Method for the automatic recognition of cropland headland images based on deep learning

Yujie Qiao[1,2], Hui Liu[1*], Zhijun Meng[2*], Jingping Chen[2], Luyao Ma[1]
(1. *Information Engineering College, Capital Normal University, Beijing 100048, China*;
2. *National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China*)

Abstract: For self-driving agricultural vehicles, the sensing of the headland environment based on image recognition is an important technological aspect. Cropland headland environments are complex and diverse. Traditional image feature extraction methods have many limitations. This study proposed a method of automatic cropland headland image recognition based on deep learning. Based on the characteristics of cropland headland environments and practical application needs, a dataset was constructed containing six categories of annotated cropland headland images and an augmented headland image training set was used to train the compact network MobileNetV2. Under the same experimental conditions, the model prediction accuracy for the first ranked category in all the results (Top-1 accuracy) of the MobileNetV2 network on the validation set was 98.5%. Compared with classic ResNetV2-50, Inception-V3, and backend-compressed Inception-V3, MobileNetV2 has a high accuracy, high recognition speed, and a small memory footprint. To further test the performance of the model, 250 images were used for each of the six categories of headland images as the test set for the experiments. The average of the harmonic mean of precision and recall (F1-score) of the MobileNetV2 network for all the categories of headland images reached 97%. The MobileNetV2 network exhibits good robustness and stability. The results of this study indicate that onboard computers on self-driving agricultural vehicles are able to employ the MobileNetV2 network for headland image recognition to meet the application requirements of headland environment sensing.

Keywords: cropland image, deep learning, image recognition, model compression, MobileNetV2 network
DOI: 10.25165/j.ijabe.20231602.6195

Citation: Qiao Y J, Liu H, Meng Z J, Chen J P, Ma L Y. Method for the automatic recognition of cropland headland images based on deep learning. Int J Agric & Biol Eng, 2023; 16(2): 216–224.

1 Introduction

Agricultural vehicle automatic navigation technology can lead to significant reductions in overlap and skip in field farming operations and labor intensity. It is widely welcomed by farmers and has become the most widely used technology for promoting precision agriculture[1]. After nearly 20 years of development, the automatic navigation technology of agricultural vehicles has become industrialized. The goal of the next stage of development will be the ability to automate the entire agricultural vehicle operation. Although there are differences in terms of plot size and vehicle types, an operating tractor unit must perform headland turning. Therefore, the sensing of the headland environment is the key point in the research of agricultural vehicle self-driving technology but it has been difficult to determine thus far.

A headland is commonly considered the boundary of a cropland plot. As early as the 1990s, the concept of digital cropland management was proposed in the field of precision agriculture research and was suggested to be used to map cropland plots through surveying and mapping or remote sensing imagery[2,3]. In digital maps of cropland, the boundaries of the plots are abstracted as a line attribute. However, a headland for agricultural vehicle turning is an area, not a line. Furthermore, the accuracy of the current digital maps of cropland cannot meet the requirement of self-driving of agricultural vehicles. Therefore, it is necessary to explore more suitable headland environment sensing technology.

As an important research field of artificial intelligence, image recognition has been widely employed. A fundamental step of image recognition is image feature extraction. Traditional image feature extraction methods use manual extractors, such as Speeded Up Robust Features (SURF)[4], Scale Invariant Feature Transform (SIFT)[5], and Histogram of Oriented Gradient (HOG)[6], to extract local features such as color, texture, and shape through expert knowledge and complex parameter adjustments. However, manual feature extraction methods do not fully represent image semantics, and extractors are generally application-specific, and are poor in generalization and robustness.

Over the past decade, “deep learning” has been an important breakthrough in the field of artificial intelligence. It has achieved great success in many areas, including speech recognition, natural language processing, computer image, and video analysis. The image recognition methods based on deep learning can automatically learn image features from big data and extract multiple layers of information from low-level features to abstract semantic concepts. For a deep learning algorithm, the more features are used to train the deep learning network, the better the robustness and generalization ability of the algorithm. In 1998, LeNet5[7] became the first large-scale commercial deep-learning network model. Subsequently, other network models such as AlexNet[8], GoogLeNet[9], VGG-Nets[10], and ResNet[11] have been developed,

Received date: 2020-09-30 **Accepted date:** 2022-05-22
Biographies: Yujie Qiao, Master, research interest: computer vision, Email: qiaoyujie_123@126.com; **Jingping Chen**, Master, Senior Engineer, research interest: multi-source spatial data analysis in precision agriculture. Email: chenjp@nercita.org.cn; **Luyao Ma**, Master, research interest: computer vision, Email: maluyao_mall@163.com.
***Corresponding author: Hui Liu**, PhD, Professor, research interest: ICT application in precision agriculture. No.56, North Road of Western 3rd-Ring, Beijing 100048, China. Tel: +86-10-68901370, Email: liuhui_mail@163.com; **Meng Zhijun**, PhD, Professor, research interest: intelligent agricultural equipment. Room A-517, Beijing Nongke Mansion, No.11 Shuguang Huayuan Middle Road, Beijing 100097, China. Tel: +86-10-51503785, Email: mengzj@nercita.org.cn.

intensifying the research and application of convolutional neural network automatic image feature extraction methods in different fields. There have been studies and applications of deep learning technology in the field of agriculture. The average recognition accuracy of a wheat kernel image detection system based on AlexNet is 96.67%[12], and the average recall of recognition of agricultural vehicle and equipment images based on convolutional neural network is 98.8%[13]. Based on transfer learning, good recognition results have been achieved in image recognition of diseases and pests in cotton leaves[14] and diseases in maize[15]. These studies can serve as references for crop disease diagnosis. In recent years, deep learning techniques have been applied to farm area recognition[16] and boundary line detection[17] in the research of self-driving agricultural vehicles.

In actual cropland environments, the types of headlands are complex and diverse. The headlands can be the boundaries of adjacent fields where crops are grown, or they can be non-field areas, such as ridges, gravel roads, ditches, or bare soil. In addition, for a self-driving agricultural vehicle to determine whether it is near a headland area, it needs to recognize the cropland environment in which the vehicle is operating. Cropland environments are also diverse due to the growth of different crops and different crop growth periods. Manual feature extraction methods that use traditional image recognition have many limitations and are clearly unable to adapt well to the complex and varying characteristics of cropland and headland environments in nature[18]. Therefore, this study proposed an automatic recognition method based on deep learning to study cropland headland images, for the vehicles to better adapt to the complex headland environment, and further promote the development and practical application of self-driving agricultural vehicle technology.

Based on the characteristics of a cropland headland environment and practical application needs, an annotated cropland headland image dataset was constructed and an augmented headland image training set was used to train the compact network MobileNetV2 in this study.

2 Materials and methods

2.1 Classification of headlands

A headland, or a boundary of a field, can be defined as a transition zone between a cropland field and other areas, and it can appear as a variety of scene types of natural or man-made structures. According to the Current Land Use Classification (GB/T 21010-2017)[19], croplands can be roughly divided into three categories: dry fields, irrigated fields, and paddy fields. This study used agricultural vehicle dry field operations for the research background and classifies the dry field operation environment scenes into two categories: field and headland, as shown in Figure 1.

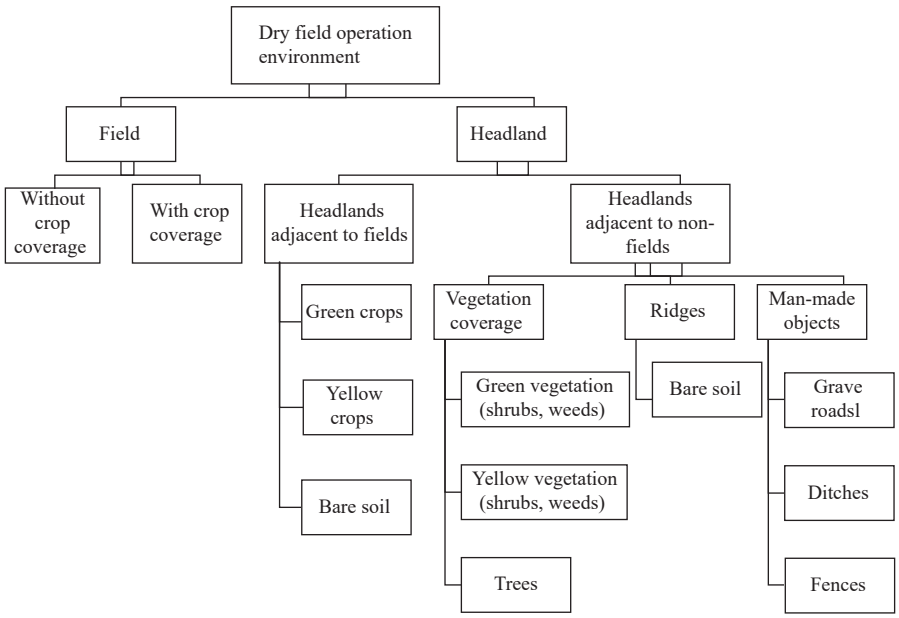


Figure 1 Classification of scene elements of cropland headlands

The cropland field environment was categorized as either fields with crop covering or fields without corps covering. According to the neighboring environment, the headland environment, as a transition zone of the fields, was categorized as either headlands adjacent to fields or headlands adjacent to non-fields. The headlands adjacent to the fields category, according to the difference in field covering, were further categorized into three categories: covering by immature green crops, covering by mature yellow crops, and bare soil without crop covering. The headlands adjacent to non-fields are complex. According to the landscape element structures, they are divided into vegetation covering (weeds, shrubs, trees), ridges (bare soil), and man-made objects.

2.2 Creation of cropland headland image dataset

The classification and analysis of cropland headlands showed that although some headland environments have different field covering types, such as green crop covering adjacent to fields and green vegetation covering adjacent to non-fields, these environments can be classified in terms of image recognition. According to the characteristics of the images such as color and texture and using a visualization interpretation method[20] this study assigned a total of six categories of the dry field operating environments in the automatic image recognition test. As shown in Figure 2, the six categories of images are the following: 1) images of fields without crops; 2) images of fields with crops; 3) images of headlands with green vegetation, including images of headlands adjacent to green crops, and images of headlands with bushes and weeds green vegetations; 4) images of headlands with yellow vegetation, including images of headlands with mature crops and

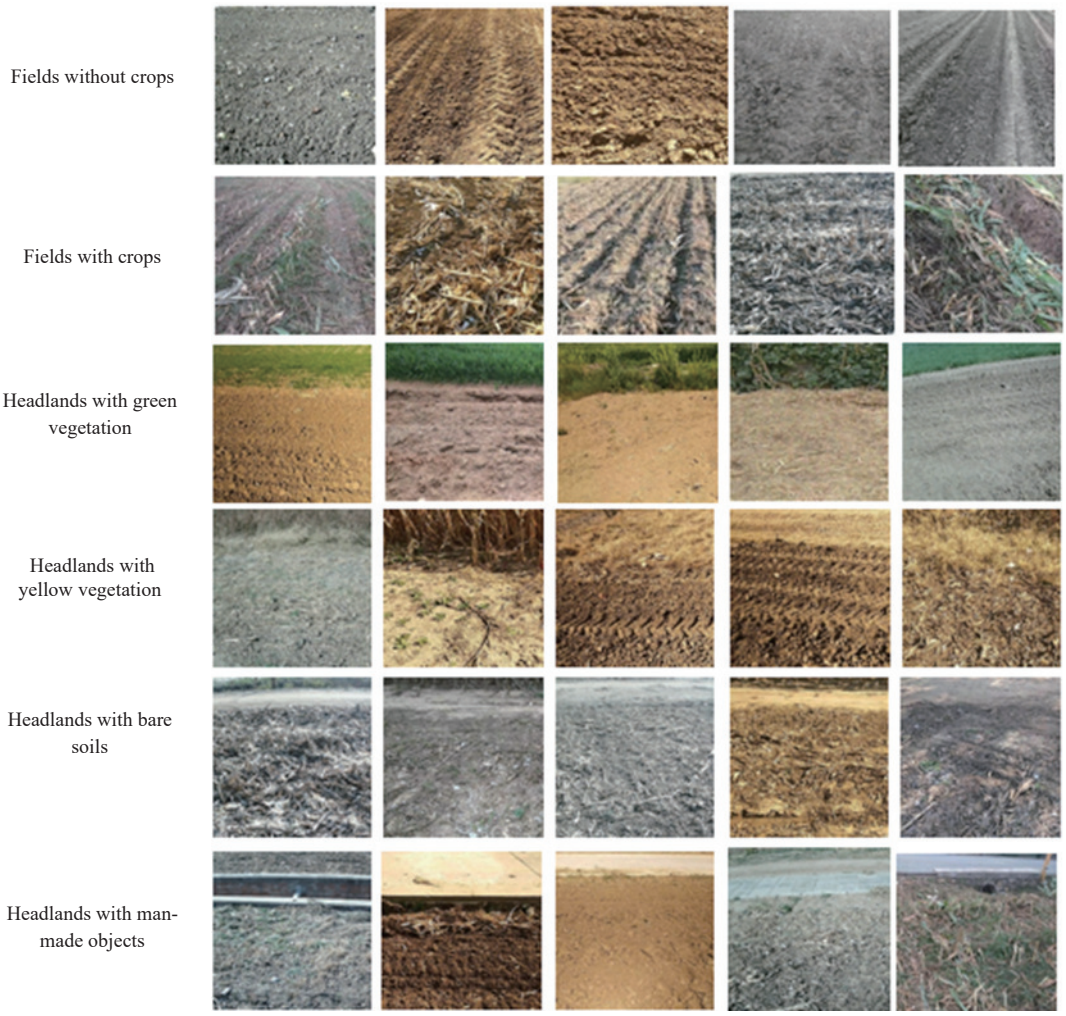


Figure 2 Illustrations of six classes of cropland headland images

images of headlands with yellow weeds and bushes; 5) images of headlands with bare soils. Including images of headlands adjacent to fields with bare soils and images of headlands adjacent to ridges; 6) images of headlands with man-made objects, such as gravel roads, ditches, and fences.

A dataset of 9000 cropland headland images was created in this study, and it was divided into a training set, a validation set, and a test set by the "hold-out" method[21]with the image number ratio of 4:1:1. The training set of 6000 images was used to train the model, with 1000 images in each category. The validation set of 1500 images was used to validate the accuracy of the model, with 250 images in each category. The test set of 1500 images, 250 images in each category, was used to test the performance in practical applications. The composition of the cropland headland image dataset is listed in Table 1.

Table 1 Composition of the cropland headland image dataset

Image class	Training set	Augmented training set	Validation set	Test set
Fields without crops	1000	5000	250	250
Fields with crops	1000	5000	250	250
Headlands with green vegetation	1000	5000	250	250
Headlands with yellow vegetation	1000	5000	250	250
Headlands with bare soils	1000	5000	250	250
Headlands with man-made objects	1000	5000	250	250
Total	6000	30 000	1500	1500

2.3 Image preprocessing

In this study, the acquired images of cropland headlands were divided, scaled, and augmented.

1) Image division and scaling

The acquired cropland headland images are generally larger than the input image size of a convolutional neural network; therefore, the original images need to be resized to the same size. Because the length and width of the original images are different, if the images are scaled at an aspect ratio of 1:1, the image contents are distorted, which will impact the classification and recognition effectiveness. As shown in Figure 3, the short side of an original image is taken as the side length, cuts it into an image with an aspect ratio of 1:1, and then use the nearest neighbor interpolation method to scale the image to 224×224 pixels or 299×299 pixels to meet the input requirements of different networks in the experiment and to ensure that the image information is not lost. In addition, this method can divide an image into two, which can solve the problem of imbalanced data caused by an insufficient quantity of images for

Figure 3 Diagram schematic of image division

specific categories.

2) Image augmentation

A multi-scaling method was used to augment the training set. Enlarging the original images to different scales makes the field boundary features more distinctive, and thus benefits the extraction of useful features. As an agricultural vehicle approaches a headland area, useful information about the headland first appears from the top of the images. Therefore, when resizing an image, with the upper side along the top of the image, and the left side and the right side being center symmetrical, the original image was resized to the images at 60%, 70%, and 80%, respectively, of its original size. Then, they were zoomed to the original size. Next, the original image was flipped horizontally. This way, the original training set was expanded by a factor of five. In addition, the brightness, contrast, and chroma of the images were randomly transformed to eliminate the disturbances of the light on the images, and they were combined with the multi-scaling method to form an augmented training set.

2.4 Image dataset annotation

This study used Google's TensorFlow deep learning framework to annotate the image dataset of all categories of the cropland headland images. The training set, validation set, and test set respectively contained the six classes of images listed in Table 1. The original image data was stored in a folder for each image class. The original image data was converted into TFRecord data format files using the TensorFlow framework. The beginning part of the file name indicates the types of datasets (training set, validation set, or test set). Finally, the complete dataset of annotated cropland headland images is created and serves as the foundation for subsequent studies on automatic recognition of the cropland headland images.

3 Cropland headland image recognition method

3.1 Overall technical roadmap

The trained model of this study needs to be applied to onboard computers on self-driving agricultural vehicles. The computing performance and memory size of the onboard computers are lower than desktop computers in the same price range. The current popular deep learning models consume a huge amount of computing resources and are not suitable for direct deployment to onboard equipment, and thus need to be compressed. Model compression can be divided into frontend compression and backend compression. Frontend compression is a technique that does not change the original network structure. Its major methods include knowledge distillation[22] and compact model structure designs. The backend compression may change greatly the original network structure. Its major methods include low-rank approximation[23], model pruning[24], and parameter quantization[25]. The overall technical roadmap of cropland headland image recognition is shown in Figure 4. A frontend compression method and a backend compression method were used, respectively, on preprocessed data, to train and validate the headland images. The performance scores were compared using comparative experiments, and the model that performed the best was used for cropland headland image recognition.

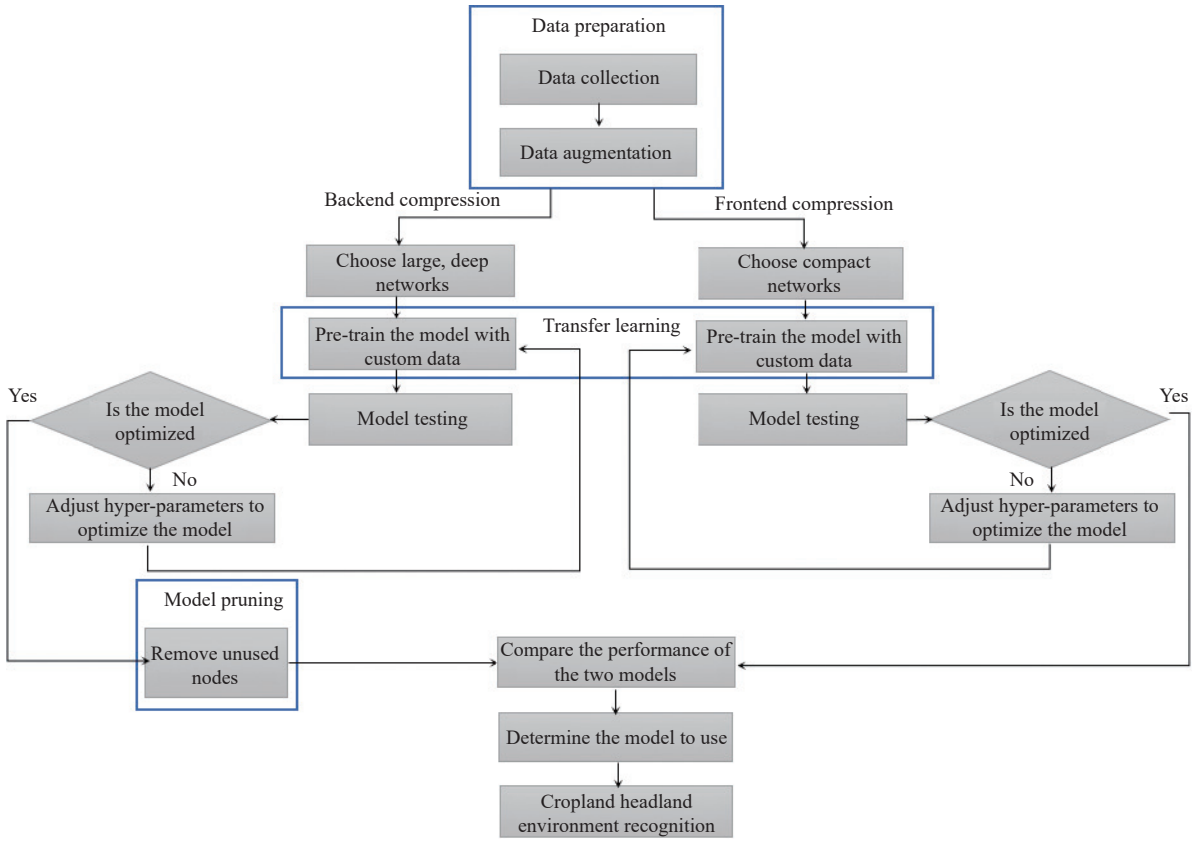


Figure 4 Cropland headland recognition algorithm

3.2 Transfer learning

Transfer learning[26] is a method to transfer knowledge from one domain (i.e., the source domain) to another domain (i.e., the target domain), allowing the target domain to achieve better learning results. In the transfer learning method employed in this study, ImageNet[27] was used as the source dataset, and the network models, MobileNetV2, ResNetV2-50, and Inception-V3, which were used in the experiments of this study described later, were trained to obtain the pre-trained models. The pre-trained models, which already have a certain image recognition ability, were trained

using the headland image dataset as the source data. By fine-tuning the model parameters, the models are able to complete the task of headland recognition faster.

3.3 Frontend compression model

Compact network MobileNetV2[28] was used. This network is a lightweight deep neural network proposed by Google for embedded devices such as mobile phones. It uses a large number of depth separable convolutions[29] to replace standard convolutions and can ensure that the number of model parameters is reduced without losing too much accuracy.

3.3.1 Depth separable convolution

If the size of the input feature map is $F\times F\times N$, where F is the length and width of the input feature map, and N is the number of channels of the input feature map, assuming that the number of channels is equal to the convolution kernels for the convolution operations and that the size of the convolution kernels is $K\times K\times N$, where K represents the length and the width of the convolution kernel and M convolution kernels are used, then the computational cost of standard convolutions is

$$\text{Computational cost} = K\times K\times N\times M\times F\times F \quad (1)$$

Depth separable convolution consists of depthwise convolution and 1×1 convolution (also called pointwise convolution). If the depthwise convolution uses a single filter for each input channel, and then 1×1 convolution is used to control the number of output channels, the total computational cost is

$$\text{Total computational cost}_{1\times1} = K\times K\times M\times F\times F + M\times N\times F\times F \quad (2)$$

Comparing Equations (1) and (2), Equation (3) was got,

$$\frac{K\times K\times M\times F\times F + M\times N\times F\times F}{K\times K\times M\times N\times F\times F} = \frac{1}{N} + \frac{1}{K^2} \quad (3)$$

If the size of the input feature map is 28×28×192 and the size of the depth separable convolution kernel is 3×3, the computational cost is approximately between 1/9 and 1/8 of the computational costs of standard convolutions. MobileNetV2 introduced a hyper-parameter γ, to control the thickness of each layer. γ was set to 1.4 in this study. The total computational cost is

$$\text{Total computational cost}_{3\times3} = K\times K\times \gamma M\times F\times F + \gamma M\times \gamma N\times F\times F \quad (4)$$

3.3.2 MobileNetV2

MobileNetV2 is an improved network model based on MobileNet. It references residual blocks in ResNet and designs inverted residual blocks with linear output to further improve the accuracy of the model. As shown in Figure 5a, the original residual structure first reduces the channel number using a 1×1 convolution, next it undergoes a 3×3 standard convolution, and then it restores the channel number by a 1×1 convolution. The input and output are added by way of a shortcut. Thus, the residual structure is wide on both sides and narrow in the middle. MobileNetV2 improves this design. As shown in Figure 5b, first, the number of channels is increased by the 1×1 convolution, obtaining more features, next the depthwise convolution of the 3×3 spatial convolution is conducted, then the 1×1 convolution is used to reduce the dimensionality. The structure is narrow on both sides and wide in the middle, which is referred to as an inverted residual structure[28]. In addition, to prevent non-linearity from destroying too many features, the final output does not go through ReLU[30], but is directly linear output.

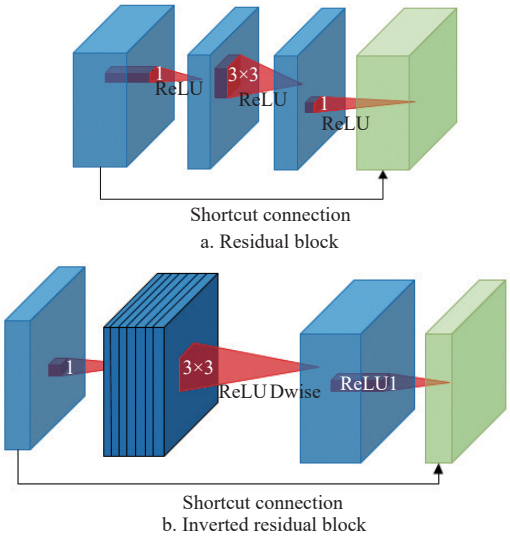


Figure 5 Comparison of residual block and inverted residual block

The MobileNetV2 network structure used in the cropland headland image recognition in this study is shown in Figure 6. The input headland images (224×224) first pass the convolutional module, which comprises standard conventional structures, including convolutional layers, batch normalization, and ReLU. Then, seven inverted residual blocks were used, where the convolutional layers of the stride of 2 do not use the shortcut, and the convolutional layers of the stride of 1 use the shortcut. The output images from the inverted residual blocks then go through the convolutional module. The global average pooling layer was used to replace the fully connected layer to further reduce the number of

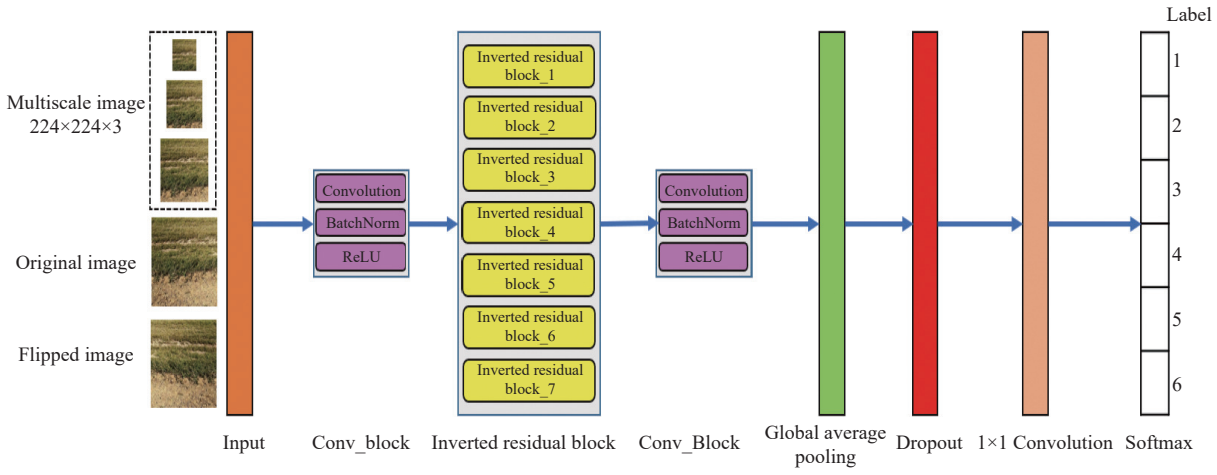


Figure 6 MobileNetV2 network structure diagram

parameters. Then, the images pass the Dropout[31] layer that prevented overfitting, followed by a 1×1 convolutional layer, and finally, they passed the Softmax layer for classification output. Table 2 lists the relevant parameters for the design of the MobileNetV2 network.

Table 2 Parameters for the design of the MobileNetV2 network

No.	Type	Convolution kernel size/step size	Output size
1	Input		224×224×3
2	Conv_Block	3×3/2	112×112×48
3	Inverted residual block1	1 Inverted residual block	112×112×24
4	Inverted residual block2	2 Inverted residual blocks	56×56×32
5	Inverted residual block3	3 Inverted residual blocks	28×28×48
6	Inverted residual block4	4 Inverted residual blocks	14×14×88
7	Inverted residual block5	3 Inverted residual blocks	14×14×136
8	Inverted residual block6	3 Inverted residual blocks	7×7×224
9	Inverted residual block7	1 Inverted residual block	7×7×448
10	Conv_Block	3×3/2	7×7×1792
11	Global average pooling		1×1×1792
12	Dropout		1×1×1792
13	Convolution	1×1/1	1×1×6
14	Softmax		6

3.4 Hyper-parameter design of the network model

3.4.1 Learning rate decay

The learning rate decay technique was used to train the network. Its function is to attenuate the learning rate during the training process. Not only can it increase the parameter update rate in the early stage of training, but also it ensures that the network will not have large fluctuations in the later stage and allows the optimal solution to be reached. The learning rate decay is calculated by Equation (5).

$$\text{decayed\_learning\_rate}=\text{learning\_rate}\times\text{decay\_rate}^{\frac{\text{global\_step}}{\text{decay\_steps}}} \tag{5}$$

where, learning_rate is the initial learning rate; global_step is the total number of training iterations; decay_steps is the decay step; decay_rate is the decay rate of the learning rate.

3.4.2 Optimization algorithm

Momentum[32] was used as the optimization algorithm. The main idea of Momentum is to introduce momentum with accumulated history gradient information to accelerate stochastic gradient descent (SGD). It can not only solve the problem of large oscillation magnitude in updates in SGD optimization algorithms, but also accelerate the convergence to the optimal solution. Assuming that the current iteration step is the t-th step, the rules of the Momentum optimization algorithm are expressed as follows:

$$V_{\mathrm{dW}}=\beta V_{\mathrm{dW}}+(1-\beta)\mathrm{dW} \tag{6}$$

$$V_{\mathrm{db}}=\beta V_{\mathrm{db}}+(1-\beta)\mathrm{db} \tag{7}$$

$$W=W-\alpha V_{\mathrm{dW}} \tag{8}$$

$$b=b-\alpha V_{\mathrm{db}} \tag{9}$$

where, V_{dW} and V_{db} are the gradient momentums that the loss function accumulates in $t-1$ iterations; β is an index of gradient accumulation; dW and db are the gradients obtained when the loss function is back propagated; W and b are the parameters that need to be updated. Equations (8) and (9) are the equations for the updates of the network parameter vector and the bias vector, respectively; α is the learning rate of the network, which is equivalent to the initial learning rate learning_rate. MobileNetV2 converges slowly, so the initial learning rate α is set relatively large.

3.4.3 Regularization

To reduce overfitting during training, L2 regularization was added. L2 regularization was used to add a regularization term to the loss function, and is as follows:

$$C=C_0+\frac{\lambda}{2n}\sum_w W^2 \tag{10}$$

where, C_0 represents the original loss function; $\frac{\lambda}{2n}\sum_w W^2$ is the L2 regularization term; n is the number of samples in the training set; λ is the coefficient of the regularization term.

3.4.4 Exponential moving average

The exponential moving average was used to enhance the generalization ability of the model. The exponential moving average maintains a shadow copy for each variable. The initial value of this shadow copy is the initial value of the corresponding variable. Each time a variable is updated, its shadow copy is updated by Equation (11).

$$\text{shadow\_variable}=\text{decay}\times\text{shadow\_variable}+(1-\text{decay})\times\text{variable} \tag{11}$$

where, shadow_variable is the value before the variable update; variable is the value after the variable update. The decay is calculated as follows:

$$\text{decay}=\min\{\text{init\_decay},(1+\text{num\_update})/(10+\text{num\_update})\} \tag{12}$$

where, init_decay is the set initial decay rate; num_update is the number of model parameter updates.

3.4.5 Model hyper-parameter setting

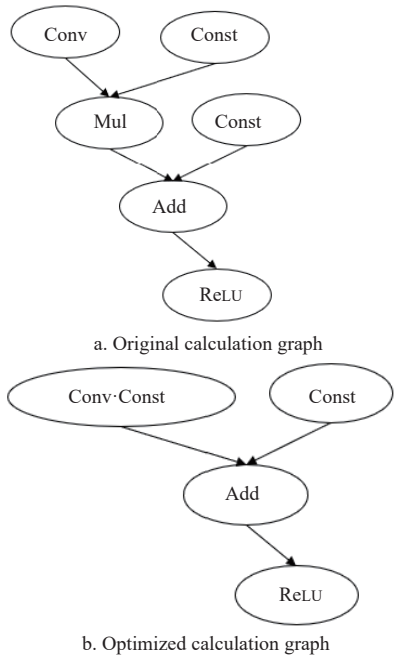
After the MobileNetV2 network was trained and adjusted multiple times, the hyper-parameter values used are listed in Table 3.

Table 3 Hyper-parameter settings for training the MobilenetV2 network

Parameter	Value
Initial learning rate (learning_rate)	0.045
Total number of training iterations (global_step)	100 000
Decay step (decay_steps)	2343
Decay rate of the learning rate (decay_rate)	0.96
Number of training samples in each batch (batch_size)	32
Index of gradient accumulation (β)	0.9
Coefficient of the regularization term (λ)	0.000 04
Initial decay rate (init_decay)	0.999 9

3.5 Backend compression

The backend compression method was adopted to prune the trained models of the large deep networks to reduce the model complexity. The Inception-V3 large deep network was selected for training and model optimization. On this basis, the backend compression processing was performed with the strategies as follows: 1) Remove those unused nodes between the input and the output; 2) Search for the expressions that are always constants in the model and replace them with the constants; 3) Search for all the nodes that perform the multiplication operations immediately after the convolution operation, and perform the multiplication operation in advance, thus reducing the number of nodes of the original computational graph and in turn reducing the computational cost for the entire model. The strategy is shown in Figure 7.



a. Original calculation graph

b. Optimized calculation graph

Figure 7 Backend compression strategy

4 Experiment and analysis

4.1 Model training and result analysis

With an NVIDIA GeForce GTX 1080 GPU, MobileNetV2 was trained using a pre-trained model. In the experiment, the original training set and the augmented training set were used for the training. The compositions of the original training set and the enhanced training set are listed in Table 1. As shown in Figure 8, MobileNetV2 represents the network using the original training set, and aug_MobileNetV2 represents the network using the augmented training set. After the network model was trained for 100 000 iterations, the loss values level off at about 0.2, the training time of MobileNetV2 and aug_MobileNetV2 are both 7 h, and the Top-1 accuracy rate, which refers to the accuracy of the first ranked category in all the results predicted by a model, of the two models in the validation set is 98.5% and 97.7%, respectively. It can be seen that when the augmented training set is used for training, the Top-1 accuracy rate has improved by nearly 1%, indicating that MobileNetV2, after being trained with the augmented training set, effectively mitigates overfitting, and has better generalization capability.

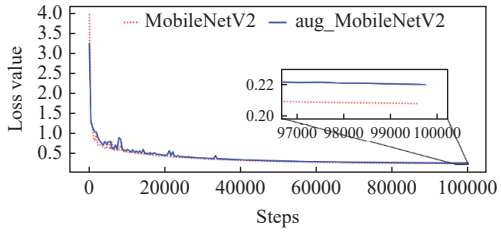


Figure 8 Changes in loss values in training process

To examine the MobileNetV2 network classification results, ResNetV2-50[33] and Inception-V3[34] were included to conduct the comparative experiment. The corresponding pre-trained models were obtained by transfer learning. The three networks were separately trained using the same augmented headland image datasets and the same set of hyperparameters. After the network training was completed, the validations were conducted using the validation set. The recognition results are listed in Table 4.

Table 4 Performance comparison of three convolutional neural networks

No.	Network name	Size of input image	Top-1 accuracy/%	Recognition speed/s per image	Memory footprint/MB	Training time/h
1	ResNetV2_50	224×224	98.4	4.89	505.3	7
2	Inception_V3	299×299	98.8	4.49	483	11
3	MobileNetV2	224×224	98.5	1.30	119.49	7

It can be seen from Table 4 that although three convolutional neural networks have high recognition accuracies, ResNetV2-50 and Inception-V3 consume a lot of computer resources and have lower recognition speeds, thus they are not suitable for direct application to embedded devices. In contrast, the MobileNetV2 classification accuracy was comparable to those two networks, but it has a greater advantage in memory use, which makes it possible to be deployed deep on onboard computers.

4.2 Comparative experiment of frontend compression and backend compression

An experiment was conducted for comparing the model performance of frontend compression with the performance of backend compression. According to model training experimental results listed in Table 4, Inception-V3 which has better performance was used to perform the backend compression. The model that resulted from the backend compression was referred to as Press_Inception-V3. The validation was conducted using the validation set in an i5-8250U CPU computing platform. The recognition results are listed in Table 5.

Table 5 Performance comparison of Inception-V3, Press_Inception-V3, and MobileNetV2

No.	Network name	Size of input image	Top-1 accuracy/%	Recognition speed/ s·image^{-1}	Memory footprint/ MB
1	Inception_V3	299×299	98.8	4.49	483.00
2	Press_Inception_V3	299×299	98.8	4.20	469.00
3	MobileNetV2	224×224	98.5	1.30	119.49

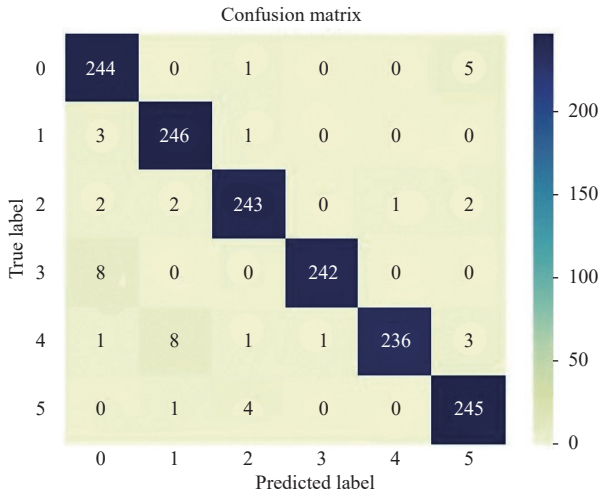
After the Inception-V3 model was compressed, the accuracy remained the same. The recognition speed and memory footprint were enhanced slightly but were still far below the performance of MobileNetV2. The comparison of the experimental results of the two compression methods showed that the MobileNetV2 has better network performance, and was the deep learning model more suitable for deployment in mobile or embedded devices.

4.3 Experiment and analysis of network classification accuracy

To verify the actual classification accuracy of the MobileNetV2 network, a network generalization test experiment with the test set on MobileNetV2 after the final hyper-parameter training was conducted in the study. The confusion matrix and F1-score are used to evaluate the model in the test of the MobileNetV2 network. The number of samples in the test set is listed in Table 1.

4.3.1 Confusion matrix and F1-score

In evaluating the accuracy of image recognition, the precision, recall, and F1-score can be calculated using the confusion matrix. Figure 9 shows the confusion matrix of the six classes of headland images. Each column in the matrix represents the predicted classes. The total data number in each column represents the number of data predicted to be in the class. Each row represents the true class the data belongs to. The total data number in each row indicates the number of true classes. Precision refers to the ratio of the number of positives the model correctly predicted to the total number of the



Note: The meaning of the values of axes: 0-headlands with bare soils; 1-fields with crops; 2-headlands with yellow vegetation; 3-headlands with man-made objects; 4-fields without crops; 5-headlands with green vegetation.

Figure 9 Confusion matrix from MobileNetV2

predicted positives. Recall refers to the ratio of the positives correctly predicted by the model to the total number of the predicted positives. F1-score can be regarded as a harmonic mean of the precision and recall of the model. Its maximum value is 1 and its minimum value is 0 and is calculated by

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recall} + \text{Precision}} \tag{13}$$

The results of the precision, recall, and F1-score of the six types of headland images are listed in Table 6. The average of the precisions and the average of the recalls of MobileNetV2 on the test set are 0.97, respectively, indicating that the network can accurately recognize each class of the samples. The average of F1-scores is 0.97, indicating that the MobileNetV2 network can accurately recognize the six classes of headlands in the natural environment, and has good robustness and stability.

Table 6 MobileNetV2 network test experimental results

Class of cropland headland	0	1	2	3	4	5	Average value
Precision	0.95	0.96	0.97	0.99	0.99	0.96	0.97
Recall	0.98	0.98	0.97	0.97	0.94	0.98	0.97
F1-score	0.96	0.97	0.97	0.98	0.96	0.97	0.97

Note: Precision is the proportion of all positive predictions that are correct; Recall is the proportion of all real positive observations that are correct; F1-score is the harmonic mean of precision and recall.

4.3.2 Error analysis

In this study, the incorrectly recognized cropland headland images were sorted and analyzed, as shown in Figure 10. The primary recognition errors occur for the following reasons:

a. Predicted: headland with green vegetation
actual: headland with bare soils

b. Predicted: field with crops
actual: field without crops

c. Predicted: headland with yellow vegetation
actual: field with crops

d. Predicted: headland with bare soils
actual: headland with man-made objects

e. Predicted: headland with bare soils
actual: headland with man-made objects

f. Predicted: headland with yellow vegetation
actual: headland with green vegetation

Figure 10 Incorrectly recognized images

1) Disturbance from large shadows in the images, as shown in Figure 10a;

2) Lighting conditions result in a great change in the soil color, as shown in Figure 10b;

3) Some images in different categories are similar. Figure 10c shows a field actually with arable crops. Because a clear dividing line is present in the image, the model mistakes it for a headland with yellow vegetation. In addition, some man-made objects and bare soils appear very similar, as shown in Figures 10d and 10e;

4) Because the field environment is quite complex, the color and texture features of some scenes are not captured, and the model is not familiar with these features, as shown in Figure 10f.

The error analysis shows that the model still has shortcomings. The model fails to correctly recognize the images of headlands with a high degree of similarity, the images with large shadow areas, and the images with blur field boundaries. Further improvement and optimization are needed on this model.

5 Conclusions

1) To meet the requirement for self-driving agricultural vehicle

headland turning, a cropland headland image annotation dataset covering six types of images was constructed. This dataset contained a total of 9000 images, 6000 of which were in the training set and 3000 of which were in the validation set. This dataset could be intelligently used for the automatic recognition of cropland headland.

2) The compact network MobileNetV2 was trained using the training set of augmented headland images. The Top-1 accuracy of the MobileNetV2 network on the validation set was 98.5%, which was similar to that of ResNetV2-50 and Inception-V3, but the recognition speed and memory footprint of the MobileNetV2 network was much better than ResNetV2-50 and Inception-V3. Compared with the mainstream large-scale deep networks, MobileNetV2 has remarkable advantages in meeting the requirement of deployment in onboard computers. In addition, compared with a backend-compressed network, Press_Inception-V3, MobileNetV2 has better network performance.

3) The test set was used to further test the generalization ability of the network. The average of the F1 scores of the MobileNetV2 network in recognition of the six classes of headland images was 97%, indicating that the network was robust and can perform the recognition task well in the natural environment.

In conclusion, a cropland headland image annotation dataset was constructed according to the headland environment-aware application requirement. This was followed by training the compact network MobileNetV2, which is more suitable to be deployed on embedded devices. The future work will improve the recognition efficiency, and apply the model to the onboard computer of self-driving agricultural vehicles to realize automatic recognition of cropland headland.

Acknowledgements

This work was financially supported by the National Nature Science Foundation of China (Grant No. 31971800), and the National Key Research and Development Project of China (Grant No. 2019YFB1312304).

[References]

[1] Hu J T, Gao L, Bai X P, Li T C, Liu X G. Review of research on automatic guidance of agricultural vehicles. Transactions of the CSAE, 2015; 31(10): 1–10. (in Chinese)

[2] Csornai G, László I, Suba Z, Nádor G, Bognár E, Hubik I, et al. 2007. The integrated utilization of satellite images in Hungary: Operational applications from crop monitoring to ragweed control. In: New Developments and Challenges in Remote Sensing, 2007; pp.15–23.

[3] Chen J, Chen T Q, Mei X M, Shao Q F, Deng M. Hilly farmland extraction from high resolution remote sensing imagery based on optimal scale selection. Transactions of the CSAE, 2014; 30(5): 99–107. (in Chinese)

[4] Bay H, Tyutelaars T, Van Gool L. SURF: Speeded up robust features. In: Computer Vision - ECCV 2006, Springer, 2006; pp.404–417. doi: 10.1007/11744023_32.

[5] Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004; 60(2): 91–110.

[6] Watanabe T, Ito S, Yokoi K. Co-occurrence histograms of oriented gradients for human detection. IPSJ Transactions on Computer Vision and Applications, 2010; 2: 39–47.

[7] Lecun Y, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998; 86(11): 2278–2324.

[8] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2012; 60(6): 84–90.

[9] Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015; pp.1–9.

[10] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv, 2014; arXiv: 1409.1556.

[11] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision & Pattern Recognition (CVPR), IEEE, 2016; pp.770–778.

[12] Zhu S P, Zhu J X, Huang H, Li G L. Wheat Grain Integrity Image Detection System Based on CNN. Transactions of the CSAM, 2020; 51(5): 36–42. (in Chinese)

[13] Yang K, Liu H, Wang P, Meng Z J, Chen J P. Convolutional neural network-based automatic image recognition for agricultural machinery. Int J Agric & Biol Eng, 2018; 11(4): 200–206.

[14] Zhao L X, Hou F D, Lu Z C, Zhu H C, Ding X L. Image recognition of cotton leaf diseases and pests based on transfer learning. Transactions of the CSAE, 2020; 36(7): 184–191. (in Chinese)

[15] Xu J H, Shao M Y, Wang Y C, Han W T. Recognition of corn leaf spot and rust based on transfer learning with convolutional neural network. Transactions of the CSAM, 2020; 51(2): 230–236, 253. (in Chinese)

[16] Kim W-S, Lee D-H, Kim T, Kim G, Kim H, Sim T, et al. One-shot classification-based tilled soil region segmentation for boundary guidance in autonomous tillage. Computers and Electronics in Agriculture, 2021; 189: 106371.

[17] He Y, Zhang X Y, Zhang Z Q, Fang H. Automated detection of boundary line in paddy field using MobileV2-UNet and RANSAC. Computers and Electronics in Agriculture, 2022; 194: 106697.

[18] Qiao Y J, Yang P S, Meng Z J, Wang Q, Liu H. Detection system of headland boundary line based on machine vision. Journal of Agricultural Mechanization Research, 2022; 44(11): 24–30. (in Chinese)

[19] GB/T 21010-2017. Current land use classification. Ministry of Natural Resources, China, 2017. (in Chinese)

[20] Addo K A. Urban and peri-urban agriculture in developing countries studied using remote sensing and in situ methods. Remote Sensing, 2010; 2(2): 479–513.

[21] Dwork C, Feldman V, Hardt M, Pitassi T, Reingold O, Roth A. The reusable holdout: Preserving validity in adaptive data analysis. Science, 2015; 349(6248): 636–638.

[22] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint, 2015; arXiv: 1503.02531.

[23] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions. arXiv preprint, 2014; arXiv: 1405.3866.

[24] Li H, Kadav A, Durdanovic I, Samet H, Graf H P. Pruning filters for efficient convnets. arXiv preprint, 2016; arXiv: 1608.08710.

[25] Cai Z W, He X D, Sun J, Vasconcelos N. Deep learning with low precision by half-wave Gaussian quantization. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017; pp.5406–5414. doi: 10.1109/CVPR.2017.574.

[26] Weiss K, Khoshgoftaar T M, Wang D D. A survey of transfer learning. Journal of Big Data, 2016; 3: 9.

[27] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 2015; 115(3): 211–252.

[28] Sandler M, Howard A, Zhu M L, Zhmoginov A. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018; pp.4510–4520. doi:10.1109/CVPR.2018.00474.

[29] Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W J, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint, 2017; arXiv: 1704.04861.

[30] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. Journal of Machine Learning Research, 2011; 15: 315–323.

[31] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014; 15(1): 1929–1958.

[32] Şimşekli U, Zhu L, Teh Y W, Gürbüzbalaban M. Fractional underdamped langevin dynamics: Retargeting SGD with momentum under heavy-tailed gradient noise. arXiv preprint, 2020; arXiv: 2002.05685.

[33] He K M, Zhang X Y, Ren S Q, Sun J. Identity mappings in deep residual networks. In: Computer Vision - ECCV 2016, Springer, 2016; pp.630–645.

[34] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016; pp.2818–2826.