International Journal of Applied Mathematics in Control Engineering 6 (2023) 198-205

Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Dynamic Recognition of CAPTCHA Based on Convolutional Neural Network

Mengcheng Li[a], Lin Wang[a], Lingjie Meng[a], Jincai Chang[a,b*]

[a] *College of Science, North China University of Science and Technology, Tangshan Hebei 063210, China*

[b] *Hebei Key Laboratory of Data Science and Application, Hebei Tangshan 063210, China*

ARTICLE INFO

Article history:
Received 10 September 2023
Accepted 20 October 2023
Available online 26 October 2023

Keywords:
CNN
Captcha
Adaptive image processing
Dynamic identification

ABSTRACT

In the internet era, captcha is one of the fundamental and crucial safeguards for website data security, which is widely used in data submission scence. In this article, an end-to-end neural network model is proposed to address different types of images with various features and character lengths encountered in automated testing. The model combines adaptive image processing technology, maximum length label alignment method and feature post-processing method on the basis of convolutional neural network(CNN). The model achieves dynamic recognition of character captcha with an accuracy rate of around 82%. Compared with classic models such as VGG16 and ResNet18, this model demonstrates better accuracy and broader applicability, which has significant practical significance and application prospects in character captcha dynamic recognition.

1. Introduction

With the development of the Internet, image captchas have been widely used in scenarios such as websites, login, and data submission. This mixed image is composed of multiple character sets such as numbers, letters, and graphics, and has a high degree of randomness and diversity. In practical situations, the recognition rate of captchas by the human eye is only over 80%, while the recognition accuracy of automated programs is even lower. For website automation testing, different verification interfaces face different image features of captchas, different character lengths of captchas, and more complex structures and features.

With the rapid development of machine learning, domestic and foreign scholars have begun to use machine learning models to study captcha recognition. Kumar M et al. introduced different types of captcha based on text, audio and video in detail, analyzed the creation and cracking techniques of captcha, and discussed the availability and security of different types of captcha. This article can be a benchmark for in-depth study of captcha. Wu L et al. preprocessed the captcha character image with grayscale, binarization, denoising, and then used vertical projection segmentation method to segment and extract characters. Machine learning algorithms such as support vector machine algorithm and decision tree algorithm were used to recognize individual characters. Yu S et al. used DBSCAN density clustering method and K-means clustering method for data preprocessing of captcha images with noise points and adhesions, improving the accuracy of Fisher's character discrimination method. Wang X, Jin D. et al. preprocessed the image, used an end-to-end method to input the whole image into the neural network, and designed their own convolutional neural networks for captcha recognition research. Cui X et al. used an improved parallel cascaded convolutional Inception module to replace the convolutional layer and pooling layer, and used a global average pooling layer to replace the fully connected layer for captcha recognition research. Chen Z et al. used morphological corrosion and color filling methods to remove image noise, used an improved drip algorithm to segment characters, and used convolutional neural networks to study the recognition of individual characters. Li H et al. constructed a lightweight neural network that fusion dilated convolution and multi-scale sparse structure to extract image features and achieve end-to-end recognition of distorted and adhered character captcha images. Wang J et al. input the entire image to adjust the number of convolutional blocks for different feature datasets, construct different cross layer connected dense convolutional networks to identify captchas, and effectively alleviate the problem of gradient vanishing. Shi B et al. regarded image captcha recognition as a sequence recognition problem and proposed an end-to-end sequence recognition model based on convolutional recurrent neural networks, and successfully migrated the network model to music note recognition. Thobhani A et al. make a specific number of copies of the input captcha image to make it equal to the number of characters of the input captcha, and use additional binary images to attach to each copy to build a CNN model to identify the captcha, which reduces the storage space of the

* Corresponding author.
E-mail addresses: jincai@ncst.edu.cn (J. Chang)
Doi:

model and achieves end-to-end recognition. Long J et al. aiming at the problem of Tor dark point automatic monitoring, Long et al combined with the neural network model of CNN network, gated cycle unit GRU network and CTC loss, conducted a study with small samples and high accuracy. Li J et al. proposed a joint model of particle swarm optimization algorithm and convolutional neural network, using particle swarm optimization algorithm to find the optimal recognition model parameters for captcha recognition research. Wang Z et al. proposed a convolutional neural network model based on attention mechanism to address the issue of feature loss in image preprocessing. The attention mechanism was used to automatically learns different feature weights in the process of model training to improve the recognition accuracy of the model. Zhu H et al. aiming at the problem of information loss when Yolov7 extracts captcha features, proposed RDN-Yolov7 algorithm by drawing on the method of extracting multi-layer features in residual dense network to realize multi-feature fusion in deep network and improve the verification accuracy. Other scholars widely use deep learning methods for captcha recognition, but the recognition effect is not ideal for captcha type datasets with different features, uncertain character lengths, and multiple interferences. Conventional neural network models based on image segmentation are not suitable for multi feature datasets. Therefore, this article proposes a convolutional neural network model that combines adaptive image processing technology and label maximum length alignment method to achieve dynamic recognition of captcha characters.

2. Preliminary

2.1 Image processing method

Image scaling is a method of equal scale image size transformation, also known as up sampling or down sampling, the main purpose is to obtain a higher resolution image, the changed image and the original image have the same content, so the new region of pixels need to be interpolated, commonly used for the Nearest Interpolation, Bilinear Interpolation and Bicubic Interpolation.

Gray processing refers to the intensity of the color, RGB color components are all equal; the color value of each pixel on the grayscale image is also called grayscale, which refers to the color depth of the point in the black and white image, and the range is generally from 0-255, from white to black. Grayscale methods mainly include maximum method, average method and weighted average method.

$$F(i,j)=\max(R(i,j),G(i,j),B(i,j)) \tag{1}$$

$$F(i,j)=(R(i,j)+G(i,j)+B(i,j))/3 \tag{2}$$

$$F(i,j)=\omega_1 R(i,j)+\omega_2 G(i,j)+\omega_3 B(i,j)) \tag{3}$$

Where $F(i,j)$ is the gray value of the image at position (i,j), $R(i,j)$, $G(i,j)$, $B(i,j)$ is the gray component of the three channels, and ω is the weight.

Image binarization processing means that each pixel on the image has only two possible values or gray level states, and the gray value of any pixel in the image is 0 or 255.

$$\begin{gathered} if: src(i,j) >= threshold, \\ dst(i,j)=0 \\ else: dst(i,j)=255 \end{gathered} \tag{4}$$

Where $src(i,j)$ is the pixel value of the initial image at position (i,j), t is the $threshold$ value, and $dst(i,j)$ is the pixel value after binarization.

2.2 Convolutional neural network

Convolutional Neural networks (CNN) are network models based on multi-layer convolutional operations in deep learning, similar to conventional artificial neural network architectures. The basic structure of CNN is composed of input layer, convolution layer, activation layer, pooling layer and fully connected layer. It can accept multiple feature graphs in non-vector form as input. Meanwhile, because of the parameter sharing and interpretability characteristics of convolution operation, CNN has a wide range of applications in image processing.

The convolution layer is composed of multiple convolution kernels, which are a weight matrix, and each convolution kernels can effectively extract image local features. Convolution is the superposition of two functions, that is, the integration of the product of two functions. Image convolution is expressed as a two-dimensional function convolution discrete form:

$$\begin{aligned} g(x,y) &= f(x,y)*h(x,y) \\ &= \sum_{\xi}\sum_{\eta} f(\xi,\eta)h(x-\xi,y-\eta) \end{aligned} \tag{5}$$

Where $g(x,y)$ is called the convolution of the function $f(x,y)$ and $h(x,y)$. Where $*$ represents the convolution symbol. For a given pixel point (x,y), the first function is the pixel value $f(\xi,\eta)$ of the neighborhood matrix of that point, then the second function is the convolution kernel parameter matrix is $h(x-\xi,y-\eta)$ and needs to be summed over any possible location ξ,η.

The convolution process of two-dimensional images is shown in Fig. 1.

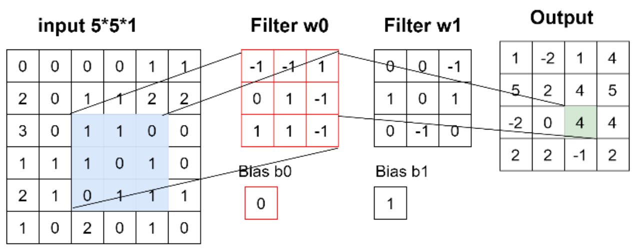

Fig. 1. Convolution process

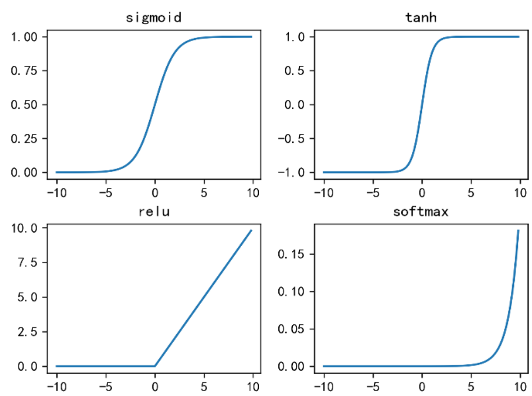

Fig. 2. Activation function image

The activation layer is a function of nonlinear mapping of the output results of the convolutional layer, which aims to help the network learn complex patterns in the data and enhance the representation and learning ability of the network. Activation functions ultimately determine whether the signal is transmitted and what content is transmitted to the next neuron. Common activation functions include Sigmoid, Tanh, ReLU, Softmax, etc.

The function image is shown in Fig. 2. The pooling layer down-samples the features extracted by the convolution layer to compress the data and parameters, compress the irrelevant information of the image, learn the edge and texture structure of the image, and prevent the model from overfitting.

Fig. 3. depicts the 2*2 filters and the maximum pooling with stride size 2.

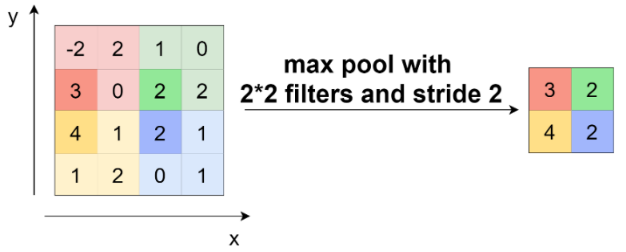

Fig. 3. The maximum pooling processes

The fully connected layer has weight connections to all neurons from the upper layer, usually passing values to the output layer at the tail. The selection of parameters of the fully connected layer comes from specific tasks such as classification and regression, which has an important impact on the final recognition accuracy.

3. Design of captcha recognition model

The function of captcha dynamic recognition model is to use convolutional neural network to preprocess images for multi-feature and uncertain length data, and then directly input images into the model for feature extraction, classification and training, so as to achieve accurate recognition, directly obtain captcha characters, and realize end-to-end captcha recognition model. The main process of captcha recognition model is divided into four parts: (1) Image preprocessing; (2) Network forward propagation; (3) Label loss calculation; (4) Backpropagation update parameters. The recognition model adjusts the weight parameters iteratively after continuous training, and completes the captcha recognition model when the loss value tends to be stable. The model flow is shown in Fig. 4.

The model in this article is improved from the perspective of how to save computing resources and how to increase the width of the network. A lightweight convolutional neural network (CNN8 in this article) with 6 convolutional layers and 2 fully connected layers in total of 8 layers is used to realize the dynamic recognition of captcha. Each layer of convolution is followed by a Batch Normalization function (BN) and a ReLU activation function, and the model uses Cross Entropy loss to update iteration parameters. The fourth and sixth convolutional layers are feature enhancement layers operated only by the convolutional, activation, and Dropout functions, which are designed to enhance the extracted features and enable the network to learn better features.

In the process of model training, the weak changes in parameters will be amplified with the deepening of the network, resulting in a butterfly effect. The deeper the neural network is, the more difficult it is to train and adjust parameters, so it is necessary to try different learning rates, initialize parameter methods or use optimized hyperparameter methods to help accelerate the model convergence. The addition of Batch Normalization function makes each layer of network less sensitive to the initial parameters, simplifies the parameter adjustment process, and makes the network learning more stable. BN makes the mean and variance of the input data of the network in the same range through normalization and linear transformation, destroys the original data distribution, removes the coupling between layers, and makes the model have stronger generalization effect. The data transformation after BN is added is as follows.

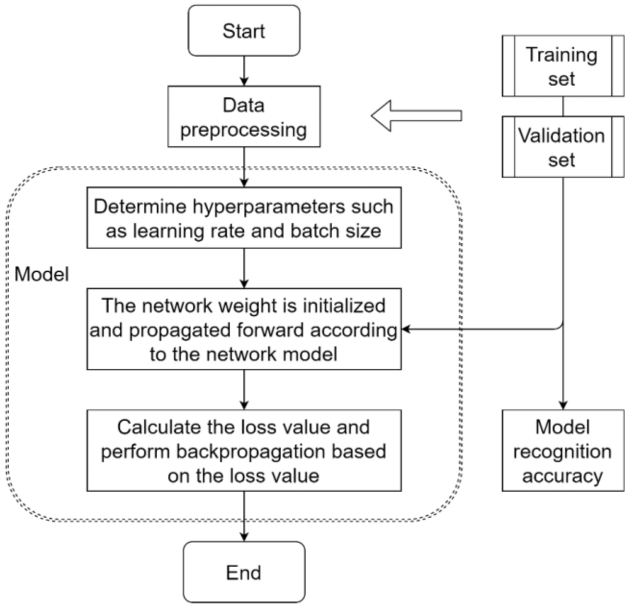

Fig. 4. Flow chart

$$
\begin{aligned}
BN(a\omega_u) &= \gamma \cdot \frac{a\omega_u - \mu_2}{\sqrt{\sigma_2^2}} + \beta = \gamma \cdot \frac{a\omega_u - a\mu_1}{\sqrt{a^2\sigma_1^2}} + \beta \\
&= \gamma \cdot \frac{\omega_u - \mu_1}{\sqrt{\sigma_1^2}} + \beta = BN(\omega_u)
\end{aligned} \tag{6}
$$

Where a is the scaling multiple, ω_u is the parameter value before scaling, the mean and variance before scaling are μ_1, σ_1^2 and the mean and variance after scaling are $\mu_2 = a\mu_1, \sigma_2^2 = a^2\sigma_1^2$.

In order to prevent overfitting in the process of network learning and enhance the generalization ability of the model, the addition of Dropout makes the neurons in the hidden layer be temporarily deleted with probability p during the forward propagation of the neural network, the input and output remain unchanged, and then the loss result is backpropagated through the modified network. The gradient descent method is carried out on the neurons that are not deleted to update the corresponding parameter weights and deviations ω, b, recover the deleted neurons and repeat the above process. The following is the forward propagation formula of the neural network after adding Dropout.

$$r_j^{(l)} \sim Bernoulli(p) \tag{7}$$

$$\tilde{y}^{(l)} = r^{(l)} \times y^{(l)} \tag{8}$$

$$z_i^{(l+1)} = \omega_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)} \tag{9}$$

$$u_i^{(l+1)} = f(z_i^{(l+1)}) \tag{10}$$

Where l is the number of layers in the neural network, y is the input vector, z is the output vector, r is the probability vector that follows the Bernoulli distribution, u is the activated output vector, and f is the activation function.

According to the image comparison of activation functions in Section 2.2, ReLU activation functions learn more quickly than other activation functions. Meanwhile, due to unilateral inhibition of negative value changing to 0, neurons in the model have sparse activation, which improves the recognition accuracy for the end-to-end recognition of image captcha containing background noise.

The ReLU activation function is calculated as.

$$f(x) = \max(0, x) \tag{11}$$

Cross entropy mainly measures the difference between two probability distributions, information entropy is the degree of uncertainty of information, relative entropy measures the difference between two probability distributions of the same random variable, and cross entropy is the sum of relative entropy and information entropy. The CNN8 model is a multi-classification task, which is suitable for using the cross-entropy loss function, which makes the predicted data distribution of the model learned from the training data better approximate the real data distribution.

The formula for calculating cross entropy as

$$H(p,q) = -\sum_{i=1}^{n} p(x_i)\log(q(x_i)) \tag{12}$$

Where $H(p,q)$ is the cross entropy, $p(x)$ is the true probability distribution, $q(x)$ is the model prediction probability distribution.

The cross-entropy loss function is calculated as

$$L = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_i \sum_{c=1}^{M} y_{ic} \log(p_{ic}) \tag{13}$$

Where M is the number of classes, N is the number of samples, y_{ic} is the symbolic function, if the true class of sample i is c, take 1, otherwise take 0, p_{ic} is the prediction probability that the observed sample belongs to class c

The structure of the CNN8 model constructed in this article is shown in Fig. 5.

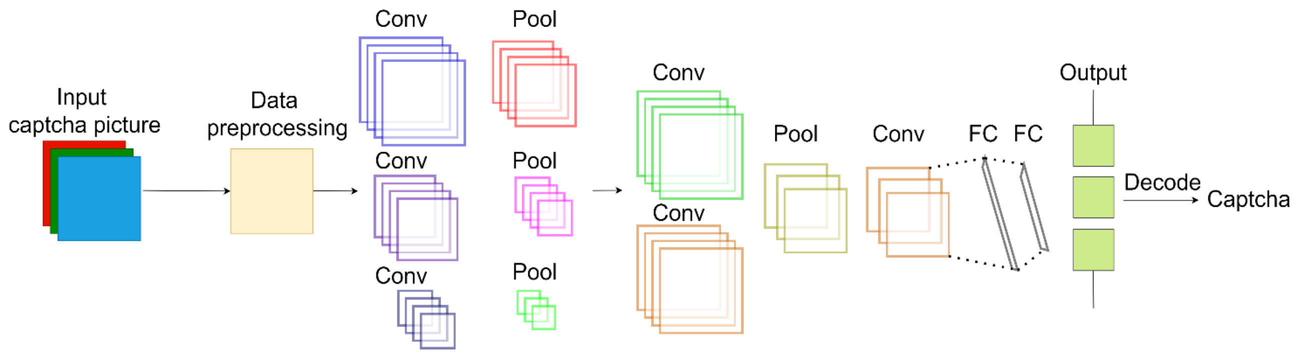

Fig. 5. Captcha recognition model network structure

The model adopts the end-to-end overall recognition method of the input image through used supervised learning to obtain the image captcha. One-Hot encoding is a common label encoding method. Each character label is mapped to a fixed-length vector whose length is equal to the size of the character set. In the vector, only the element in the position corresponding to the character label is 1, and the other elements in the position are 0, which is mainly used to convert the character into a numerical form that the neural network can handle. In the learning process of the neural network, the numerical form is required to participate in the calculation of parameters such as weight and bias, and participate in the calculation of the real result and the loss of the real label. The network output also needs to use One-Hot coding to represent the predicted result of each character.

The model uses the maximum length label alignment method to determine the output features of the neural network. As space characters have no features in the captcha image and do not affect character features, they only align the label length. Therefore, space like space characters are added to the character set to achieve dynamic recognition of variable length captchas. Calculate the maximum length len of the training label data, add space characters to align the label data based on the maximum length of the data. One-Hot encoding is used to encode 63 characters to obtain the coding value of the label, and the one-dimensional tensor is flattened and input into the CNN8 model as a label to automatically obtain and recognize the character feature rules.

The formula for the maximum length label alignment method is

$$output = len \times 63 \tag{14}$$

Therefore, the corresponding label encoding value in this article is len row 63 columns, rather than the combination of 63 characters, each row represents one character. The character one-hot encoding in part of the character captcha in this article is shown in Tab. 1.

Tab. 1. Example of One-Hot encoding of captcha

Captcha	One-Hot encoding
m5ym2	[[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]

In the process of model learning and training, the captcha image of 1×H×W is input into the model to generate the tensor of $output$ dimension. In the process of model classification, the decoder function is established by the feature post-processing method, that is, the $output$ dimension tensor with features output by the model is transformed into the len tensors of the 63-dimensional probability value features. For each tensor, the index of the maximum value of the 63-dimensional probability features is taken, and the corresponding letters and numbers are the recognition results of the model. The output tensor is matched with the label in the decoder function, and the loss value is calculated by gradient descent method.

With the letter ‘m’ , for instance, the output through the CNN8 model is [-1.1108,-0.2548,0.3827,-1.5825,0.7003,1.3648,-1.2066,3.2331,-2.7596,-1.1026,-1.6449,-0.4974,-0.2748,-1.7048, 0.2580, 3.0300, 1.0072, 0.6365, 0.4282, 0.3689, 0.9619, 0.2474, 4.7907, 2.0626, 0.5841, 3.3961, 0.5432, 0.1826, 1.8062, 0.7439, 0.3402, 0.3878, 0.2344, 0.6400, 1.7196, 0.5737, 0.7548, 1.1401, 0.8714, 2.6670, 0.6588, 0.3988, 0.8755, 0.1353, 1.1951, 1.500 ,1.4037 5, 2.1109, 2.3617, 1.5691, 1.4610, 0.2077, 0.4847, 1.4559, 2.6589, 0.0104, 0.1188, 0.5825, 0.3155, 0.8817, 3.7121, 0. 1540,-1.1728], first set everything less than 0 to 0 by activating the ReLU function, and then obtain the position 22 by the maximum index, and then the corresponding encoding is ‘m’ . When the loss between the predicted value and the label value is greater than the set loss, backpropagation training is carried out to obtain the optimal model parameter configuration.

The specific model parameters are shown in Tab. 2.

Tab. 2. Model parameters

Name	Parameter	Result
Conv1	3*3,64,stride1 pad1	64*100*60
MaxPool	2*2,stride2	64*50*30
Conv2	3*3,128,stride1 pad1	128*50*30
MaxPool	2*2,stride2	128*25*15
Conv3	3*3,256,stride1 pad1	256*25*15
MaxPool	2*2,stride2	256*12*7
Conv4	3*3,256,stride1 pad1	256*12*7
Conv5	3*3,512,stride1 pad1	512*12*7
MaxPool	2*2,stride2	512*6*3
Conv6	3*3,512,stride1 pad1	512*6*3
FC1	9216*4096	1*4096
FC2	4096*315	1*315

4. Dataset

4.1 Data introduction

This article collects captcha data of four different security features, including three real captcha data sets collected from Wikipedia, cnn.com and Yaoyoutong websites, and an ImageDraw module automatically generates captcha dataset. The source dataset contains three features: (1) Abstraction, with characters processed through rotation, translation, and distortion; (2) Adhesion, where some characters are connected together; (3) Noise, the captcha image contains noise points such as occluded lines. Tab. 3 shows examples of the features of the captcha.

Tab. 3. Captcha data feature display

Source	Sample	Feature
wikipedia.org		Character distortion and rotation;
cnn.com		Shielding line; Distance adjustment;
zyclh.com		Background noise; Different colors;
ImageDraw		Background noise; Shielding line;

The dataset contains a total of 18983 captchas. During the experiment, the samples were divided into a 7:3 ratio. 13852 captchas images were collected as training network parameters for the training set, and 5131 images were used as testing sets to test the dynamic recognition performance of the network.

4.2 Data preprocessing

Due to the differences in the size, format and features of the acquired data pictures, it is necessary to preprocess the original data sets. This article mainly carries out three steps of image scaling, gray level processing and binarization processing, and the image name is used as the label data. The input data should be adapted to the network input, using OpenCV to scale the image size to 100*60 for different images. The color features of the captcha are not very important for the recognition of the captcha. The gray processing of the image converts the color three-channel image into a single-channel gray level image, which can reduce the input data dimension and speed up the model training speed.

In order to better extract image captcha information and improve model universality, the local neighborhood block mean adaptive threshold method is used to determine the binarization threshold T at the pixel position based on the pixel value distribution of the neighboring blocks of the pixel. The formula is as follows:

$$T = R_{mean}(x, y) - c \times \sigma \tag{15}$$

$$R_{mean}(x, y) = \frac{1}{(2k+1)^2} \sum_{i=-k}^{k} \sum_{i=-k}^{k} P(x+i, y+j) \tag{16}$$

$$\sigma = \sqrt{\frac{1}{(2k+1)^2} \sum_{i=-k}^{k} \sum_{i=-k}^{k} [P(x+i, y+j) - R_{mean}(x, y)]^2} \tag{17}$$

Where (x, y) is the coordinates of the current pixel, P is the pixel value, and $R_{mean}(x, y)$ is the pixel mean within the $2k+1$ neighborhood centered on the point, k is an adjustable parameter, and c is the offset value adjustment constant, σ is the pixel standard deviation of the neighborhood. The grayscale captcha obtained after preprocessing is shown in Fig. 6.

Fig. 6. Binary image

5. Experimental

The experimental program was written in Python, and the open source machine learning framework Pytorch was used to build the convolutional neural network. The device was verified to be configured with Windows11 operating system, 16GB memory and NVIDIA GeForce GTX3060 GPU.

Since captchas are used for automated testing, attention is paid to recognition accuracy and efficiency. After referring to relevant literature, the recognition accuracy and model recognition time are used to evaluate the captcha recognition effect of the model.

The formula for model recognition accuracy is as follows:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \tag{18}$$

Where TP represents the true case, TN represents the true negative case, FP represents the false positive case, and FN represents the false negative case. The greater the Acc value, the higher the recognition accuracy and the better the model.

The formula for model recognition time is as follows:

$$Time = end - start \tag{19}$$

Where $start$ represents the start time of prediction, and end represents the end time of prediction. The larger the $Time$ value, the longer the recognition process takes and the lower the efficiency of model recognition.

5.1 CNN8 recognition and verification

During the training process, the DataLoader function was used to read the training dataset. The batch size of each batch to be trained was 64, and the order of reading data was disordered. The Adam optimizer is used to update the parameters in the training process, and the cross-entropy loss function is used to calculate the predicted loss value and backpropagation. Set the initial learning rate at 0.001, the epoch iterations at 70-120 times, save the model once for each iteration, and put the test dataset into the model for accuracy test. The loss results in the training dataset and test dataset are shown in Fig. 7, and the model accuracy rate is shown in Fig. 8.

After model training, the highest prediction accuracy of the network model training set is 93.97%, and the accuracy of the test dataset is 82.73%. Part of the captcha recognition results are shown in Fig. 9.

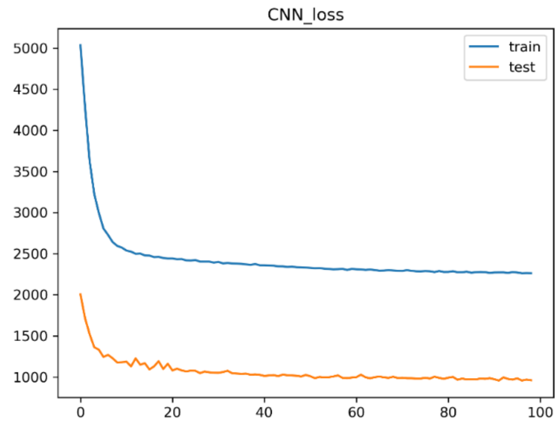

Fig. 7. Loss function descent curve

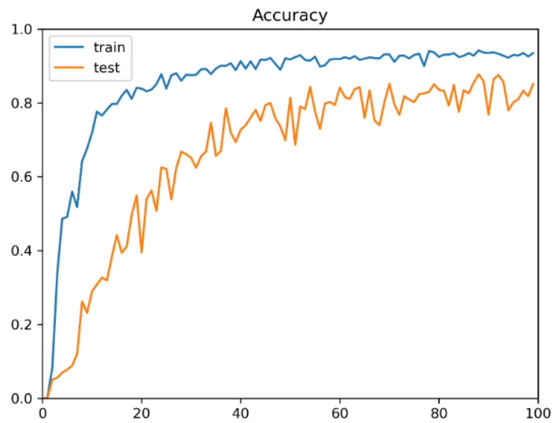

Fig. 8. Recognition accuracy rise curve

5.2 Ablation study

In order to verify the effects of these key designs on the model, such as feature enhancement layer, BN function, ReLU activation function and cross entropy loss function. This section conducts ablation experiments on the CNN8 model on the basis of experiment 5.1 these factors are removed respectively to conduct comparative experiments on model accuracy.

For the network structure without feature enhancement layer, the feature enhancement convolutional layer of the fourth and second layer was removed, and the rest of the structure was unchanged. The network structure without BN function, BN function removes from the network, the structure design of the same; The ReLU activation function is replaced by Sigmoid activation function to maintain the original network structure; The cross-entropy loss function is replaced by MSE loss function to maintain the original network structure. The ablation experiment comparison of CNN8's captcha recognition accuracy is shown in Fig. 10.

The experimental results show that the feature enhancement layer and BN function can improve the accuracy of CNN8 model by about 30%. The accuracy of the model trained by MSE loss function is higher than that of other models, but not as high as that of CNN8 model under cross entropy loss function. The accuracy of the model replaced by Sigmoid activation function is 0%. Sigmoid function is not suitable for the CNN8 captcha code recognition model proposed in this article, which has good accuracy and generalization ability.

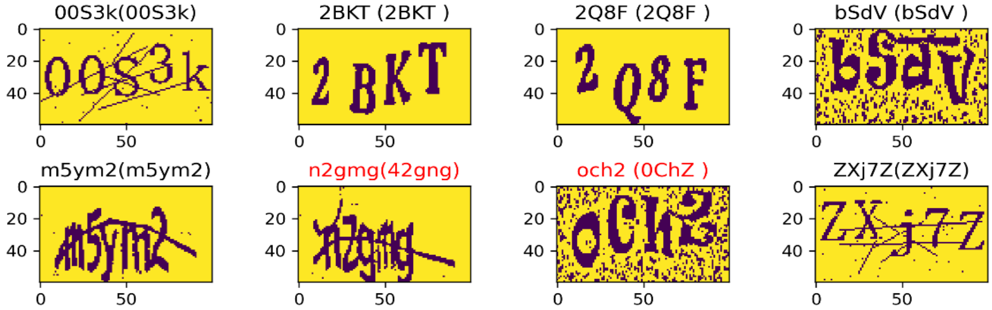

Fig. 9. Captcha recognition results

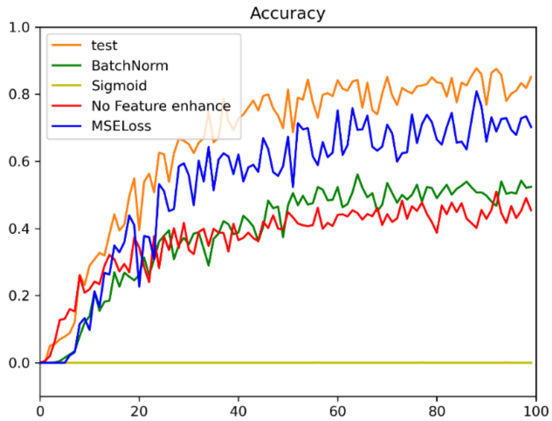

Fig. 10. Ablation study

5.3 Contrast experiment

To further evaluate the proposed captcha dynamic recognition model in the article, classic VGG16 and ResNet18 classification models were cited for comparative experiments. The three models used the same dataset, learning parameters, and loss function for model training. Quantitative analysis was conducted using indicators such as the number of model parameters, validation accuracy, model recognition time, and model training time. The results showed that CNN8 model had higher accuracy and efficiency.

The comparison results of the models are shown in Tab. 4. ResNet18 has the least network parameters, and the network model in the article has significant advantages over the two classic network models in terms of time and accuracy.

Tab. 4. Analysis of contrast experimental results

Model	Parameter	Acc of train	Acc of test	Time	Batch time
VGG16	135,552,936	92.39%	57.34%	0.169s	47.19s
ResNet18	**11,851,944**	84.87%	57.10%	0.119s	29.12s
Ours	39,054,528	**93.97%**	**82.73%**	**0.079s**	**26.65s**

The network hierarchy used in this model is concise, with fewer parameters compared to the other two classification models, resulting in shorter optimization time and higher efficiency. In addition, the model also adopts a feature post-processing algorithm, which reduces the number of neurons recognized in the overall classification from 63^5 to 315, reduces the number of neurons, reduces network parameters, and improves the accuracy of the network.

The experimental results show that the dynamic recognition model of captcha presented in this article has a good recognition performance when processing data sets composed of different features and characters. In addition, the model also shows good generalization ability for some unknown captchas with obvious features.

6. Summary

The article based on the test automation in many features, uncertain length captcha data, from end-to-end recognition based on adaptive image processing perspective, combined with the feature of label the maximum alignment and post-processing methods, puts forward a captcha dynamic identification model based on CNN. Experimental results on 5131 test dataset show that the model has a high accuracy of 82.73%. It was evaluated through ablation studies and model contrast experiments. The experimental results showed that the key choices in the proposed model, such as feature enhanced convolution, BN function and activation function, significantly improved the accuracy of the model. Compared with the classical VGG16 and ResNet18 classification models, CNN8 model showed advantages in accuracy and efficiency. The CNN8 model proposed in this article successfully realizes dynamic identification of different types of captchas, which provides a new research idea for automated testing and captcha security testing.

Acknowledgements

This work was supported by The National Natural Science Foundation of China (No. 61702184).

References

Zhang C. Research on Recognition Algorithms of Text-based Captcha based on Convolutional Neural Network[D]. China University of Mining and Technology, 2022.

Kumar M, **dal M K, Kumar M. A systematic survey on CAPTCHA recognition: types, creation and breaking techniques[J]. Archives of Computational Methods in Engineering, 2022, 29(2): 1107-1136.

Wu L, Zhang X, Liang X. Research on Automatic Identification and Security Risk of System Login Verification Code based on Machine Learning[J]. Chinese journal of health informatics & management, 2020, 17(04):523-527.

Yu S, Tian X, Wang J. CAPTCHA recognition based on multivariate statistical analysis and machine learning[J]. Journal of Shandong University of Technology(Natural Science Edition), 2019, 33(01):60-64.

Lia L, Caia K, Lia S, etal. Zero-Watermarking Algorithm Based on Image Normalization and 2D-LPEWT[J]. International Journal of Applied Mathematics in Control Engineering, 2022(5):114-120.

Wang X, Wang B. Lightweight CAPTCHA recognition method based on convolutional neural network[J]. New Technology & New Products of China, 2021, No.444(14):24-26.

Jin D, Liu T, Liu T. Verification Code Recognition Based on Python and CNN[J]. Software Engineering, 2019, 22(06):1-4.

Cui X, Bai P, Zhang C, et.al. An end-to-end CAPTCHA recognition method based on deep convolutional neural network[J]. Journal of Shandong University of Science and Technology (Natural Science), 2020, 39(02):111-117.

Li S, Chen L. Research on Object Detection Algorithm Based on Deep Learning[J]. International Journal of Applied Mathematics in Control Engineering, 2018(2):127-135.

Chen Z, Huang X, Qin Z. Text-based CAPTCHA recognition based on image processing and convolutional neural network[J]. Cyberspace Security, 2020, 11(08):75-80.

Li H, Cheng H. Lightweight Neural Network Design for Complex Verification Code Recognition Task[J]. Computer Systems & Applications, 2021, 30(04):247-252.

Wang J, Qin J, Xiang X, et al. CAPTCHA Recognition Based on Deep Convolutional Neural Network[J]. Mathematical Biosciences and Engineering, 2019, 16(5):5851-5861

Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(11): 2298-2304.

Thobhani A, Gao M, Hawbani A, et al. Captcha recognition using deep learning with attached binary images[J]. Electronics, 2020, 9(9): 1522.

Long J, Wang T, Xue Z. Quick Identification Of Verification Codes And Data Collection On Key Tor Dark Web Sites[J]. Computer Applications and Software, 2022, 39(07):315-321

Li J, Wang Z. Research on Verification Code Recognition Algorithm Based on PSO-CNN[J]. Computer Technology and Development, 2022, 32(09):51-55.

Wang Z, Li Y, Shang Z, et al. Research on CAPTCHA recognition based on attention – based convolutional neural network[J]. Journal of Southwest Minzu University (Natural Science Edition), 2023, 49(03):303-311.

Zhu H. The Application of an lmproved Yolov7 Algorithm in Verification Code Recognition[J]. Journal of Putian University,2023,(05):57-61.

Zhang Y, Lv B, etal. Research on Garbage Image Classification Method Based on ResNet-34 Deep Network[J]. International Journal of Applied Mathematics in Control Engineering, 2021(4):9-17.

Chen X, Zhang R. Target Recognition and Detection Based on Convolutional Neural Network Deep Learning[J]. International Journal of Applied Mathematics in Control Engineering, 2021(4):107-112.

Shida Caia, Liangliang Sun, Mantong Zhao, Wang Yue, Xi Li, Jiayu Peng. Prediction Of Milling Cutter Wear Based on ASO-BP Neural Network[J]. International Journal of Applied Mathematics in Control Engineering, 2022, (5): 109-113.

Zhang J. Research on Character Verification Code Recognition Based on Deep Learning[D]. Qingdao University of Science & Technology,2023.

Mengcheng Li was born in Baoding, Hebei Province, China in 2000. He is graduate student of Graduate School of North China University of Science and Technology, studying Big data technology and engineering, main research direction is 3D reconstruction based on image features.

Lin Wang was born in Zhangjiakou, Hebei Province, China in 1998. He is graduate student of Graduate School of North China University of Science and Technology, studying Mathematics, main research direction includes computational mathematics, numerical simulation, and three-dimensional simulation.

Lingjie Meng was born in Hengshui, Hebei Province, China in 2000. She is graduate student of Graduate School of North China University of Science and Technology, major in mathematics, main research interests are simulation, spline approximation.

Jincai Chang received his B.Sc. degree in 1996 from Ocean University of China, received his M.Sc. degree in 2005 from Yanshan University, received his Ph.D. degree in 2008 from Dalian University of technology, now he is Professor in North China University of Science and technology. His main research interests include theories and methods in mathematical modelling and scientific computation, numerical approximation and computational geometry, etc.