then train the pre-trained network model on a specific image dataset by using transfer learning to obtain the specified network model. By using the specified network model, we can obtain the two new one-dimensional DCNN datasets: train set and test set. Finally, use the feature selection network based on LSTM, ReLU and Dropout layers (LSTM-RDN) to train on new dataset and classify the test images on the new test set. LSTM-RDN roughly consists of two parts, as shown in Fig.4: The first part is the feature transformation component which is a LSTM layer. We regard

Fig. 3. The flowchart of the selective feature transformation for image classification.

the one-dimensional DCNN features as a sequence, and there is a strong correlation between the features. This correlation can be memorized by LSTM to optimize the transformation weights of the current input features. The second part is a feature selection component composed of a ReLU layer and several Dropout layers. This part first uses ReLU to perform nonlinear mapping on the output of the LSTM layer, and then uses Dropout layers to discard some features.

The two parts of the network are mainly to improve the discriminability of DCNN features as much as possible. The following experiments show that if only LSTM layer is used without the feature selection part (ReLU and Dropout layers), the model's performance will be greatly reduced (See the experiment and discussion part for details). At the front of LSTM-RDN is the Sequence layer, which is used to organize the serialized data, followed by a Dropout layer which is used to randomly discard some features before the DCNN features are fed to the LSTM layer.

LSTM feature transformation has the similar function as PCA, LDA, Dictionary Learning and fully connected networks, all of which can transform DCNN feature vectors into new feature vectors with more distinguishing performance. PCA / LDA needs to compute the divergence matrix of the all the DCNN feature vectors in the training set at one time; however, the performance is limited because of the lack of learning mechanisms. Fully connected network / dictionary learning obtains the transform model / dictionary by sending the samples to the model one by one or batch by batch with learning approach, but does not consider the correlation between the feature sequences. The characteristic of LSTM feature transformation is that LSTM processes one-dimensional DCNN features as time series, so it inputs these features one by one to train the model (see Fig.5), and at the same time it computes the output with the current and previous states of the model.

The schematic diagram of the feature selection part composed of a ReLU and several Dropout layers is given in Fig. 6. Rectified Linear Unit(ReLU) is a commonly used activation function in artificial neural networks, usually refers to the non-linear function represented by the ramp function and its variants. The reason why ReLU is very useful is that it makes the output of some neurons zero, which leads to the sparsity of feature sparsity (or network). That is, ReLU can let each neuron perform the screening function: if the input of neuron is larger than the given value, then the input will be enlarged

Fig. 4. LSTM-RDN for selective feature transformation. LSTM-RDN contains a LSTM, the ReLU and Dropout layers. The part of selective feature transform consists of the ReLU layer and the three Dropout layers (D0,D1 and D2) .

Fig. 5. The differences between LSTM feature transform and other methods.

vigorously; if not it will be cut off, denoted by

$$f(x) = \max(0, z_i), \quad (5)$$

where z_i is the input of neuron in the ReLU layer. ReLU and Dropout implement similar functions in different ways. Dropout is to randomly drop some nodes (or weight parameters) to participate in network training, so as to obtain more robust new features. Since both ReLU and Dropout have the function of zeroing or discarding a certain feature, we call their function here as selective feature transformation. "Attention" mechanism which is a popular technique uses weight to amplify or weaken the value of certain input variables or the block of feature map [27], and the feature selection method we proposed LSTM-RDN which has ReLU and Dropout layers also has similar select functions as Attention mechanism.

The only parameter of the Dropout layer is the drop probability, which plays a very important role. Probability for dropping out input elements, specified as a numeric scalar in the range [0,1]. In LSTM-RDN, we set up a total of three Dropout layers before and after the LSTM layer. In LSTM-RDN, we set up a total of three Dropout layers (called D0, D1 and D2 respectively) before and after the LSTM layer. Theoretically, D1 and D2 can be combined into one Dropout. Suppose the dropping probabilities of D1 and D2 are p_1 and p_2 , respectively, and their combined dropping probability p is

$$p = 1 - (1 - p_1) \times (1 - p_2), \quad (6)$$

Because the drop probability is a very important parameter in LSTM-RDN, its subtle changes may cause large fluctuations

Fig. 6. The detail of feature selection based on ReLU and Dropout layers.

in the recognition performance of the model. Therefore, we split a dropout layer into several layers for the purpose of fine-tuning the parameters. In fact, we can divide a dropout into multiple dropouts, that is to say dropping probability $p = \prod_{i=1}^n p_i$, where n is the number of splits.

IV. EXPERIMENTS

The experiment was conducted on two types of image datasets and COVID-19 prediction. The first type is object datasets, including Cifar10, Cifar100 and fashion-MNIST. The second type is Fine-Grained image datasets, including CUB200-2011, Stanford Cars and FGVC-Aircraft. Some samples selected from these datasets are shown in Fig.7 and Fig.10

The backbone network models we used include: AlexNet, VGG16, ResNet18, ResNet101, InceptionV2 and EfficientNet-b0. These pre-trained models are all trained on ImageNet. After downloading these models, we train them with transfer learning on the above datasets, respectively, to obtain the 1D DCNN features of the images. To object and fine-grained image recognition, in the feature selective transformation phase, Root Mean Square Prop (RMSProp) optimization algorithm is used to train LSTM-RDN; the initial learning rate is 0.001, and the maximum number of initial iterations is set to 2000 (if the mini-batch loss is not less than 0.01 or the mini-batch recognition accuracy is less than 100%, we will increase the

Fig. 7. Samples selected from Cifar10(the first row), Cifar100(the second row) and Fashion MNIST(the third row)

Fig. 8. Samples selected from CUB200-2011(the first row), Stanford-Cars(the second row) and FGVC-Aircraft(the third row)

number of training appropriately). To COVID-19 prediction, because the complexity and small number of CT images, we use fine-tuning approach to train LSTM-RDN model. Due to the randomness in LSTM-RDN training, the results of several tests may be different, and the classification accuracy of a model obtained by the experiment is a relatively stable result in several tests. Our experiments were performed on a computer with CPU i9 and GPU 2080ti environment.

A. Object Classification

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Cifar100 has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. Fashion-MNIST dataset contains 10 categories of images. The training dataset contains 6000 samples for each category, the test dataset contains 1000 samples for each category, and there are 10 categories in total.

After the pre-trained model is performed with transfer learning on target dataset, we compute the recognition accuracies by using the transfer-learned models which classify the images with softmax layer, and compute the recognition accuracies of our proposed LSTM-RDN based on the 1D features of the transfer-learned models. The recognition accuracies are listed in Table I. On the three datasets of Cifar10, Cifar100 and Fashion-MNIST, the recognition accuracies of our method on three datasets are higher than 98%, and the highest is 99.98% on Fashion-MNIST. The performance of these pre-trained models has been significantly improved after using

TABLE I
RECOGNITION RESULTS ON OBJECT DATASETS

Dataset	DCNN	Transfer	LSTM-RDN	Imp.
Cifar10	AlexNet	85.81	99.93	14.12
	VGG16	88.06	99.94	11.88
	ResNet18	90.44	99.91	9.47
	ResNet101	93.36	99.82	6.46
	InceptionV2	93.58	99.93	6.19
	EfficientNet-b0	89.95	99.39	9.44
Cifar100	AlexNet	62.33	96.57	34.24
	VGG16	63.27	98.21	34.94
	ResNet18	64.01	95.65	31.64
	ResNet101	66.97	97.33	30.36
	InceptionV2	68.28	98.17	29.89
	EfficientNet-b0	64.6	95.29	30.69
Fashion-MNIST	AlexNet	94.41	99.98	5.57
	VGG16	94.91	99.95	5.04
	ResNet18	94.37	99.84	5.47
	ResNet101	94.98	99.89	4.91
	InceptionV2	94.09	99.75	5.66
	EfficientNet-b0	93.12	99.75	6.63

LSTM-RDN, and the recognition accuracies on Cifar10 and Fashion-MNIST has been improved from 5% to 14% (see the rightmost column of Table 2). On Fashion-MNIST, the recognition accuracies of these models exceed 99%. Especially on Cifar100, the recognition accuracies of the transfer-learned models are only 60%, and LSTM-RDN can increase backbones' recognition accuracies to more than 95%, and the improved percentage point is more than 30%. To illustrate our

TABLE II
RECENT RESULTS ON CIFAR10

DCNN	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	99.93
Our LSTM-RDN	VGG16	99.94
Our LSTM-RDN	ResNet18	99.91
Our LSTM-RDN	ResNet101	99.82
Our LSTM-RDN	InceptionV2	99.93
Our LSTM-RDN	EfficientNet-b0	99.39
EffNet-L2[28]	ResNet	99.70
ViT-H[17]	ResNet152	99.50
EfficientNet-B7[29]	-	98.90
BiT-L[30]	ResNet	99.30
LaNet[31]	ResNet	99.03
MUXNets[32]	-	98.0
DAT[33]	AmoebaNet-B	98.80
NasMobile[34]	-	91.88
ResMLP-24[16]	MPL	98.70

method LSTM-RDN is effective and advanced, we compare it with the advanced methods that have been published. These comparisons are listed in TableII-TableIV. Currently, the best records on the three datasets are: 99.70%(Cifar10), 96.08%(Cifar100), 96.91%(Fashion-MNIST), and our highest results are 99.94% (Cifar10), 98.21%(Cifar100) and 99.98% (Fashion-MNIST). It is obvious that the recognition performance of LSTM-RDN based on pre-trained models is significantly higher than the best record.

B. Fine-Grained Image Classification

Fine-grained image recognition is a more refined distinction between target image recognition (classification), such as not only identifying that this is a dog, but also telling whether

TABLE III
RECENT RESULTS ON CIFAR100

DCNN	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	96.57
Our LSTM-RDN	VGG16	98.21
Our LSTM-RDN	ResNet18	95.65
Our LSTM-RDN	ResNet101	97.33
Our LSTM-RDN	InceptionV2	98.17
Our LSTM-RDN	EfficientNet-b0	95.29
EffNet-L2[28]	ResNet	96.08
TResNet[35]	ResNet	91.15
ViT-H[17]	ResNet152	94.55
EfficientNet[29]	-	91.70
BiT-L[30]	ResNet	93.51
MUXNets[32]	-	86.11
DARTS[36]	ResNet110	84.10
ResMLP-24[16]	MPL	89.5

TABLE IV
RECENT RESULTS ON FASHION-MNIST

DCNN	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	99.98
Our LSTM-RDN	VGG16	99.95
Our LSTM-RDN	ResNet18	99.84
Our LSTM-RDN	ResNet101	99.89
Our LSTM-RDN	InceptionV2	99.75
Our LSTM-RDN	EfficientNet-b0	99.75
Few-shot Learning[37]	ResNet	91.42
DARTS[36]	ResNet110	96.91
E2E-3M[38]	-	95.92
SpinalNet[39]	VGG-5	94.68
Shake-Shake[28]	-	96.41

it is a Labrador or a Husky. Fine-grained image recognition has been proposed in the past few years, but it has not been solved, on the contrary, it is far from our expectations.

CUB200-2011 (CUB200) dataset has 11788 bird images, including 200 bird sub-categories. The training dataset has 5994 images, and the test set has 5794 images. Stanford Cars (Cars) dataset is an image dataset containing 196 types of cars. It is mainly used for image classification. It has a total of 16185 images, of which 8144 and 8041 images are training images and test images. FGVC-Aircraft-2013b(Aircraft) is a classic benchmark in fine-grained image classification. The dataset contains 10200 images of aircraft, with 100 images for each of 102 different aircraft model variants. The dataset is divided into training, validation and test subsets.

On Fine-grained datasets, the recognition rates of LSTM-RDN with different DCNN models are listed in Table V. It can be seen that the classification ability of the pre-trained model on these three fine-grained images is relatively poor, and the recognition accuracies of these models are roughly between 50% and 70%. However, after feature selection with LSTM-RDN, their recognition accuracy soared to more than 90%, and the performance increased by 30 to 50%. This shows that LSTM-RDN is very effective.

On CUB200-2011, the current best recognition accuracy is 90%, and our proposed method based on ResNet models(ResNet18 and ResNet101) all exceed 98%; while the lowest recognition accuracy by using LSTM-RDN (AlexNet) reaches 92.30% (AlexNet), which is 2 percentage points higher than the best method. On the Stanford Cars and FGVC-

TABLE V
RECOGNITION RESULTS ON FINE-GRAINED DATASETS

Dataset	Model	Transfer	LSTM-RDN	Imp.
CUB200-2011	AlexNet	47.98	92.30	44.32
	VGG16	59.21	96.27	37.06
	ResNet18	62.72	98.11	35.39
	ResNet101	68.63	98.15	29.52
	InceptionV2	68.49	97.77	29.28
	EfficientNet-b0	66.70	97.27	30.57
Stanford-Cars	AlexNet	35.02	92.15	57.13
	VGG16	72.34	98.50	26.16
	ResNet18	63.98	98.81	34.83
	ResNet101	64.66	97.97	33.31
	InceptionV2	63.02	98.66	35.64
	EfficientNet-b0	65.5	97.57	32.07
FGVC-Aircraft	AlexNet	46.08	87.99	41.91
	VGG16	60.42	92.97	32.55
	ResNet18	59.34	95.64	36.30
	ResNet101	52.77	94.92	42.15
	InceptionV2	57.17	94.80	37.68
	EfficientNet-b0	51.54	94.89	43.35

TABLE VI
RECENT RESULTS ON CUB200-2011

Model	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	92.30
Our LSTM-RDN	VGG16	96.27
Our LSTM-RDN	ResNet18	98.11
Our LSTM-RDN	ResNet101	98.15
Our LSTM-RDN	InceptionV2	97.77
Our LSTM-RDN	EfficientNet-b0	97.27
API-Net[40]	DenseNet161	90.00
MC-Loss[41]	B-CNN	86.40
DF-GMM[42]	-	88.80
PC-Softmax[38]	-	89.73
MGE-CNN[43]	Resnet-101	89.40
DCL[44]	ResNet50	87.80
MMAL-Net[45]	ResNet-50	89.60

Aircraft datasets, the highest recognition accuracy accuracies are 96.2% DAT(AmoebaNet-B) and 94.70% MMAL-Net(ResNet-50), respectively; while, the best recognition accuracies of LSTM-RDN (ResNet18) are 98.81% and 95.64%, respectively. The specific results are shown in Table VII and Table VIII.

TABLE VII
RECENT RESULTS ON STANFORD CARS

Model	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	92.15
Our LSTM-RDN	VGG16	98.50
Our LSTM-RDN	ResNet18	98.81
Our LSTM-RDN	ResNet101	97.97
Our LSTM-RDN	InceptionV2	98.66
Our LSTM-RDN	EfficientNet-b0	97.57
API-Net[40]	DenseNet161	95.30
MC-Loss[41]	B-CNN	94.40
SpinalNet[39]	ResNet101	93.35
DAT[33]	AmoebaNet-B	96.20
MGE-CNN[43]	Resnet-50	93.90
DF-GMM[42]	-	94.80
DCL[44]	ResNet50	94.50
MMAL-Net[45]	ResNet-50	95.0
ResMLP-24[16]	MPL	89.50

TABLE VIII
RECENT RESULTS ON FGVC-AIRCRAFT

Model	Backbone model	Accuracy
Our LSTM-RDN	AlexNet	87.99
Our LSTM-RDN	VGG16	92.97
Our LSTM-RDN	ResNet18	95.64
Our LSTM-RDN	ResNet101	94.92
Our LSTM-RDN	InceptionV2	94.80
Our LSTM-RDN	EfficientNet-b0	94.89
API-Net[40]	DenseNet161	90.30
MC-Loss[41]	B-CNN	92.90
DF-GMM[42]	-	93.80
DAT[33]	InceptionV3	94.10
DCL[44]	ResNet50	93.00
MMAL-Net[45]	ResNet-50	94.70

True class	TP (True Positive)	FN (False Negative)
	FP (False Positive)	TN (True Negative)
Predicted class		

Fig. 9. The define of confusion matrix for binary classification.

C. COVID-19 prediction

We train a binary classification model by using transfer learning for predicting whether a CT image is COVID or non-COVID. The evaluation is executed on COVID_CT dataset[46] which is an internet-based dataset that contains CT images from patients. This dataset contains 349 CT images positive for COVID-19 belonging to 216 patients and 397 CT images that are negative for COVID-19. It is divided into three sets: train, validation and test sets(see TableIX). The train and validation sets are used for model learning. The test set is used to measure the recognition accuracy of the models. In order to obtain good performance, for the one-dimensional features generated by different backbone models, we use different training parameters including loss functions, learning rates and dropout probabilities to fine-tune LSTM-RDN model.

TABLE IX
DATA SPLIT IN COVID_CT DATASET[46]

	Totale	Train set	validation set	test set
COVID	349	191	60	98
Non-COVID	397	234	58	105

In the experiments, accuracy, precision, F1-score, and recall evaluation measures were used. The calculation of these measures is related to the confusion matrix. The confusion matrix

Fig. 10. Samples selected from COVID_CT dataset. The first two rows show CT images of COVID; The last two rows are CT images Non-COVID

for binary classification is shown in Fig.9.

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN}, \\
 Precision &= \frac{TP}{TP + FP}, \\
 Recall &= \frac{TP}{TP + FN}, \\
 F1 - score &= 2 \times \frac{Precision \times Recall}{Precision + Recall}.
 \end{aligned} \tag{7}$$

TABLE X
THE ACCURACY ON COVID_CT(%)

Model	Backbone model	LSTM-RDN
VGG16	73.39	95.46
ResNet18	72.41	95.51
ResNet101	75.37	99.01
HFSM[47]		93.0
DenseNet-169 (mask) [46]		87.1
TL+CSS[46]		89.1
Multi-Task deep model[48]		86.0

TableX shows the recognition accuracy of VGG16 is 73.39%, while the performance is improved to 95.46% by feature selective transformation with LSTM-RDN. Resnet18 achieved an accuracy of 93.1%, which is very close to VGG16's recognition performance. Resnet101, by contrast, performs quite well on this dataset, with a recognition rate of 99.01%. While, the accuracy of DenseNet169 with mask which has more parameters and more layers is 87.1%, and TL+CSS obtains 89.1% accuracy. The advanced method HFSM [47], which also uses feature selection to improve performance, achieves 93.0% accuracy. It can be seen that our method achieves the best performance on the three backbone models (VGG16, ResNet18 and ResNet101).

In other measures, DenseNet169 (mask) and TL+CSS achieved 88.1% and 89.6% on F1-score, respectively, and our method significantly surpassed these two methods, reaching a maximum of 98.97%, and increasing about 10 percent points. Although HFSM performs well in accuracy, it is relatively

Fig. 11. Confusion matrix. Number 1 represents the COVID class, number 2 represents the Non-COVID class.

TABLE XI
THE OTHER EVALUATION MEASURES ON COVID_CT(%)

Model	Precision	Recall	F1-score
DenseNet169(mask)[46]	-	-	88.1
TL+CSSL[46]	-	-	89.6
HFSM[47]	72	71	-
LSTM-RDN (VGG16)	96.81	92.86	94.79
LSTM-RDN(ResNet18)	94.17	98.80	96.52
LSTM-RDN(ResNet101)	100	97.96	98.97

poor in precision and recall measures, only getting 72% and 71%, respectively. LSTM-RDN(ResNet101) shows excellent performance, precision is 100%, recall also reaches 97.96%. LSTM-RDN(ResNet18) and LSTM-RDN (VGG16) are also significantly better than the other three methods.

We use the confusion matrix to visually show the performance of LSTM-RDN(ResNet101), LSTM-RDN(ResNet18) and LSTM-RDN (VGG16)(see Fig.11). Only two COVID images were misjudged as Non-COVID images by LSTM-RDN (ResNet101). LSTM-RDN (ResNet18) mistakenly judged one COVID as Non-COVID, and also mistakenly judged 6 Non-COVID images as COVID images. LSTM-RDN (VGG16) has more errors for predicting COVID, reaching 7 images, but it has fewer errors for predicting Non-COVID, only 2 images.

Due to the randomness of LST-RDN, the identification results of the network of each trained model may have slight changes. We have conducted 10 times of training and testing, and the testing accuracy obtained has slight fluctuations (see Fig.12). The fluctuation range is about 1.5%. In addition to the use of Dropout, the fluctuations of recognition accuracy maybe caused by the noise in CT images and the small number of samples.

V. DISCUSS AND ABLATION

From Table II and Table VI, it can be seen that for ordinary object image recognition, two shallow networks AlexNet and VGG16 with LSTM-RDN have obtained better results; super-deep DCNN, such as ResNet101, InceptionV2 and EfficientNet-b0, does not have obvious advantages over shallow networks. In fine-grained image recognition, ResNets, InceptionV2 have achieved excellent results, among which the lightweight ResNet18 has outstanding performance. Therefore, we have the following conclusions:

Fig. 12. The prediction accuracies of 10 times training.

- For coarse-grained image classification, LSTM-RDN combining with a simple or shallow network model such as AlexNet can achieve satisfactory results.
- For fine-grained image classification, LSTM-RDN combining with the ResNet models that take into account multi-scale convolution are more effective. To COVID prediction, ResNet101 has best performance with more than 99% recognition accuracy.
- With the LSTM-RDN's feature selective transformation, the backbone network model does not need to have so many layers and branches. For example, ResNet18 with only 71 layers has equal or better performance than ResNet101 with 347 layers and EfficientNet-b0 with 290 layers, as well as multi-branch 824-layer InceptionV2 on fine-grained image classification.
- For COVID-19 prediction, the modest deep model ResNet101 with LSTM-RDN has obvious advantage than two shallow models including ResNwt18 and VGG16; Super-deep model such as InceptionV2 and EfficientNet yield average performance. Lung CT image is a type of special fine-grained image. CT image contains large amount of noise and the number of samples in CT image dataset is far more less than ordinary image datasets. This is the main cause of performance degradation of super-deep model.

LSTM-RDN contains two important parts: a LSTM layer, as well as several ReLU and Dropout layers; both parts play an important role. LSTM can increase the recognition accuracy of AlexNet/ResNet18 to more than 80%/90% on the FGVC-

Fig. 13. The recognition performance of LSTM combining different numbers of Dropout layers on FGVC-Aircraft dataset. LSTM-D0 is LSTM combined with the first dropout, LSTM-RD1 denotes the LSTM layer combined with ReLU and the second Dropout, and so on.

Fig. 14. The recognition performance of LSTM combining different numbers of Dropout layers on Stanford Cars dataset.

Aircraft and Stanford Cars datasets, thanks to the unique memory structure of LSTM. Next, we need to know how effective the Dropout layer is in LSTM-RDN. Fig.13 and Fig.14 show the recognition performance of LSTM combined with different numbers of Dropout layers on two datasets (AlexNet and ResNet18). When the number of feature dimensions is large, more Dropout layers are needed. It also requires more training time. In the experiment, we conclude that the performance of the two Dropout layers is almost optimal based on considering the recognition accuracy and training time.

It is natural to think of using LSTM-RDN to combine the features of multiple models for further improving performance. We make two combinations including ResNet18+AlexNet and ResNet18+ResNet101 to evaluate the performance on Cifar100 (see Table XII). The experiment on Cifar100 shows that compared with a single DCNN features, the feature combination of two DCNNs only slightly improves the recognition accuracy. The advantage of combination is we can obtain more robust performance compared to single model.

TABLE XII
PERFORMANCE COMPARISON (RECOGNITION ACCURACY) OF
LSTM-RDN COMBINING MULTI-MODELS AND A SINGLE MODEL ON
CIFAR100

ResNet18	AlexNet	ResNet18+AlexNet
95.65	96.57	96.59
ResNet18	ResNet101	ResNet18+ResNet101
95.65	97.33	97.44

VI. CONCLUSIONS

At present, a large number of image classification models have appeared, and have achieved unprecedented performance improvements. However, they still have not broken the bottleneck of classification tasks. The parameter scale of hundreds of millions is the main factor restricting performance improvement. With mainstream optimal techniques such as gradient descent algorithm, it is difficult to achieve optimal model parameters even using millions of samples and thousands of training epochs. Multi-stage processing strategy can reduce the workload of optimizing parameters exponentially, which is an effective strategy to overcome the problem of huge parameters. We have improved the classification accuracy to a new level on several image databases through a two-stage feature processing strategy (feature extraction and feature selective transformation). The performance of the backbone model may not be the main factor that determines the final classification performance. Multi-stage feature transformation can be used to make up for the lack of performance. Experiments show that the accuracy of the backbone model network can be increased by more than 30%.

Based on feature transformation of multi-stage strategy, we proposed LSTM-RDN feature selective network, which can improve the performance of the pre-trained network models such as AlexNet and ResNet18 to a very satisfactory level. We also come to the conclusion the performance of shallow network can be even better than that of super-deep network by feature selective transformation. The LSTM-RDN's performance may have room for improvement, so, in future work, we will continue to conduct more research on the feature selection of DCNN one-dimensional feature, such as exploring the influence of the feature dimension, as well as the number of LSTM layers and the number of hidden units of LSTM-RDN.

REFERENCES

- [1] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [2] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [3] D. G. Lowe, "Lowe, d.g.: Distinctive image features from scale-invariant key-points," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [4] H. Du, L. Ma, G. Li, and S. Wang, "Low-rank graph preserving discriminative dictionary learning for image recognition," *Knowledge Based Systems*, vol. 187, p. 104823, 2020.
- [5] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua, "Learning separable filters," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2754–2761.
- [6] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive o-cnn: a patch-based deep representation of 3d shapes," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–11, 2019.
- [7] Krizhevsky, Alex, Sutskever, Ilya, Hinton, and E. Geoffrey, "Imagenet classification with deep convolutional neural networks," *Communications of the Acm*, pp. 1–9, 2017.
- [8] S.Karen and Z.Andrew, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556v6*, 2014.
- [9] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *CVPR*, 2018, pp. 8698–8710.

- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [11] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016.
- [12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *JMLR.org*, 2015.
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE*, pp. 2818–2826, 2016.
- [14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.
- [15] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, and M. Shah, "Transformers in vision: A survey," *arXiv:2101.01169v1*.
- [16] H. Touvron, P. Bojanowski, M. Caron, M. Cord, and H. Jégou, "Resmlp: Feedforward networks for image classification with data-efficient training," in *CVPR*, 2021.
- [17] a. K. D. W. X. Z. T. U. M. D. M. M. G. H. S. G. J. U. N. H. A. Dosovitskiy, Lucas Beyer, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv:2010.11929v1*, pp. 1–21, 2020.
- [18] C. Li, Y. Huang, and Y. Xue, "Dependence structure of gabor wavelets based on copula for face recognition," *Expert Systems with Applications*, vol. 137, pp. 453–470, 2019.
- [19] S. Yao, S. Pan, T. Wang, C. Zheng, W. Shen, and Y. Chong, "A new pedestrian detection method based on combined hog and lss features," *Neurocomputing*, vol. 151, no. mar.3pt.3, pp. 1006–1014, 2015.
- [20] J. Li, J. H. Cheng, J. Y. Shi, and F. Huang, *Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement*. Advances in Computer Science and Information Engineering, 2012.
- [21] W. Jiang, Y. Yi, J. Mao, Z. Huang, and X. Wei, "Cnn-rnn: A unified framework for multi-label image classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [22] J. Ma and Y. Yuan, "Dimension reduction of image deep feature using pca," *Journal of Visual Communication and Image Representation*, vol. 63, p. 102578, 2019.
- [23] R. Varatharajan, G. Manogaran, and M. K. Priyan, "A big data classification approach using lda with an enhanced svm method for ecg signals in cloud computing," *Multimedia Tools and Applications*, 2017.
- [24] L. Sun, Z. Mo, F. Yan, L. Xia, F. Shan, Z. Ding, B. Song, W. Gao, W. Shao, F. Shi *et al.*, "Adaptive feature selection guided deep forest for covid-19 classification with chest ct," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2798–2805, 2020.
- [25] A. K. Bhunia, A. Konwer, A. K. Bhunia, A. Bhowmick, P. P. Roy, and U. Pal, "Script identification in natural scene image and video frames using an attention based convolutional-lstm network," *Pattern Recognition*, vol. 85, pp. 172–184, 2019.
- [26] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional cnn-lstm model for dimensional sentiment analysis," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 581–591, 2020.
- [27] J. Choe, S. Lee, and H. Shim, "Attention-based dropout layer for weakly supervised single object localization and semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2020.
- [28] H. M. B. N. Pierre Foret, Ariel Kleiner, "Sharpness-aware minimization for efficiently improving generalization," *arXiv:2010.01412*, pp. 1–18, 2020.
- [29] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv:1905.11946v5*, 2020.
- [30] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," *European Conference on Computer Vision*, pp. 491–507, 2019.
- [31] L. Wang, S. Xie, T. Li, R. Fonseca, and Y. Tian, "Sample-efficient neural architecture search by learning action space," *arXiv:1906.06832v1*, 2019.
- [32] Z. Lu, K. Deb, and V. N. Boddeti, "Muxconv: Information multiplexing in convolutional neural networks," in *CVPR*, 2020, pp. 12 041–12 050.
- [33] V. V. S. K. Q. V. L. R. P. Jiquan Ngiam, Daiyi Peng, "Domain adaptive transfer learning with specialist models," *arXiv:1811.07056v1*, 2018.
- [34] Y. Xiong, H. Liu, S. Gupta, B. Akin, G. Bender, P. J. Kindermans, M. Tan, V. Singh, and B. Chen, "Mobiledets: Searching for object detection architectures for mobile accelerators," *arXiv*, pp. 1–18, 2020.
- [35] T. Ridnik, H. Lawen, A. Noy, I. Friedman, E. B. Baruch, and G. Sharir, "Tresnet: High performance gpu-dedicated architecture," *arXiv:2003.13630v3*, 2020.
- [36] C. K. M. S. Tanveer, M. U. K. Khan, "Fine-tuning darts for image classification," *arXiv:2006.09042v1*, pp. 1–9, 2020.
- [37] X. Sun, H. Xv, J. Dong, H. Zhou, C. Chen, and Q. Li, "Few-shot learning for domain-specific fine-grained image classification," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3588–3598, 2021.
- [38] B. R. N.H. Phong, "Rethinking recurrent neural networks and other improvements for image classification," *arXiv:2007.15161v1*, pp. 1–18, 2020.
- [39] S. J. A. K. A. A. S. N. D. S. HMD Kabir, M. Abdar, "Spinalnet: Deep neural network with gradual input," *arXiv:2007.03347v*, pp. 1–10, 2020.
- [40] Y. Q. Peiqin Zhuang, Yali Wang, "Learning attentive pairwise interaction for fine-grained classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3130–13 137, 2020.
- [41] D. Chang, Y. Ding, J. Xie, A. K. Bhunia, X. Li, Z. Ma, M. Wu, J. Guo, and Y. Z. Song, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 29, pp. 4683–4695, 2020.
- [42] Z. Wang, S. Wang, S. Yang, H. Li, J. Li, and Z. Li, "Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning," in *CVPR*, 2020, pp. 9746–9755.
- [43] W. L. D. T. L. Zhang, S. Huang, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *CVPR*, 2019, pp. 8332–8340.
- [44] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *CVPR*, 2019.
- [45] G. Z. Y. L. Fan Zhang, Meng Li, "Multi-branch and multi-scale attention learning for fine-grained visual categorization," *arXiv:2003.09150v3*, 2020.
- [46] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "Covid-ct-dataset: A ct scan dataset about covid-19," *arXiv:2003.13865*, 2020.
- [47] A. Wms, B. Ahr, B. Ais, and C. Ae, "A new covid-19 patients detection strategy (cpds) based on hybrid feature selection and enhanced knn classifier," *Knowledge-Based Systems*, vol. 205, pp. 1–18, 2020.
- [48] A. Amyar, R. Modzelewski, and R. Su, "Multi-task deep learning based ct imaging analysis for covid-19: Classification and segmentation," vol. 126, 2020.