Journal of Computers Vol. 32 No. 3, 2021, pp. 172-187
doi:10.3966/199115992021063203013

Research on Increasing the Robustness of Image Recognition Models Against Different Sets

Zhi Tan*, Xing-Ye Liu

School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, 102616, Beijing, China
tanzhi@bucea.edu.cn, liu_xing_ye@163.com

Received 21 September 2020; Revised 15 October 2020; Accepted 21 October 2020

Abstract. In view of the problem that the recognition performance of the image recognition model trained by a specific dataset is significantly reduced after being transplanted to different dataset, an improved algorithm for attribute combination optimization based on the minimal weighted random search (MinW-Rsearch) and "Equal-Sum" (E-S) judgment method is proposed. First, the MinW-Rsearch algorithm is used to search the image attribute combination and the searched attribute combination is filter through the E-S judgment method. Then, the image that being transformed by the selected attribute combination is input to the improved neural network. Finally, the Adam optimization algorithm is used to train the model. The improved model was transplanted after training, and many experiments were carried out to compare the average classification accuracy of the transplanted model with that of the original model. The experimental results show that the recognition accuracy is increased by at least 5% after the improved digital recognition model is transplanted, which obviously improves the robustness of the recognition performance after the model is transplanted.

Keywords: image recognition, combinatorial optimization, model transplantation

1 Introduction

At present, image recognition has been widely used and developed in machine learning. However, in research and experiments, it will be found that an image recognition model trained on a specific dataset is used to recognize other datasets, the recognition accuracy of the model will be significantly reduced. This means that it is less robust to different sets. This phenomenon is usually called "distribution Shift" [1]. Which has been fully verified in the experiment of Antonio et al. [2], and attributed the reason to "dataset bias".

In response to the above problems, Volpi et al. [1] adopted a combined optimization model to improve the robustness of the image recognition model against different set. However, in the optimization process, there are defects such as relatively low complexity of optimized combination and uneven selection of attributes, which limits the performance of the model.

This paper proposes an improved attribute combination optimization algorithm, which can improve the defects in the algorithm of Volpi et al. The experimental results show the optimized algorithm can effectively improve the "distribution shift", so that the image recognition model is transplanted to obtain higher recognition accuracy. The minimal weighted random search (MinW-Rsearch) algorithm is used in the improved attribute combination optimization algorithm. This search method can control the search direction. It overcomes the shortcomings of the simple random search method that is used by Volpi et al. in the uneven search distribution, so that the search for image attributes is more balanced. At the same time, the improved attribute combination optimization algorithm also includes the "Equal-Sum" (Equal-Sum, ES) judgment method. This judgment method achieves the purpose of judging whether a certain combination of attributes needs to be changed, while reducing computational complexity. In addition, in

* Corresponding Author

this work, the structure of the neural network was changed so that it can more fully extract the image features. To summarize, an improved attribute combination optimization algorithm used to improve the performance of the model was proposed and this work makes the following contributions:

(1) A minimal weighted random search (MinW-Rsearch) algorithm is proposed, which can control the system to search in the direction of distribution balance.

(2) An "Equal-Sum" (E-S) judgment method is proposed to judge whether the searched attribute combination needs to be optimized.

(3) In this work, the complexity of the optimized combination is increased, and a small ConvNet with 3 convolutional layers is constructed based on the Leaky Relu activation function.

(4) According to MinW-Rsearch algorithm and E-S judgment method, the ResNet is used for optimization.

Section 2 summarizes the recent solutions to the impact of "distribution shift" and related research literature; Section 3 proposes several improved algorithms and introduces the implementation process of the algorithm in detail; Section 4 uses experimental data to verify the proposed algorithm and analyze the experimental results; Section 5 discusses and analyzes the experimental results; Section 6 makes conclusions and prospects for the next step.

2 Related Work

Antonio et al. [2] proved the influence of dataset bias on machine learning models, and came to the following conclusions: (1) Although the creator has made the greatest efforts in the process of creating the dataset, the built-in bias of the dataset still seems to be tenacious; (2) Although the built-in bias of most datasets can be solved by using different datasets for different goals, they still exist in a certain form of uncertainty; (3) When a typical object detector trained on a representative dataset is tested on other representative datasets, the performance will decrease. The reason for this result is likely to be the built-in bias of the dataset.

Related studies have shown that deep learning models may have inherent weaknesses, so they are susceptible to certain interferences [4-5]. To solve this problem, Madry et al. [6] studied neural networks from the perspective of robust optimization, and proposed a more robust deep learning model. However, Hosseini et al. [7] tested the model proposed by Madry et al. [6] by using color deviation images, and the experimental results explained the accuracy of the model decreased significantly. This means that even if the model has strong robustness, it cannot eliminate the influence of the bias of the dataset.

The latest progress in deep learning also exposes there are unknowable image disturbances, which may change the prediction of image labels by the most advanced network classifiers [8]. Machine learning systems will expose corresponding vulnerabilities in different situations, which has attracted widespread attention in the field of machine vision models.

A large number of documents [5, 9] have conducted extensive research on the solution to this problem. Among them, Szegedy et al. [5] proved that although deep neural network classifiers perform well in identifying challenging classification tasks, these classifiers are extremely vulnerable to imperceptible adversarial disturbances. The research of Seyed et al. [9] showed that a general small disturbance vector can cause natural image classification errors. They also proposed a system algorithm for calculating general interference. The results explain that, although it is almost invisible to the human eye, advanced deep neural networks are extremely susceptible to such interference.

For such disturbances, the field of "content retention" [1] has received extensive attention from researchers. The image is converted through a series of "content retention" operations to modify the input, and the transformation performed during the "content retention" process does not modify the essential content of the image, as shown in Fig. 1. It will only modify its drawing method, for example, modify RGB intensity, enhance contrast, etc. And there are many documents [10, 7] that have studied this field, and strive to improve the impact of disturbances through the form of "content retention".

At the same time, researchers noticed the research on domain adaptability [11-14]. Among them, in domain generalization [3, 15-19], the problem of invisible distribution interference is solved. Volpi et al. [3] considered the worst-case expression of the data distribution close to the source domain in the feature space. They also propose an iterative process of adaptive data augmentation, which uses examples from the virtual target domain to augment the dataset. This method can improve the performance of number recognition and semantic segmentation tasks in a series of a priori unknown target domains.

Fig. 1. The left image is the original image. The two images at the right are the original images after changing a certain attribute. The essential content of the image has not changed during the whole process

In this context, Volpi et al. proposed a combinatorial optimization problem based on "content retention" [1]. Standard search algorithms can be used to evaluate and process the more susceptible areas in the image space of a given model. Experimental results show that their proposed model is more portable than previous models.

3 Improved Algorithm for Attribute Combination Optimization

Volpi et al [1]. performed feasible optimization of the recognition model based on the combinatorial optimization problem, which effectively improved the phenomenon of "distribution shift". However, there are defects in the optimization process, which limits the performance of the model after being transplanted. Mainly manifested in the following aspects: (1) Random search is a very powerful baseline, and this is the simplest method that can be used, so it is worth exploring [1]. Therefore, Volpi et al. used a simple random search method in the process of searching for harmful attribute combinations. However, the search direction cannot be controlled during the search process, which may cause too many or too few searches for certain combinations, thereby affecting the recognition effect of the model. (2) They need to calculate the model's temporary recognition accuracy of the image when judging whether a certain attribute combination of the image needs to be optimized. This method increases the complexity of the calculation. (3) The complexity of the attribute combination they set is too low and the neural network model is relatively simple, which will make it impossible to fully extract the features of the image.

3.1 Improved Attribute Combination Optimization Algorithm

Faced with the problems of the original model mentioned above, this section will propose solutions to each problem. For the search problem, you can use the minimal weighted random search (MinW-Rsearch) algorithm to solve; for the calculation problem, this paper proposes "Equal-Sum" (Equal-Sum, ES) judgment method; for the last problem, the method given in this work is to increase the complexity of the combination (This way will be explained in section 3.2.) and improve the neural network. In addition to improving the neural network used in the original model, another feasible neural network is also proposed.

3.1.1 Minimal Weighted Random Search Algorithm

In order to prevent the system without the direction of search, the MinW-Rsearch algorithm is used. According to the weight value, the probability of being searched for an object with a small search weight value is increased. On the contrary, reduce the probability of being searched for objects with larger search weights, so as to achieve the goal of balance. The specific implementation process of MinW-Rsearch algorithm is as follows.

(1) Determine the initial values of W_i and P_{si}. Before searching, set the weight of the i-th object as $W_i = 1 \ \ i \in [1, N]$, the specific form is:

$$W_1 = W_2 = ... = W_i = 1 \ \ i \in [1, N]. \tag{1}$$

Where N is the total number of objects, it can be obtained that the probability of the i-th object being searched (P_{si}) is the same during the initial period. The specific form is:

$$P_{s1} = P_{s2} = ... = P_{si} = \frac{1}{N} \quad i \in [1, N]. \tag{2}$$

(2) Each time an object is found, its weight value increases by 1. The weight of the i-th object is W_{sni} when searching for this object for the n-th time, and the specific form is:

$$W_{sn} = \{W_{sn1}, W_{sn2}, ... W_{sni}\} \quad i \in [1, N]. \tag{3}$$

Then the weight's probability of the i-th object at the n-th time is P_{wsni}, and the specific calculation method is:

$$P_{wsni} = \frac{W_{sni}}{\sum_{1}^{N} W_{sni}}. \tag{4}$$

However, the weight's probability is larger, which means that the object has been selected more times. In order to achieve the purpose of selecting equilibrium, it is necessary to reduce the possibility of it being searched in the next search process. In other words, it needs to reduce its searched probability (The probability of object being searched), defined the searched probability as P_{ssni} $i \in [1, N]$. Conversely, if the P_{wsni} of the object is small, its P_{ssni} needs to be increased.

In order to satisfy the above relationship between the P_{wsni} and the P_{ssni}, the P'_{ssni} $i \in [1, N]$ is defined as the expression form of the P_{ssni}. P'_{ssni} and P_{ssni} have the same properties, but their values are different. This paper approximates the P_{wsni} as the expression form of the probability that object not being searched. Correspondingly, 1-P_{wsni} is set to P'_{ssni}, which is the expression form of the probability of the object being searched (P_{ssni}). The specific calculation method of P'_{ssni} is:

$$P'_{ssni} = 1 - P_{wsni} = 1 - \frac{W_{sni}}{\sum_{1}^{N} W_{sni}} = \frac{\sum_{1}^{N} W_{sni} - W_{sni}}{\sum_{1}^{N} W_{sni}}. \tag{5}$$

Because the sum of the expressions form of all searched probabilities is not equal to 1, so P'_{ssni} cannot be used as the probability of an event that the object can be searched (P_{ssni}). Further, the P_{ssni} of each object can be obtained from the proportion of the P'_{ssni} of each object in the P'_{ssni} of all objects. The specific form is:

$$P_{ssni} = \frac{P'_{ssni}}{\sum_{1}^{N} P'_{ssni}} = \frac{\frac{\sum_{1}^{N} W_{sni} - W_{sni}}{\sum_{1}^{N} W_{sni}}}{\sum_{1}^{N} \frac{\sum_{1}^{N} W_{sni} - W_{sni}}{\sum_{1}^{N} W_{sni}}} = \frac{\sum_{1}^{N} W_{sni} - W_{sni}}{\sum_{1}^{N} (\sum_{1}^{N} W_{sni} - W_{sni})}. \tag{6}$$

(3) Based on the P_{ssni} of each object (P_{ssn1}, P_{ssn2}, ..., P_{ssni} $i \in [1, N]$), the objects with smaller weights are searched according to higher P_{ssni}, and the objects with larger weights are searched according to lower P_{ssni}.

Through the MinW-Rsearch algorithm, the search direction can be biased towards objects with smaller weights. Appropriately "ignore" objects with larger weights to achieve the purpose of search balance.

3.1.2 E-S Judgment Method

To achieve the purpose of judging whether the objects selected by subsection 3.1.1 need to be optimized, an E-S judgment method is proposed to screen them. The expression of E-S is as follows:

(1) Use the Equal function to determine whether the predicted value y^* is equal to the true value y. If the predicted value is equal to the true value, the return value of the function is 1, otherwise it is 0:

$$\delta_i = \begin{cases} 0, & y_i^* \neq y_i \\ 1, & y_i^* = y_i \end{cases} \quad i \in [1, N]. \tag{7}$$

Among them, y_i^* is the predicted value of the i-th target, y_i is the true value of the i-th target, δ_i is the return value of the Equal function, and N is the number of targets predicted each time.

(2) The Sum function is used to sum δ. If the return value of the Sum function is equal to N, it indicates that the object selected based on section 3.1.1 has the best prediction effect on the target element. If the return value of the Sum function is equal to 0, the prediction effect is the worst. If the return value of the Sum function is between 0 and N, it indicates that the prediction effect is general.

$$S = \sum_{1}^{N} \delta_i . \tag{8}$$

S is the return value of the Sum function. In order to make the model achieve the best prediction effect, this paper sets N (the total number of targets for each prediction) as the threshold of the judgment method (define it as f). Extract the objects that make the return value of the Sum function less than N.

Compared with the screening method used by Volpi et al., the E-S judgment method reduces the complexity of further calculation accuracy, and also achieves the purpose of screening objects.

3.1.3 A Small ConvNet

This paper improves the small convolutional neural network used in the original model. A neural network with three convolutional layers is constructed. And the Leaky Relu activation function is used in the neural network, which improves the performance of the neural network. The structure of the small ConvNet is shown in Fig. 2.

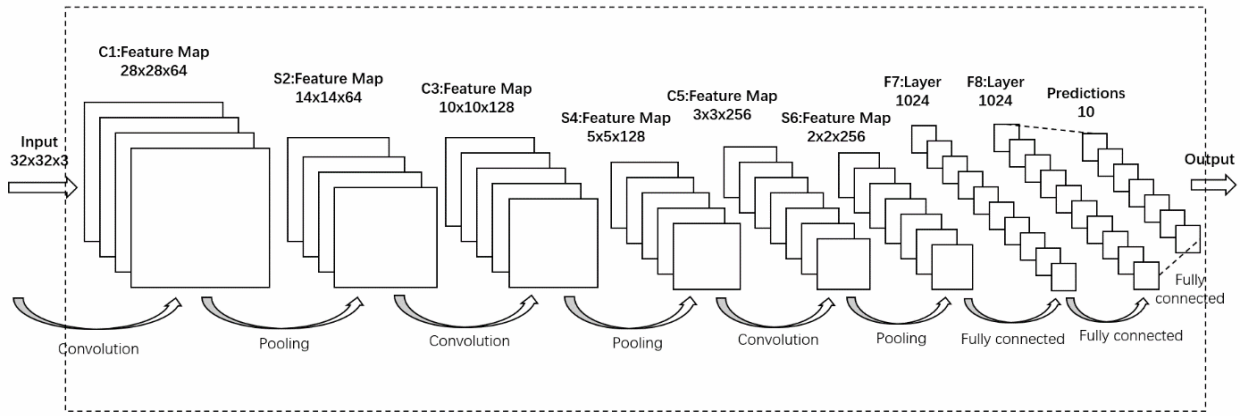


Fig. 2. The structure of the small ConvNet

3.1.4 Another Feasible Neural Network

This work further studies another feasible image recognition model. Compared with the original model, the new neural network is based on ResNet. The definition of ResNet neural network structure is as follows: First, for the input data, convolution and pooling operations are performed in sequence. Then access four residual blocks (sequentially, residual blocks with the same input and output shapes, residual blocks with different input and output shapes, residual blocks with the same input and output shapes, and residual blocks with different input and output shapes). Finally connect the two fully connected layers to get the output of the network. The specific neural network structure flow is shown in Fig. 3.

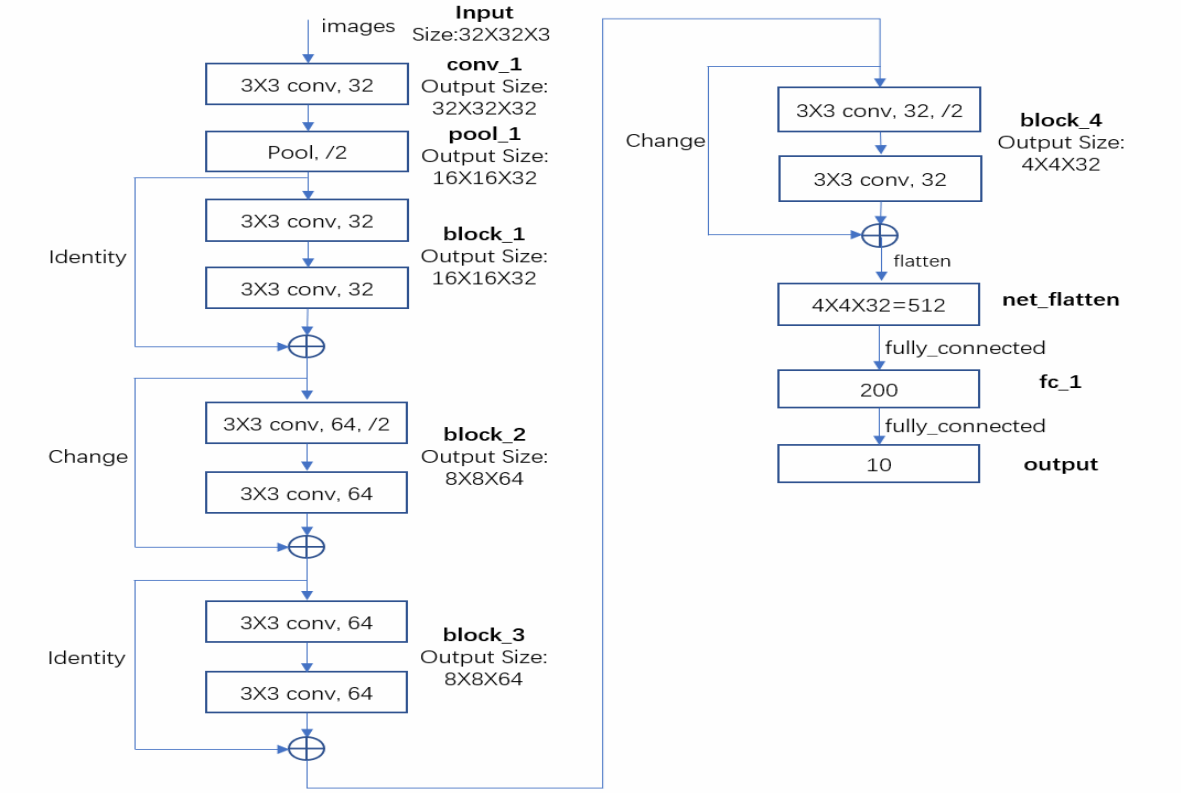


Fig. 3. Structural flow chart of ResNet

3.2 The Algorithm's Framework

(1) The selection of image attributes is the same as Volpi et al. [1]: autocontrast (20), sharpness (20), brightness (20), color (20), contrast (20), grayscale conversion (1), R-channel enhancer (30), G-channel enhancer (30), B-channel enhancer (30), solarize (20). The value in brackets is the available strength level, which is the value to be optimized for each attribute, and there are 211 optimizable attributes. These attributes will be described in Table 1. T' is used to represent all image attributes. The t_i represents the i-th image attribute, all attributes of the image can be expressed as:

$$T' = \{t_1, t_2, \ldots, t_{211}\}. \tag{9}$$

Table 1. The various parameters and their values

Attributes	Range	Values
Autocontrast	[0.0, 0.3]	Generate 20 evenly spaced data in the range
Brightness	[0.6, 1.4]	Generate 20 evenly spaced data in the range
Color	[0.6, 1.4]	Generate 20 evenly spaced data in the range
Contrast	[0.6, 1.4]	Generate 20 evenly spaced data in the range
Sharpness	[0.6, 1.4]	Generate 20 evenly spaced data in the range
Solarize	[0.0, 20.0]	Generate 20 evenly spaced data in the range
Grayscale	—	—
R-channel enhancer	[−120, 120]	Generate 30 evenly spaced data in the range
G-channel enhancer	[−120, 120]	Generate 30 evenly spaced data in the range
B-channel enhancer	[−120, 120]	Generate 30 evenly spaced data in the range
N_ε (the number of combination's elements)	—	5
l (learning rate)	—	0.0001
total_times (line 4 of Algorithm 4)	—	100
search_times (line 5 of Algorithm 4)	—	100
train_times (line 9 of Algorithm 4)	—	10000

(2) The T is image attribute combination. It is used to optimize the image and complete the conversion of some basic attributes of the image. Randomly select several (assuming n) image attributes from T' and combine them. The basic form of T is:

$$T = \{t_1, t_2, \ldots, t_{N_\varepsilon}\}. \tag{10}$$

In the improvement of this work, we increase the complexity of the attribute combination by increasing the number of elements (The N_ε is used to represent the number of elements) in the attribute combination. The number of elements in the combination has been increased from 3 in the original model to 5 now. It can be concluded that there are 211^5 combinations of T. The MinW-Rsearch algorithm is used to select five image attributes, and the process is shown in Algorithm 1.

Algorithm 1. Construct T

Input: $P_{ssni}\ i \in [1, 211]$

Output: image attribute combination with 5 elements

Initialize: $T = [\ ],\ T',\ N_\varepsilon$

for i in range N_ε **do**

Random Search t_i from T' by (P_{ssn1}, P_{ssn2}, ..., P_{ssn211})

$T = T$**.** append (t_i)

end for

Return T

According to the Algorithm 1, the final form of the attribute combination T is:

$$T = \{t_1, t_2, t_3, t_4, t_5\}. \tag{11}$$

(3) When constructing attribute combination T, part of the data in the entire dataset is used and it can be represented by D. Its basic form is:

$$D = \{(x_i, y_i), i \in [0, N]\}. \tag{12}$$

Where x_i is the image, y_i is the label of the image and N is the total number of this part of data.

(4) The prediction of the image is completed through the neural network, and its basic form is:

$$y_i^* = M(x_i). \tag{13}$$

Where y_i^* is the predicted value, and M is the neural network model used for image recognition in the experiment. For example, in the improvement process, the Leaky Relu activation function is used to construct a small ConvNet (Fig. 2).

However, the design idea of the experiment is to use the converted image as the input of the model. Before inputting the model, it is necessary to change the attributes of each image through the attribute combination T. Combining formula (11) and formula (12) to optimize formula (13), the process is shown in Algorithm 2.

Algorithm 2. The optimization process of formula (13)

Input: $D = (x_i, y_i), i \in [0, N],\ T$

Output: the predicted value of an image

for t in T **do**

Change t of x_i

end for

$y_i^* = M(x_i)$

Return y_i^*

According to the Algorithm 2, the image prediction process can be summarized as:

$$y_i^* = M\{T[D(x_i, y_i)]\}, i \in [0, N]. \quad \textbf{(14)}$$

(5) Screen the combination that needs to be optimized. According to the above four steps (1) to (4), the predicted value of the image optimized by T has been obtained. Next, we need to determine whether we need to continue to optimize the attributes of this combination on subsequent images. Here we need to use the E-S judgment method proposed in section 3.1.2. The larger the return value of the E-S judgment method is, the image with the attributes in the attribute combination can be easily recognized by the recognition model. Therefore, the subsequent process does not need to spend more time to learn the parameters corresponding to these attributes. On the contrary, you only need to learn the parameters corresponding to the attribute combinations that make the return value of the E-S judgment method smaller.

As mentioned in section 3.1.2, in order to make the model achieve the best prediction effect, this paper sets N (the total number of targets for each prediction) as the threshold f of the judgment method. Extract the attribute combination when the return value of the Sum function is less than N. Combining the MinW-Rsearch algorithm and the E-S judgment method, the attribute combination that needs further optimization can be searched out. The specific implementation process is shown in Algorithm 3.

Algorithm 3. Search for attribute combinations that need further optimization

Input: $D = (x_i, y_i), i \in [0, N] \leftarrow$ part of the data in the entire dataset, T', M

Output: the attribute combinations that need to be further optimized

Initialize: $W_i = 1 \ \ i \in [1, 211], \ T = [\ ], f =$ N, N_ε

for i in range N_ε **do**

$W_{sn} \leftarrow$Calculate it by formula (1) and formula (3)

$P_{ssni} \leftarrow$Calculate it by formula (6) according to W_{sn}

$t_i \leftarrow$Random Search from T' according to step (3) in section 3.1.1

$T = T.$ append (t_i)

end for

for (x_i, y_i) in $D=(x_i, y_i)\ i \in [0, N]$ **do**

$y_i^* \leftarrow$ Calculate it by formula (14)

S $\leftarrow$ Calculate it by formula (7) and formula (8)

end for

if S <= f

Return T

end if

To achieve the purpose of more convenient to use in the subsequent process, the T that be selected needs to be stored in a storage space in turn:

$$T^* = T^*. \text{ append } (T). \quad \textbf{(15)}$$

Among them, T^* is the space for storing "harmful" attribute combinations, and the initial value is empty.

3.3 The Implementation Process of the Optimization Algorithm

In summary, according to a set of labeled sample dataset D, an attribute combination T selected by the MinW-Rsearch algorithm, a recognition model M, E-S judgment method, and a screening threshold f can be used to define the optimization model. The process is shown in Fig. 4.

Combining the flow chart of the optimization process and the process of searching for "harmful" attribute combinations, the process of the entire optimization algorithm is detailed. During the training process, the Adam optimization algorithm is used to train the optimization model. The weights of the features of the image (the image optimized by the optimized combination) are updated by gradient descent, so as to improve the performance of the model. The specific algorithm is shown in Algorithm 4.

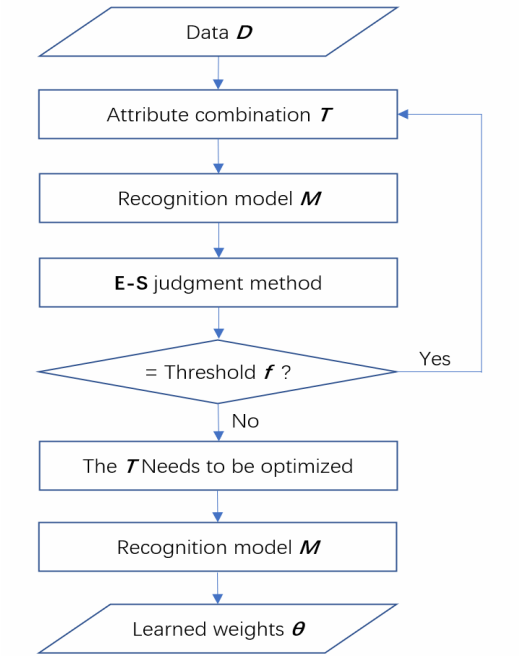


Fig. 4. Flow chart of the optimization process

Algorithm 4. The process of optimization

Input: dataset, T', M
Output: learned weights
Initialize: $T^* = [\,]$, $l \leftarrow$learning rate
for time in total_times **do**
  for search_time in search_times **do**
    $T \leftarrow$Search it through the process in Algorithm 3
    $T^* = T^*.\text{append}(T)$
  end for
  for train_time in train_times **do**
    Select T from T^*
    $D^* = (x_i, y_i)\ i \in [0, batch\_size]$ ←the entire dataset needs to be trained in batches
    for t in T **do**
      Change t of x_i
    end for
    $y_i^* \leftarrow$Calculate it by formula (14)
    $L = \text{loss}\ (y_i^*, y_i)$
    $\partial_\theta = \partial L / \partial \theta$
    $\theta = \theta - 1 * \partial_\theta$
  end for
end for
Return θ

4 Experiment

4.1 The Principle of Experiment

First, through an improved algorithm, five attribute values of the image are randomly selected to form an attribute combination, and these attributes of the training image are changed. Then, a series of changed

images are input to the neural network, and the model training is completed through gradient descent. The specific experimental principle process is shown in Fig. 5.

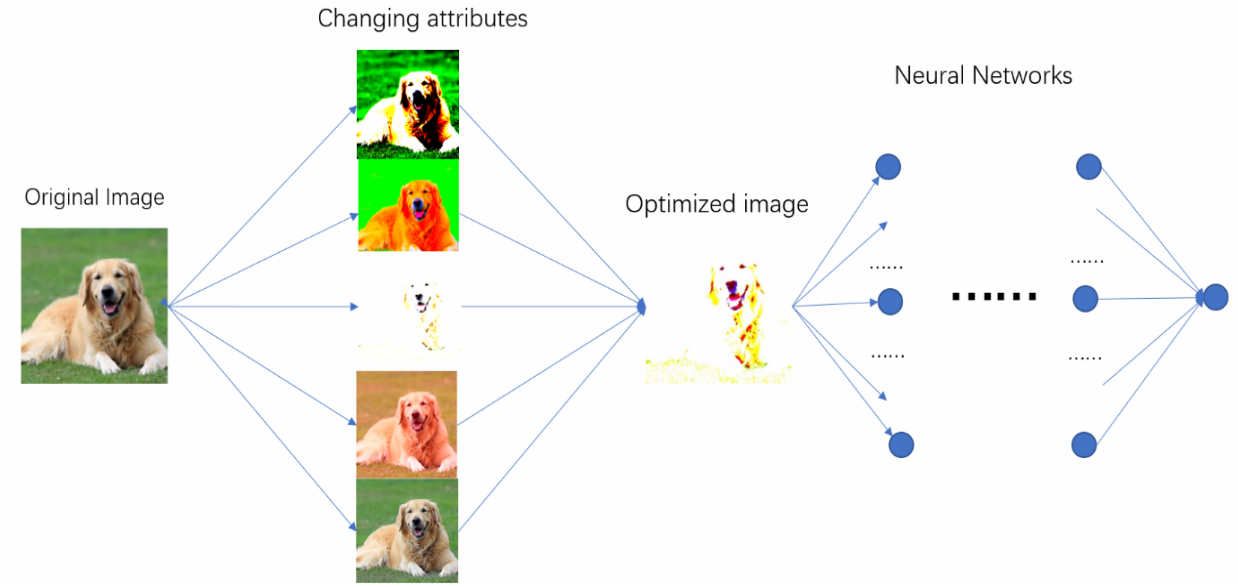


Fig. 5. The original image is optimized by attribute combination (for example, the changed attributes in the Fig. are the image contrast, color, brightness, RGB channel intensity and sharpness from top to bottom), and then the neural network is trained with the optimized image

4.2 Training and Testing the Improved Model with A Small ConvNet

In order to eliminate some accidental factors caused by the selection of different image attributes, and to make the comparison between the effect obtained by the improved algorithm and the effect obtained by the original algorithm more convincing, this improvement process uses the same image attributes as Volpi et al. [1]. The specific attributes and their values are detailed in Table 1. All the operations of changing attributes in the experiment are implemented based on the PIL package in Python.

There are 211^5 combinations of attributes, so at least 211^5 times training can be done to complete all combinations. The number of training times is too large. Both reducing the times of training and achieving a better comparison effect are very important. The model was trained 10^6 times in the experiment (Volpi et al. also trained 10^6 times). In the experiment, the values of some parameters in the algorithm are also explained in Table 1.

The training dataset is MNIST [20], and the testing dataset used to verify the accuracy of the improved model after being transplanted is SVHN [21]. No other data sets will be used during the experiment, in turn to ensure the reliability of the experimental data. At the same time, a small ConvNet as shown in Fig. 2 is constructed based on the Leaky Relu activation function in the experiment. In order to match the size of the image with the input size required by the model during the later test, the size of the image in the training set is adjusted to 32×32 pixels.

Usually the digital recognition model is used to identify three-channel images, and the images in the SVHN dataset used for testing are all three-channel. Therefore, before training the model, the images of the training set (MNIST dataset) need to be converted into three channels. During the experiment, the image transformed into three channels was randomly changed with 5 attributes, and the effect is shown in Fig. 6.

A reference metric is defined in the experiment—model's portability, which can better demonstrate the robustness of the model mentioned above. Perform five experiments to obtain and record the accuracy of the trained model tested on its own MNIST dataset and the accuracy after being tested on the SVHN dataset. Take the average of the data obtained from five experiments. Take the average of their five experimental data. Then divide the average value of the model after being tested on the SVHN dataset by the average value of the model after being tested by MNIST dataset.

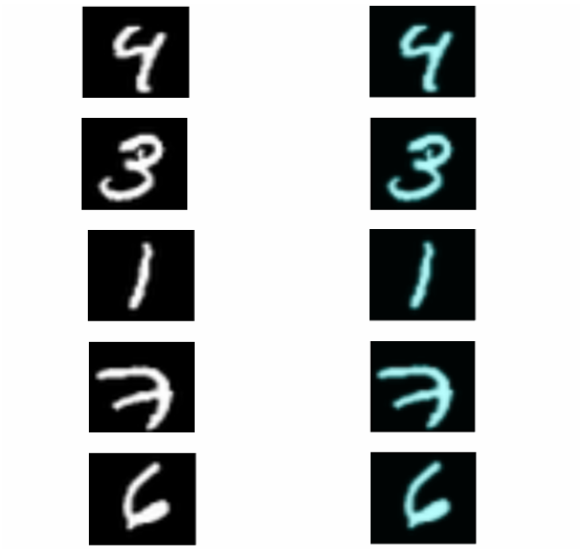


Fig. 6. The image on the left is the original image that has not been converted, and the image on the right is the image that has been converted to three channels and optimized for attribute combination

First of all, it is necessary to eliminate some untestable differences between the dataset used in this experiment and the dataset used by Volpi et al. The model proposed by Volpi et al. was tested with the same MNIST dataset and SVHN dataset as this work. The results of the experiment are in the 2nd and 3rd columns of Table 2.

Table 2. The results of the experiments

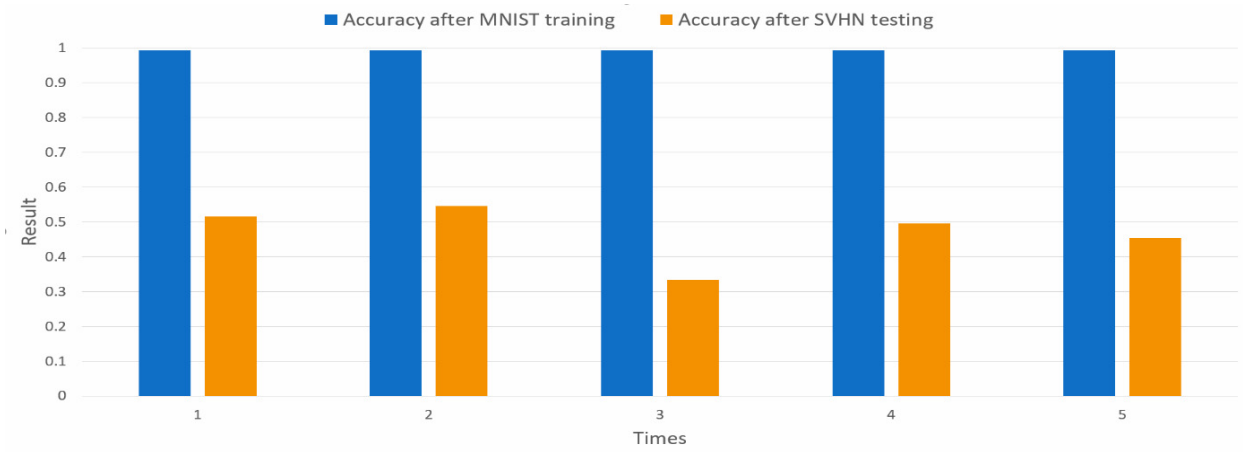
	The model proposed by Volpi et al.		The model with a small ConvNet		The model with ResNet	
	Accuracy after MNIST training	Accuracy after SVHN testing	Accuracy after MNIST training	Accuracy after SVHN testing	Accuracy after MNIST training	Accuracy after SVHN testing
1	0.9927	0.5164	0.9916	0.5431	0.9930	0.4919
2	0.9929	0.5459	0.9924	0.5951	0.9920	0.5103
3	0.9937	0.3346	0.9925	0.5572	0.9932	0.5093
4	0.9927	0.4952	0.9916	0.5388	0.9922	0.5698
5	0.9923	0.4548	0.9916	0.5609	0.9930	0.5199
Average value	0.9929	0.4694	0.9919	0.5590	0.9927	0.5202
Model's portability	0.4728		0.5635		0.5243	

Then according to the above method, the improved model with a small ConvNet (Fig. 2) was trained and tested many times, and the recorded test results are shown in the 4th and 5th columns of Table 2. And the intuitive results of the experiment can be obtained from Fig. 7.

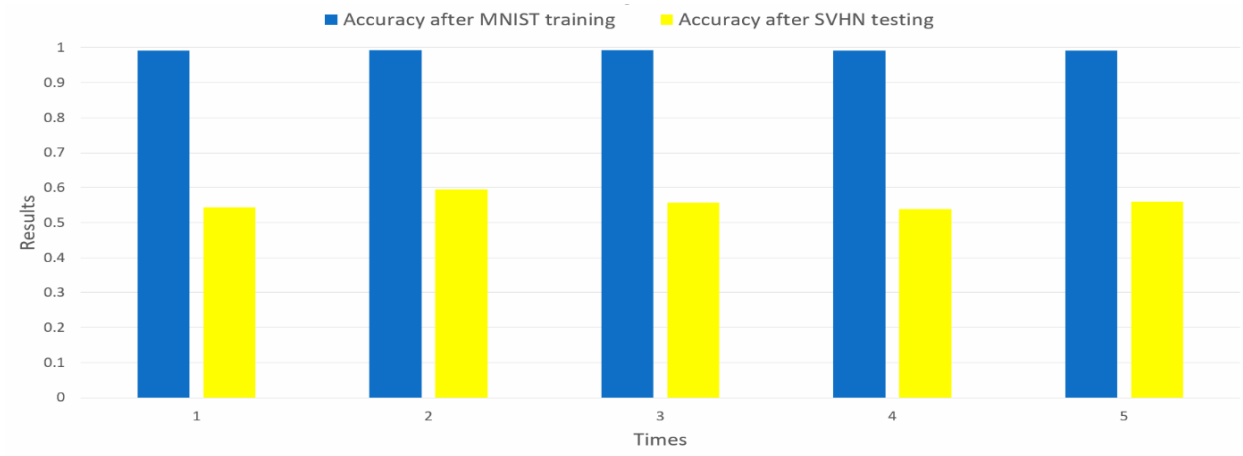
4.3 Research on Image Recognition Model that Based on ResNet

Combine the improved algorithm proposed in Section 3 with ResNet (Replace M in Equation 13 with ResNet) to train a new recognition model with high robustness that on different sets. The training dataset and testing dataset used in the process of verifying the feasibility experiment are the datasets used in Section 4.2. The size of the optimized combination is set to 5, the number of trainings is also 10^6 times, and the model's robustness against different sets is also measured according to the Model's portability defined in Section 4.2.

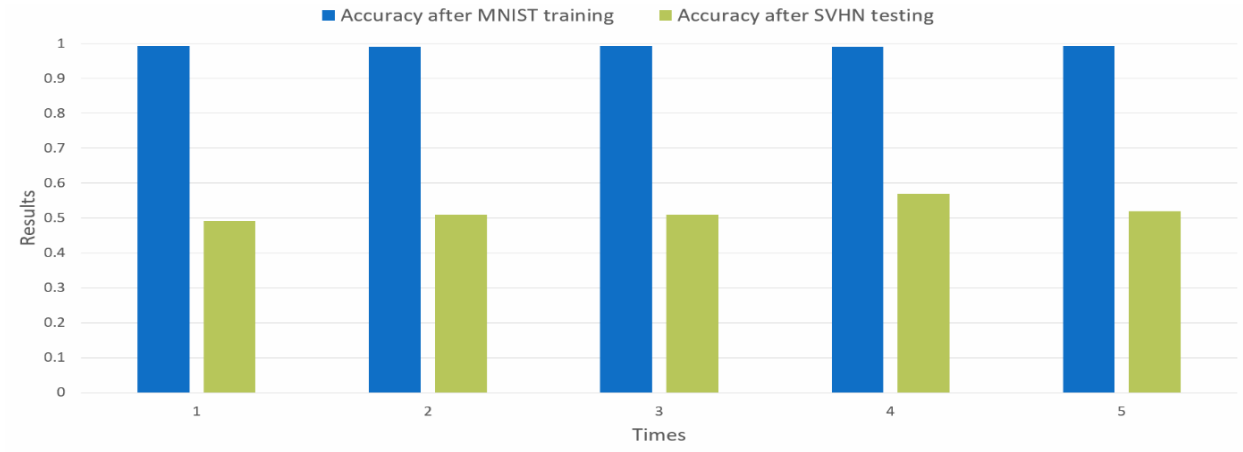
Follow the experimental steps in Section 4.2 to verify the feasibility of the newly constructed image recognition model. It is obvious that the effect of the new model is better than the effect of the model trained by Volpi et al. The specific experimental data is shown in the 6th and 7th columns of Table 2.



(a) is an intuitive diagram of the experimental results that the model proposed by Volpi et al. was tested



(b) represents the experimental results of the improved model with a small ConvNet



(c) shows the experimental results of the improved model based on ResNet

Fig. 7.

5 Results and Discussion

In order to more intuitively reflect the effects of the three models, Fig. 7 is drawn for illustration. Through the diagram, you can intuitively see the results of each model after being trained and the results after it has been tested. The comparison of the results that the model is tested can be observed more clearly.

Through Table 2 and the diagram (a) in Fig. 7 can get conclusion that the accuracy of the model proposed by Volpi et al. reaches a satisfactory state (average 0.99+) when the model after being trained with the same MNIST dataset. Then the same SVHN dataset was used to test the trained model, and the average test accuracy was close to 0.47. According to the definition in Section 4.2, the model's portability of the training model is 0.47+. This result is consistent with the experimental results of Volpi et al. Which indicating that there is no difference between the dataset used in this experiment and the dataset used by Volpi et al. Therefore, there is no effects of some unknown interference.

From Table 2 and the diagram (b) in Fig. 7, it can be seen that when the accuracy of the training model after training reaches a satisfactory state (average 0.99+), the training model is tested again with the testing set, and the average test accuracy is close to 0.56. Through the definition in Section 4.2, the model's portability of the training model is 0.56+. The performance is better than the reference model.

The information shown in Table 2 and the diagram (c) in Fig. 7 is that when the accuracy of the new model reaches a satisfactory state (average 0.99+) after being trained. The training model is tested with the testing set, and the average test accuracy is close to 0.54. Through the definition in Section 4.2, the model's portability of the training model based on ResNet is 0.52+.

In Fig. 8, the portability of the three models is very refreshingly compared. The Curve A coincides with curves C and E in Fig. 8. This phenomenon means that after the same number of trainings on the MNIST dataset, the accuracy of the new models has reached the level of the model proposed by Volpi et al. The overall results of the two curves B and D show that the accuracy of the improved recognition model with a small ConvNet is significantly higher than the recognition model proposed by Volpi et al. after being transplanted. Combined with the specific data in Table 2, it can be seen that the portability of the optimized recognition model with a small ConvNet is greatly improved (from 0.4728 to 0.5635) compared with the original model. The model has achieved the expected effect of improving the portability of the model.

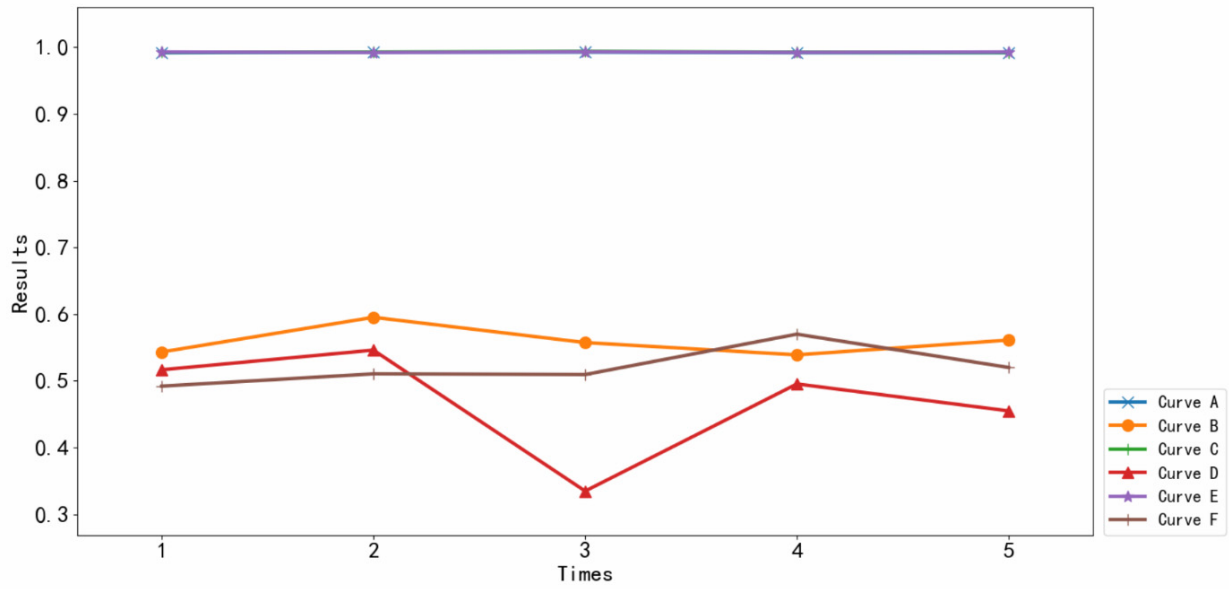


Fig. 8. Curve A is the accuracy of the model with a small ConvNet being trained on the MNIST dataset, and Curve B is the accuracy of the model with a small ConvNet being tested on the SVHN dataset; Curve C is the accuracy of the model proposed by Volpi et al. being trained on the MNIST dataset, and the Curve D is the accuracy of the model proposed by Volpi et al. being tested on the SVHN dataset; Curve E is the accuracy of the ResNet-based model being trained on the MNIST dataset, and the Curve F is the accuracy of the ResNet-based model being tested on the SVHN dataset

From the Curve F and Curve D in Fig. 8, it can be observed that the accuracy of the recognition model proposed by Volpi et al. is higher than the improved recognition model in the first and second experiment. However, in the first and second experiments, the degree of curve F lower than curve D is lower than the degree of curve D higher than curve F in the subsequent experiments. This phenomenon can prove that the portability of optimized recognition model with ResNet is also better than the original model. Combined with the data in Table 2, the portability performance is improved by 5%.

As can be seen from Curves B and F in Fig. 8, the portability of the two optimized models is not much different. And in the fourth experiment, the value of curve F is higher than that of curve B. But the portability of the new model with ResNet after being tested on the SVHN dataset is reduced by about 3.9% compared with the improved model with a small ConvNet (combined with the data in Table 5).

Fig. 9 can summarize the above discussion and get the results. The portability of the two optimized models in the experiment is higher than the original model. But there is a slight difference between the performance of the two optimized models. Fig. 9 can accurately show a conclusion that the models with robustness from high to low are the optimized model with a small ConvNet, the optimized model with ResNet, and the model proposed by Volpi et al. in turn.

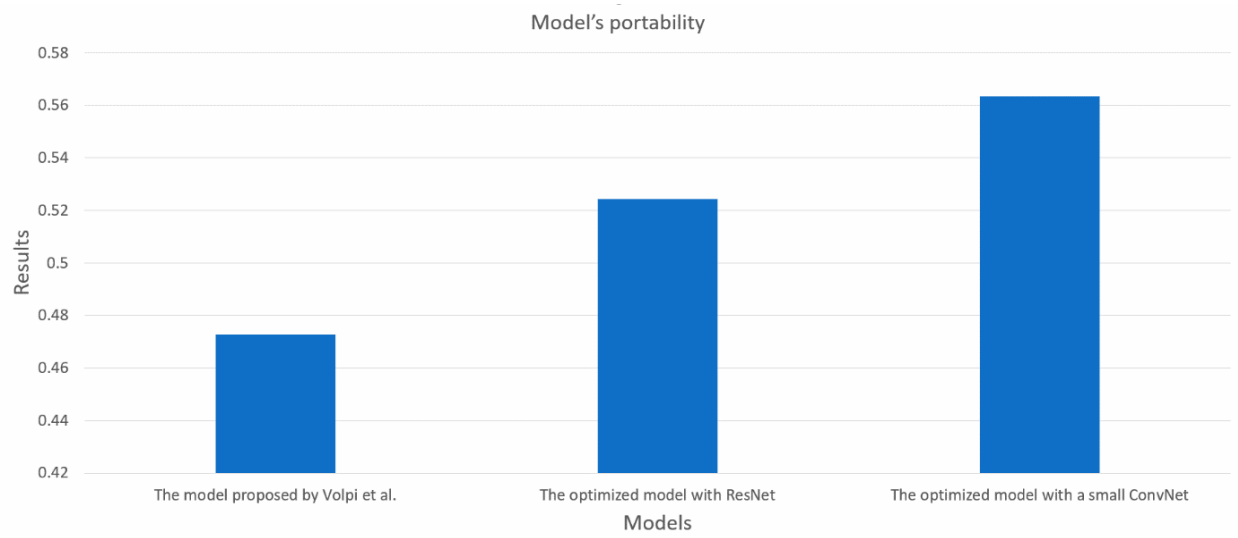


Fig. 9. The portability of the three models in the experiment

Although satisfactory results were achieved in this improvement work, some limitations were also exposed during the experiment. Using the E-S judgment method only reduces the complexity of some calculations, but the training process of the overall model is longer than that of the original model, which is caused by the increased complexity of attribute combination. At the same time, the image attributes used in the improved algorithm are limited, which cannot explain the accuracy of the model when other attributes are used.

6 Conclusions

The MinW-Rsearch algorithm and the E-S judgment method are proposed to improve the defects of the model proposed by Volpi et al. Two different improved models are constructed by combining two kinds of neural networks. The improved algorithm of attribute combination optimization was used to train a number recognition model on the MNIST dataset, and the model was transplanted to the SVHN dataset for number recognition. Experimental results show that after the improved recognition model is transplanted, it obtains higher accuracy than the model proposed by Volpi et al. This verifies that the improved algorithm can effectively improve the "distribution shift" phenomenon and make the image recognition model with more robustness against different sets. At the same time, this paper proposed a new image recognition model with higher robustness by combining ResNet neural network with improved algorithms. And the experimental results show that the model is feasible.

In future work, there are still some aspects that need further research: (1) exploring effective methods to improve model training efficiency and reduce model training time; (2) verifying whether the improved

method is suitable for other neural networks and other attributes.

References

[1] R. Volpi, V. Murino, Addressing Model Vulnerablity to Distributional Shifts Over Image Transformation Sets, in: Proc. 2019 IEEE International Conference on Computer Vision, 2019.

[2] A. Torralba, A. A. Efros. Unbiased look at dataset bias, in: Proc.2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011.

[3] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, S. Savarese, Generalizing to unseen domains via adversarial data augmentation, in: Proc. 2018 NIPS, 2018.

[4] A. Nguyen, J. Yosinski, J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proc. 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: Proc. 2014 ICLR, 2014.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, in: Proc. 2018 ICLR, 2018.

[7] H. Hosseini, R. Poovendran. Semantic Adversarial Examples, in: Proc.2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018.

[8] N. Akhtar, J. Liu, A. Mian, Defense Against Universal Adversarial Perturbations, in: Proc. 2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[9] S.M.M. Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal Adversarial Perturbations, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[10] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial Machine Learning at Scale, in: Proc. 2017 ICLR, 2017.

[11] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial Discriminative Domain Adaptation, in: Proc. 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2017.

[12] R. Volpi, P. Morerio, S. Savarese, V. Murino, Adversarial Feature Augmentation for Unsupervised Domain Adaptation, in: Proc. 2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[13] M.S. Long, Y. Cao, J.M. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, in: Proc. 2015 International Conference on Machine Learning, 2015.

[14] Y. Ganin, V. Lempitsky, Unsupervised Domain Adaptation by Backpropagation, in: Proc. 2015 International Conference on Machine Learning, 2015.

[15] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, Domain Generalization for Object Recognition with Multi-task Autoencoders, in: Proc.2015 IEEE International Conference on Computer Vision, 2015.

[16] D. Li, Y. Yang, Y.Z. Song, T. M. Hospedales, Deeper, Broader and Artier Domain Generalization, in: Proc.2017 IEEE International Conference on Computer Vision, 2017.

[17] M. Ghifary, D. Balduzzi, W. B. Kleijn, M.J. Zhang, Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization, IEEE Transactions on Pattern Analysis and Machine Intelligence 39(7)(2017) 1414-1430.

[18] N. Li, W. Li, D. Xu, J.F. Cai, An Exemplar-Based Multi-View Domain Generalization Framework for Visual Recognition, IEEE Transactions on Neural Networks and Learning Systems 29(2)(2018) 259-272.

[19] M. Mancini, S. R. Bul`o, B. Caputo, E. Ricci, Robust place categorization with deep domain generalization, IEEE Robotics and Automation Letters 3(3)(2018) 2093-2100.

[20] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proc. 1998 IEEE, 1998.

[21] Y. Netze, T. Wang, A. Coates, R. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: Proc. 2011 NIPS, 2011.