IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Multi-feature Learning by Joint Training for Handwritten Formula Symbol Recognition

Dingbang Fang[1], Chenhao Zhang[2]

[1]School of Information Science and Engineering, Huaqiao University, Xiamen, Fujian 361021, China
[2]School of Computer, Central China Normal University, Wuhan, Hubei 430079, China

Corresponding author: Chenhao Zhang (e-mail: zhangchenhao@mails.ccnu.edu.cn).

This work was supported in part by the Graduate Student Scientific Research Innovation Project Foundation of Huaqiao University.

ABSTRACT Given the similarity of handwritten formula symbols and various handwriting styles, this paper proposes a squeeze-extracted multi-feature convolution neural network (SE-MCNN) to improve the recognition rate of handwritten formula symbols. The system proposed in this paper integrates the eight-directional feature of the original sequence in the convolutional layer, which significantly compensates for the lost dynamic trajectory information in the handwritten formula symbol. Meanwhile, the joint loss is constructed to improve the discriminability of features in the way of supervised learning, which enlarges the inter-class difference and decreases inner-class similarity. The standard mathematical formula symbol library provided by the Competition Organization on Recognition of Online Handwritten Mathematical Expression (CROHME) is used to verify the effectiveness of the proposed algorithm. Experiments show that the proposed SE-MCNN approach outperforms the state-of-the-art methods even at the condition of without using the data augmentation.

INDEX TERMS Artificial neural network, multi-feature, joint training; iscriminative feature

I. INTRODUCTION

With the wide applications of handwriting input in electronic devices, the research of handwriting recognition technology has attracted attentions both from the academia and the industry [1]-[6]. Although optical character recognition (OCR) has high skills in handling the tasks related to handwriting recognition, it is difficult to apply to handwrite mathematical formulas with a strong dependency on contextual information [3][4]. On the one hand, a single text line is one-dimensional and discrete, but mathematical formulas have a complex two-dimensional layout (superscript, subscript, nesting) in the space. On the other hand, the symbol recognition is not a simple problem because it contains rich characters (numbers, Latin and Greek letters, operators, relation symbols), multiple fonts (regular, bold, italic), and different sizes of the font. Handwriting mathematical formula recognition can divide into two steps [5][6], including symbol recognition and structural analysis. The structural analysis aims to analyze the intrinsic relationship between symbols in the previous stage. In other words, symbol recognition affects the subsequent structural analysis. Therefore, this paper focuses on designing a method to improve the performance of isolated handwritten formula symbol recognition (IHFSR).

At present, according to the form of handwriting, IHFSR can be divided into two modes [11]-[20]: the online and offline mode. The online mode mainly stores as sequential data in a certain way through the interactive interface; the offline mode mainly adopts the grayscale image collected by the camera. the IHFSR is a branch of significant research on handwriting recognition, which promotes the development of handwriting recognition to a certain extent. Intuitively, the convolutional neural network (CNN) [7][27]-[29] can directly classify the scanned images corresponding to the handwritten trajectory, but this method ignores the trajectory information of the time domain. Thus, this paper combines these two modes to improve the performance of IHFSR. This paper summarizes the three major problems of IHFSR: (1) Compared with only 10 categories of MNIST datasets, the datasets have hundreds

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

of formula symbols, including numbers, Latin letters, and arithmetic symbols, such as " ∇ ", " Ω ". (2) There are a number of similar symbols in the formula symbol library, resulting in lots of confusion among categories, such as "O" and "o", "S" and "s". (3) For the same type of symbol, handwriting style for formula symbols varies hugely from person to person. Parts of the handwritten mathematical formula symbol are randomly selected from CROHME and shown in Figure 1.

To address these problems, this paper proposes a multi-feature learning network model to improve the accuracy of IHFSR. The main novelties and contributions of the proposed method originate from the following aspects: (1) In the online mode, the eight-directional features are designed for the first time to compensate for the missed trajectory information in the image formation process, and the eight-directional Gabor feature is extracted in the offline mode to obtain the change of multi-directional gradient. (2) The proposed network can adaptively learn the relevant features and improve the network robustness. (3) For the network classification, the joint loss is used to train SE-MCNN to penalize the categories of misclassification and increase the inter-class difference, which effectively improves the generalization performance of the network.

The rest of the paper is organized as follows: Section II is about the related work about IHFSR. Section III introduces data preprocessing and directional feature extraction. Section IV is the principle of SE-MCNN model. In Section V, the experimental results are applied to verify the superiorities of the proposed method. Section VI summarizes the proposed method.

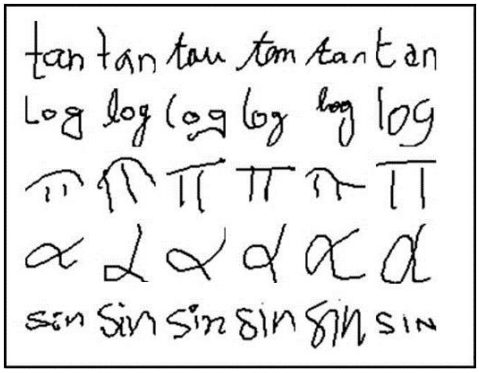


FIGURE 1. A variety of handwritten style fonts

II. RELATED WORK

In the early study, it is difficult to verify the generalization performance of the algorithm by hand-crafted data limited to partial symbols [8][9]. Currently, the mainstream algorithm divides the IFHSR into three categories according to the feature selection, namely online, offline, and hybrid modes. For online symbol recognition, some structural features and non-structural features are commonly used in the online mode, such as stroke pairs, context shapes, and geometric features. The main classifiers are HMM [10], Adaboost [11], Markov Random Fields [12] and generalized classifier of generalized learning vector quantization (GLVQ) [13]. The advantage of online feature extraction is that it can capture dynamic trajectory information, but the use of classifiers as traditional machine learning algorithms cannot guarantee feature redundancy.

At the same time, for offline features, the conventional approach is to convert online data into offline images, and CNN extracts higher-level semantic features from images [14][15][16]. The literature [14] takes Simple Linear Iterative Clustering (SLIC) to segment mathematical formulas, and the segmented symbols are classified by the pre-trained SqueezeNet model. Ramadhan et al. [15] designed a CNN model trained on the images transformed from the CHROME 2014 training set, but this method did not significantly improve the recognition rate. Dong et al. [16] carefully designed a new network CNN framework called VGG-HMS. Unlike the literature [14][15], the framework is characterized by increasing the depth of the network and making full use of the resources of the network, which achieves the highest recognition rate in the CROHME2014 dataset, and the recognition rate in the CROHME2016 dataset is second only to Myscript [20].

Regarding the features of fusion, most authors combine the online features and offline features from the original data to improve the accuracy of IFHSR [17]-[20], where shallow features, such as Crossings features, 2D fuzzy histogram features, polar histogram features, et al., extracted by manual are derived from the sequence data. It is worth noting that the Myscript team [20] wins the CROHME 2016 competition, using a multi-layer perceptron (MLP) combined with recurrent neural network (RNN) to classify the histogram features and curvature features in the trajectory.

From the above research, it can be seen that the key to IHFSR is feature extraction and model construction. The method of offline feature extraction still has an extensive exploration space due to the information is lost in the temporal domain. As compensate for the lack of spatial trajectory information in offline mode, this paper introduces directional features to enhance the robustness of the model effectively. Meanwhile, the joint training makes the convolutional neural network extract more discriminative features to represent handwritten formula symbol images, and various loss functions are conceived in IHFSR based on the SE-MCNN model. The overall flowchart is shown in Figure 2. In the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

following sections, we would describe how each module works.

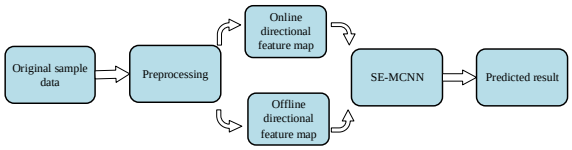


FIGURE 2. overall flowchart

III. PREPROCESSING AND FEATURE EXTRACTION

3.1 Preprocessing

The CROHME adopts a file in InkML's format that belongs to an Ink Markup Language [21]. The user makes a trace on the graphical interface either by hand or using an electronic pen, and the terminal converts the handwritten trajectory data into a specific format. As shown in Figure 3, each id represents a set of trajectories, namely "id=11" and "id=12". Besides, the trajectory is made up of a large number of sample points. Online handwritten data easily interfered with by external factors, proper pre-processing [22] can alleviate these noises and effectively improve the quality of the dataset. The main steps of preprocessing are as follows: (1) Adding data points to the sample, the user may have handwritten jitter that causes some data points to be lost. (2) Cubic spline interpolations. (3) Smooth filtering that the original data points are replaced by the mean of the adjacent two points. (4) Removing duplicate data points to avoid the redundancy of information. Given a consistent aspect ratio of the trajectory, the size of the character sample is specified to a range of 32×32. In order to process image efficiently, the background pixels labelled as 0, and the foreground pixel is normalized to 0 and 1.

```
<ink xmlns="http://www.w3.org/2003/InkML">
<traceFormat>
<channel name="X" type="decimal"/>
<channel name="Y" type="decimal"/>
</traceFormat><annotation type="truth"></annotation>
<annotation type="UI">2013_IVC_CROHME_F124_E416_7293</annotation>
<trace id="11">
694.0 115.0, 709.0 121.0, 714.0 123.0, 719.0 125.0, 725.0 126.0, 731.0 128.0,
737.0 130.0, 743.0 132.0, 750.0 133.0, 757.0 134.0, 764.0 136.0, 771.0 137.0,
777.0 139.0, 784.0 140.0, 789.0 141.0, 794.0 143.0, 799.0 144.0, 803.0 145.0,
806.0 146.0, 809.0 147.0, 810.0 148.0, 812.0 149.0, 812.0 151.0, 813.0 152.0,
812.0 154.0, 810.0 156.0, 809.0 158.0, 805.0 161.0, 802.0 164.0, 799.0 168.0,
794.0 172.0, 788.0 176.0, 782.0 181.0, 776.0 186.0, 769.0 190.0, 761.0 196.0,
753.0 200.0, 746.0 205.0, 738.0 210.0, 730.0 215.0, 724.0 219.0, 717.0 223.0,
711.0 227.0, 705.0 231.0, 701.0 233.0, 697.0 236.0, 694.0 238.0, 691.0 240.0,
690.0 241.0, 688.0 243.0
</trace>
<trace id="12">
844.0 180.0, 837.0 187.0, 833.0 190.0, 827.0 194.0, 822.0 199.0, 815.0 203.0,
809.0 208.0, 801.0 214.0, 794.0 220.0, 787.0 225.0, 780.0 230.0, 773.0 236.0,
767.0 241.0, 761.0 247.0, 755.0 251.0, 749.0 256.0, 744.0 260.0, 740.0 265.0,
737.0 268.0, 734.0 271.0, 731.0 274.0, 728.0 276.0, 727.0 277.0, 726.0 278.0
</trace>
<traceGroup>
        <traceGroup xml:id="0">
                <annotation type="truth"></annotation>
                <traceView traceDataRef="11"/>
                <traceView traceDataRef="12"/>
        </traceGroup>
</traceGroup>
</ink>
```

FIGURE 3. An example of InkML file

3.2 Multi-feature layer acquisition

According to the previous studies [23][24], the usage of domain-specific knowledge can improve the recognition rate of the Chinese character to some extent. For methods of offline IHFSR, the process of converting online data into images would lose spatial and temporal information, which might degrade the recognition performance. Oppositely, an effective method is to transform the online trajectory into a feature map by decomposing the direction of the trajectory to develop useful information as much as possible, as shown in Figure 4.

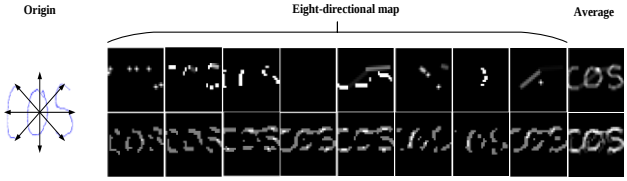


FIGURE 4. the left side is the original track. The first layer and the second layer are respectively an online 8-directional feature map and an offline 8-directional feature map. The last column is the corresponding average feature map.

It is shown that the average feature map retains the outline of the original image in Figure 4, indicating that the eight-directional feature map preserves the structural information of the original image. Feature extraction mainly decomposes gradients (for offline images) and partial strokes (from 0° to 360°) from two perspectives, online and offline. We can get a variety of feature maps that represent the original gradient and the direction of the stroke.

3.2.1 ONLINE EIGHT-DIRECTIONAL FEATURE MAP EXTRACTION

The InkML provides data in a sequence that has a large amount of dynamic trajectory information. To fully utilize online information, this paper employs the method proposed in the literature [25] to extract the directional features of each group of trajectories. The direction vector is projected onto the set direction axis to form a corresponding feature map. The eight directions are divided into two groups, namely $d_j^{\,1}=\{D_1,D_3,D_5,D_7\}$ and $d_j^{\,2}=\{D_2,D_4,D_6,D_8\}$.

$$d_j^{\,1}=\frac{|d_x-d_y|}{s},\qquad d_j^{\,2}=\frac{\sqrt{2}\min(d_x,d_y)}{s} \tag{1}$$

where d_x and d_y respectively represent the difference between the horizontal and vertical coordinates of the adjacent two points, s is the distance between two adjacent points, and each directional feature map represents direction and gradient information of local stroke.

3.2.2 OFFLINE EIGHT-DIRECTIONAL FEATURE MAP EXTRACTION

For off-line images, the Gabor filter [26] is used to obtain the edge gradient of the image, and the gradient is decomposed in a specific predefined direction. Unlike the un-structured features such as directional features are extracted from an online pattern by discarding temporal and structural information, we can also extract offline eight-directional feature map. It's corresponding detail that is described as:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

$$\psi = \mu(x\cos\vartheta_r + y\sin\vartheta_r) \tag{2}$$

$$G_1(x,y) = \frac{\mu^2}{\sigma^2}\exp[-\frac{\mu^2(x^2+y^2)}{2\sigma^2}] \tag{3}$$

$$G(x,y;\mu,\vartheta_r) = G_1(x,y)[\exp(j\psi) - \exp(-\frac{\sigma^2}{2})] \tag{4}$$

$$F(x,y;\mu,\vartheta_r) = I(x,y) * G(x,y;\mu,\vartheta_r) \tag{5}$$

where $\mu = \frac{2\pi}{l}$, $\vartheta_r = r\frac{\pi}{R}$, $r = 0,...,R-1$, $\sigma = \pi$, the parameter ϑ_r is the direction apart and $G(x,y;\mu,\vartheta_r)$ denotes Gabor filters. $F(x,y;\mu,\vartheta_r)$ represents the feature map after filter processing and $I(x,y)$ stands for original image. In the experiment, $R = 8$, the gradient in eight directions is $0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ$, respectively, and the wavelength of the Gabor l is set to $4\sqrt{2}$.

IV. NETWORK MODELS

The convolutional neural network has become an essential tool in the field of computer vision [27]-[30], which these frameworks are also applied to IHFSR. The convolutional neural network is more suitable for extracting the deeper semantic features. There are many handwriting styles and each style has enormous symbols with a similar appearance in the CROHME library, which brings serious challenges to handwritten formula symbol recognition. Therefore, the strategy of increasing the depth of the network as an essential aspect of CNN architecture design, and changing the relationship of channels is employed in this paper to improve the capability of learning the discriminative features of CNN, as in [27][31]. we proposed a squeeze and excitation fusion method, which uses the "Squeeze and Excitation net (SEnet)"[31] structure to make full use of the interdependence between convolution channels to enhance the representation ability of multi-feature. Specifically, the squeeze layer adopts global average pooling (GAP) instead of average pooling in the convolution network to avoid the loss of information on spatial location relationships caused by average pooling. The extraction layer is composed of a fully connected layer with a gating unit (FC1, FC2), in which the number of activation units C/ℓ and C, respectively, where ℓ is the attenuation factor, and the nonlinear activation functions are Relu and Sigmoid, respectively. The following formula summarizes the above operations:

$$\hat{X} = Sigmoid(FC_1(Relu(FC_2(GAP(X))))) \tag{6}$$

The simplification of equation (6) can be described as $\hat{X} = \alpha X$, where α is the compression factor, and the parameters are adaptively adjusted during backpropagation.

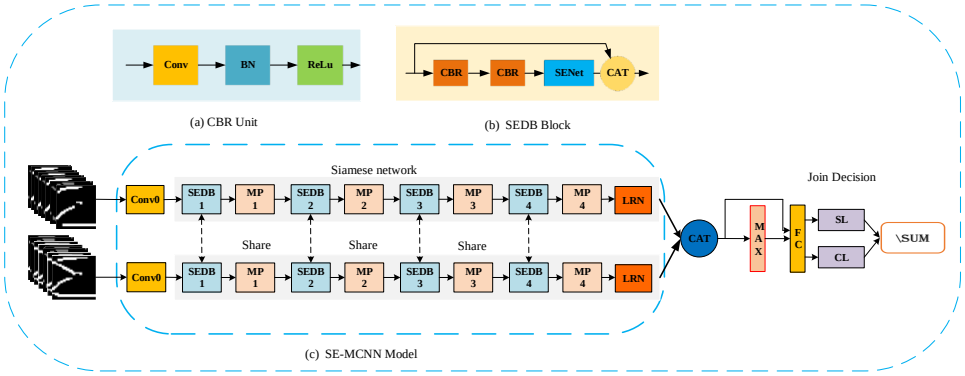


FIGURE 5. CBR Unit (a), SEB Block (b), and SE-MCNN Model (c)

4.1 SE-MCNN model framework

Considering the advantages of SEnet and dense connections, the SE-MCNN is designed and the architecture is shown in Figure 5. First, in the design process of a convolutional layer, this paper employs convolution (Conv)-batch normalization (BN) [32]-non-linear activation function (Relu), called as CBR unit, as shown in Figure 5(a). BN enables the convolutional eigenvalues to be evenly distributed in the linear region of the nonlinear activation function, and thus improves the operational efficiency of the network. Second, the two CBR units and Senet are packaged in turn to build the in-depth feature learning branch. At the same time, the shallow and deep features are connected through self-joining. A block with the characteristics of SEnet and dense connections is called SEDB, as shown in Figure 5(b). The SEDB can extract shallow fine-grained features and in-depth semantic features, and robust feature maps are obtained through concatenating (CAT) connection following by each SEDB module. Third, a convolution layer (Conv0), the SEDBs (SEDB1-SEDB4), maximum pooling layers (MP1-MP4) and local response normalization (LRN) layer [33] are sequentially stacked to construct a branch in the leading network, as shown in Figure 5(c). Besides, the online eight-directional feature map and the offline eight-directional feature map are used as inputs to the respective branches, which is referred to as Siamese architecture between the branch networks. Finally, the fused feature in the form of CAT is fed to the Maxout layer [34] to improve the robustness of the model, which makes the decision layer more discriminative.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

4.2 Objective function

It is hard for the softmax function to distinguish handwritten symbols with similar handwriting style. Therefore, the softmax function is not suitable for this case and more strong loss function should be designed to construct the identification objective function. Specifically, softmax loss (SL) of the identification objective function is defined as follows:

$$L_{SL} = \frac{1}{K}[\sum_{k=1}^{K}\sum_{c=1}^{C}\delta(y_c^{\,k})\log\frac{e^{W_c^T X(k)}}{\sum_{p=1}^{C} e^{W_p^T X^{(k)}}}] + \frac{1}{2}\alpha\|W\|_2^2 \qquad (7)$$

where $W = [W_1, W_2, W_3 .... W_C]$ is the set of weight vectors for all label categories and the total number of labels is C. $\delta(y_c^{\,k})$ is an indication function, denoted as the label of the sample in class, and all labels are encoded by one-hot. The α is a penalty factor in reducing the over-fitting of the model. $X^{(k)}$ express any of the k-th samples in a total of K samples. Moreover, the central loss (CL) function enlarges inter-class differences and reduces the variation of features within the inner-class. A backpropagation algorithm updates the distance between the depth feature and the corresponding class, which is formulated as follows:

$$L_{CL} = \frac{1}{2}\sum_{k=1}^{K}\left\|x_k - c_{y_k}\right\|_2^2 \qquad (8)$$

where c_k is the in-depth feature of the k-th category, the joint training mechanism is used in SE-MCNN to learn discriminative features in a self-supervised manner. The softmax loss (SL) function and the central loss function are employed to train the neural network jointly [35]. The final objective function of Joint Loss (JL) is defined as follows:

$$L_{JL} = L_{SL} + \lambda L_{CL} \qquad (9)$$

The hyperparameter λ is the weight ratio to balance the identification objective functions. The input of the SE-MCNN consists of a grayscale image and an eight-directional feature map, described as a three-dimensional tensor of size $h \times w \times c$. h and w are the length and width of the input, and D is the number of channels (this paper is 9). The convolutional channels of the first layer are 32, the number of channels of SEDB1-SEDB4 is 32, 64, 128, 512, and the stride of the SEDB1-SEDB4 and MP1-MP4 is configured as 1 and 2, respectively. All convolutional layers in the network are designed with a convolution kernel of 3×3 small size. Intuitively, a larger convolution kernel, such as a 7×7 or 11×11 convolution kernel, can be used to obtain a larger receptive field. The receptive field of two 3×3 convolution kernels is equivalent to the receptive field of a 5×5 convolution kernel. The receptive field of three 3×3 convolution kernels is equivalent to a 7×7 convolution kernel receptive field. The advantage of using a small convolution kernel repeatedly is that it can improve the learning capacity on semantic features.

V. Experiment analysis

5.1 Introduction to the formula library

The CROHME has successfully hosted five years, CROHME2011-CROHME2016[13][20][36] and has become the main event in ICDAR [37], which has been also one of the organizations providing the mathematical formula symbol library. The University of Nantes in France collected the dataset of CROHME, by marking the Latex and drawing formula symbols from formulas. There are totals of 101 classes of symbols, but the distribution among all classes is an imbalance. In order to verify the generalization performance of SE-MCNN, CROHME2013 is used as the experimental validation set to estimate the models and adjust the parameters of the network during training. Besides, the CROHME2014 and CROHME2016 are used as experimental test sets. The dataset distribution of CROHME is shown in Table I:

TABLE I
DISTRIBUTION OF CROHME EXPERIMENTAL DATASETS

DATA SETS	DATASET CATEGORY	IMAGE SIZE	QUANTITY
TRAINS	CROHME2016 TRAIN	32×32	85802
VALIDATIONS	CROHME2013 TEST	32×32	6082
TESTS	CROHME2016 TEST	32×32	10019
	CROHME2014 TEST	32×32	10061

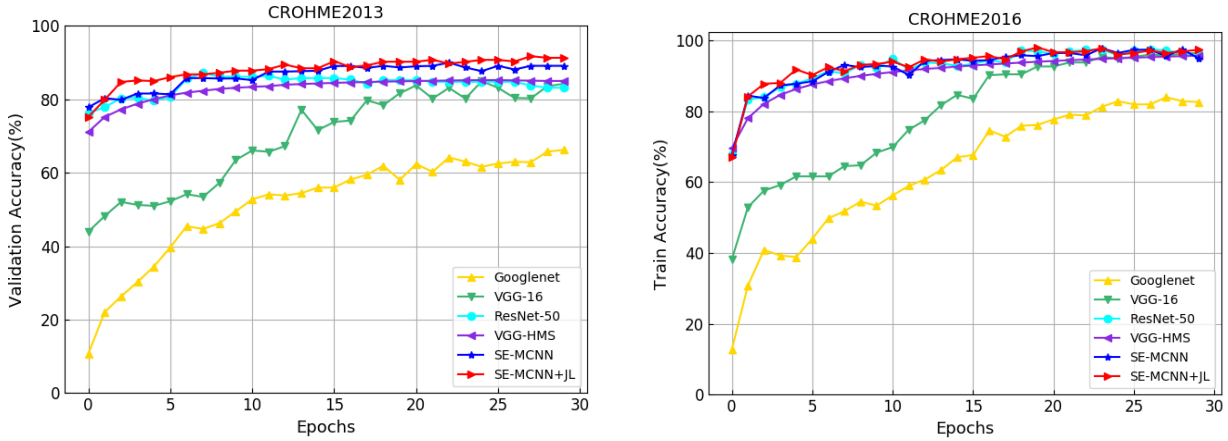
5.2 Implementation Details

The algorithm proposed in this paper is implemented in the deep learning framework of TensorFlow. The hardware configuration is 12-thread Corei7-8700CPU@3.2GHZ 16G RAM and GTX1060 graphics card. The weight parameters are initialized by the MSRA [38], and biases parameters are initialized to 0 for each layer in the network. The advantage of such a scheme is that the variance of the weights can be randomly adjusted to avoid gradient vanish. Each mini-batch consists of 64 images, and each image is randomly selected from the entire dataset. According to the empirical value, the regularization coefficient in equation (7) is set to 0.005. Considering that there is a weight between the two target loss functions, in the formula (9) is set to 0.5 to balance the relationship between the objective functions. The model was trained by Adam [39], and all data is iterated 30 times. The learning rates start with 0.1 and the minimum learning rates are 0.0001, decreased by 1% every 6000 steps until the objective function tends to converge in the training stage.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

5.3 Comparison with existing algorithms

To demonstrate superiority, the proposed method is compared with some existing deeper networks and state-of-the-art IHFSR methods, as shown in Figure 6 and Table II. On the one hand, the frameworks of VGG-16 [27], ResNet-50 [28], GoogLeNet [29], VGG-HMS [16] are compared with SE-MCNN proposed in this paper. In Figure 6(a), all of our models (SE-MCNN to SE-MCNN+JL) can easily outperform previous benchmarks in the CROHME2013 test set. In Figure 6(b), for the GoogLeNet model [29] and the VGG-16 model [27], the trend of the curve appears to have large fluctuations, and the other models gradually begin to converge. The ResNet-50[28] model is too complicated to avoid the redundancy of the network structure. The SE-MCNN adopts a compact structure and its self-joint to avoid the problem of gradient disappearance effectively.



(a). Accuracy of the CROHME2013 dataset **(b). Accuracy in the CROHME2016 dataset**

FIGURE 6. Comparison with the current state-of-the-art model framework during the training and testing phases

TABLE II
COMPARISON OF THE PROPOSED ALGORITHM WITH THE CURRENT DOMINANT ALGORITHM

System	CROHME 2014 test accuracy	CROHME 2016 test accuracy	Feature used
Ours	**92.44%**	**92.96%**	Online + Offline
VGG-HMS [16]	91.82%	92.42%	Offline
MLP+RNN [13][20]	91.04%	92.81%	Online + Offline
CNN+LSTM [19]	91.28%	92.27%	Online + Offline
RNN [17][20]	91.24%	-	Online +Offline
SVM-RBF [18][20]	88.66%	88.85%	Online + Offline
CNN [15]	87.72%	-	Offline
AdaBoost [11]	85.00%	-	Online
GLVQ [13]	84.31%	-	Online

On the other hand, our proposed algorithm is compared with some current mainstream algorithms, as shown in Table II. According to the feature extraction method, the algorithms can be divided into three categories, namely offline features, online features, and fusion features. As revealed in Table II, the model proposed in this paper achieves the highest accuracy for IHFSR. From the perspective of model and feature selection, CNN [15] is superior to traditional machine learning methods (SVM-RBF [18], AdaBoost [11], GLVQ [13]), which is higher by 2% to 3%. However, it is lower than the method of combining online, such as MLP+RNN [13][20], CNN+LSTM [19], RNN [17][20]. Each model utilizes RNN to capture long-term context dependencies of sequential data, but its temporal dynamic behavior with the delay brings serious shortcoming. The recognition rate of VGG-HMS [16] is higher than that of the above method, and the accuracy in CROHME2016 is 0.39% lower than that of MLP+RNN [13], but it lacks dynamic trajectory features. In view of the above problems, this paper integrates the eight-direction feature map into the convolutional layer and adopts the mechanism of joint training. Compared with VGG-HMS [16], the increase is 0.62% and 0.54%, respectively, which results verified the effectiveness of the proposed algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

5.4 Effectiveness of multi-features and joint training

In this subsection, we comprehensively evaluate the performance of offline features, multi-features, and JL on the CROHME 2013, 2014, and 2016 datasets, in the following three aspects. Firstly, under the condition that the SE-MCNN parameters are consistent, the process that directly extracts the offline features in the image by a single CNN is called the baseline. Second, Multi-feature (MF) contains online and offline direction information, which is integrated with CNN to train the model. Lastly, the mechanism of JL introduced in the Softmax layer has a feedback effect on the backpropagation. Moreover, the top-N accuracy rate is used to measure the performance of these three different training methods. The formula of Top-N is expressed as follows:

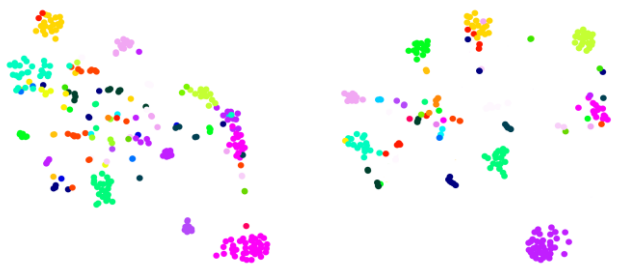
$$\text{Top-}N = \frac{\sum_{i=1}^{M}(\{1|\text{argmax}(y_i^*) \in \underset{N}{argmax}(f(x_i))\})}{M} \quad (10)$$

where y_i^* is the true value, $f(x_i)$ is the predicted value, M is the total number of samples. N is the top N highest score in the ground truth. One can see from Table III that the proposed method (i.e., Baseline+MF+JL) achieves the best performance about Top-1 accuracy in the test sets of CROHME2013, CROHME2014, and CROHME2016. However, the recognition rates in CROHME2013 test sets are much lower than CROHME2016 test sets and CROHME2014 test sets, because CROHME2013 test sets have a higher proportion of confusing symbols than the other data sets. It can be obviously found that the proposed MF method significantly improves the baseline model. We can obverse that the accuracy rates have an improvement of 2% to 3% in the Top1, while Top3 and Top5 are also increased by 1% to 2%. The interval of Top-3 and Top-5 recognition rates are increased by 0.7% to 1%. The gaps between the recognition results of other methods (Baseline +MF and Baseline +MF+JL) are relatively small in terms of corresponding data. It is revealed that the multi-feature can improve the robustness of the model. Likewise, we can conclude that JL has the capability of improving the discriminability.

We compare the offline and online eight-direction feature maps as the prior knowledge embedded in the input layer of the baseline model, and the method of MF fusion. The recognition rates obtained by the corresponding methods are 90.26%, 91.32%, and 92.96% in the CROHME2016, as shown in Table IV. This clearly shows that these methods are improved based on the baseline, and the MF fusion method further improves the performance of the model. In addition, Figure 7 shows the t-SNE [40] visualization of the baseline model and the proposed SE-MCNN, which represent the offline and multi-feature distributions after training, respectively. It can be found that the method of using multi-feature training has better separability. The design of the algorithm proposed in this paper takes into account the lack of dynamic trajectory information of offline features, which effectively reduces the problem of similar symbol recognition.

TABLE IV
RECOGNITION RATES OF THE BASELINE MODEL USING DIFFERENT DIRECTIONAL FEATURES

METHODS	TOP-1 (%)	TOP-3 (%)	TOP-5 (%)
BASELINE	89.56	95.40	97.40
BASELINE + OFFLINE	90.26	96.56	98.67
BASELINE + ONLINE	91.32	96.87	98.79
BASELINE + MF	92.96	98.36	99.77



(a) Offline feature (b) Multi-feature

FIGURE 7. The visualization of feature distribution via t-SNE resulted from offline feature and multi-feature on CROHME test dataset, where different colors represent different kinds of handwritten symbols, 300 samples were randomly selected from the test data set as an example.

TABLE III
TOP-N ACCURACY IN CROHME2013-CROHME2016

DATASET	CROHME 2013 TEST			CROHME2014 TEST			CROHME2016 TEST		
METHODS	TOP-(1,3,5) ACCURACIES (%)			TOP-(1,3,5) ACCURACIES (%)			TOP-(1,3,5) ACCURACIES (%)		
	TOP-1	TOP-3	TOP-5	TOP-1	TOP-3	TOP-5	TOP-1	TOP-3	TOP-5
BASELINE	87.59	96.50	98.20	89.75	95.40	97.40	89.56	95.40	97.40
BASELINE+MF	88.42	97.80	98.40	91.25	95.70	96.59	91.11	97.18	98.36
BASELINE+MF+JL	**89.55**	97.90	98.80	**92.44**	96.02	99.85	**92.96**	98.36	99.77

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access

5.5 Category analysis

Although our algorithm achieves excellent global accuracy, there are a large number of confusing symbols in the CROHME data set. Therefore, this paper retests the average accuracy of each class. The accuracy of some symbols randomly selects in the CROHME2016 test set, as shown in Table V. One can see that the accuracy of the SE-MCNN with JL (Baseline+MF+JL) is significantly improved. For confusing symbols such as '\times-X, C-c, [-\{-(' in Table V, it is difficult for the baseline model to distinguish such similar symbols since the lack of online trajectory information. However, the eight-directional feature map is integrated into the convolutional layer of baseline, which effectively reduces the misclassification rate. Meanwhile, it is shown that the JL is introduced in the Sotfmax layer to further improve the robustness and discriminability of the model.

TABLE V
LOCAL ACCURACY OF RANDOMLY SELECTED PARTIAL SYMBOL CATEGORIES

METHOD / CLASS	BASELINE+MF+JL	BASELINE+MF	BASELINE
\times	0.7083	0.5302	0.0278
\ldots	0.775	0.725	0.775
6	0.9556	0.8444	0.8222
[	0.7419	0.7419	0.9032
\lim	1	0.8571	0.8571
G	0.8556	0.7556	0.6667
o	0.323	0.22	0.099
\exists	0.85	0.75	0.65
\{	0.4286	0.2857	0.2857
X	0.5333	0.3212	0.001
+	0.9887	0.9486	0.9791
.	0.7438	0.7286	0.7143
\mu	0.9714	0.9127	0.8515
(	0.9498	0.8989	0.7978
n	0.9625	0.9205	0.7045
c	0.9382	0.9034	0.9176
a	0.9355	0.6316	0.7895
\div	0.6842	0.5333	0.6333
g	0.8667	0.7838	0.8108
j	0.973	0.8923	0.8308
C	0.61	0.3648	0.2395
\sum	0.9654	0.9082	0.8545
\alpha	1	0.8947	0.7895
\div	0.6842	0.5333	0.4333

5.6 Discussion

The algorithm proposed in this paper achieves a new benchmark for IHFSR without data augmentation and model ensemble, but there are still many potentialities for improvement. The CROHME dataset is mainly presented in the form of a long tail, as shown in Figure 8. The number of symbols '-' has 7940, and some special characters, such as ' ∃ ', have only four. In addition, the number of most symbol types is few in the datasets, which makes it difficult to learn in convolutional networks. This issue can be solved by utilizing augmentation in most classification cases so that the number of each category keeps a balanced state. For example, the literature [16][18] augments online data and offline data separately, but this measure cannot avoid poorly effects on similar symbols. Currently, few-shot learning [41] is also an important research topic, which will be used in IHFSR to improve accuracy in future work. Meanwhile, the previous work is applied to the field of mathematical formula recognition to achieve new progress.

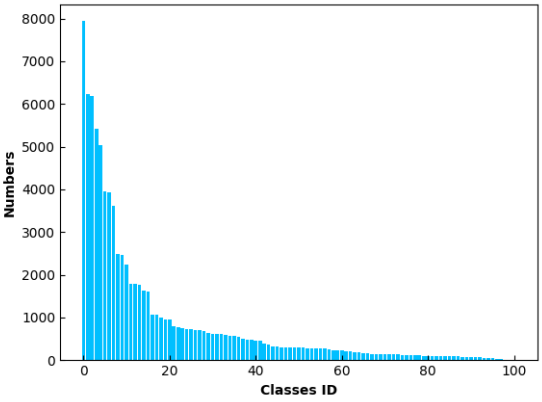


FIGURE 8. 101 symbol class distribution map

VI. CONCLUSION

This paper proposes a novel convolutional neural network, SE-MCNN, to improve the accuracy of handwritten formula symbol recognition. Merged with the original images, the online and offline eight-directional feature maps were input into the Siamese network. Meanwhile, the joint loss function was utilized to promote model performance in a self-supervised learning manner. The effectiveness of the proposed algorithm was tested with CROHME2014 and CROHME2016. It's noted that the proposed algorithm is superior to the existing handwritten formula symbol recognition algorithm. An attractive and important future job is to use robust few-shot learning to train models to improve robustness.

REFERENCES

[1] K V Govindan, P A Shivaprasad, "Character recognition—a review," *Pattern Recognit.*, vol. 23, no. 7, pp. 671-683, 1997.

[2] A J Sánchez, V Romero, H A Toselli, et al., "A set of benchmarks for Handwritten Text Recognition on historical documents," *Pattern Recogniti*, vol. 94, pp. 122-134, Oct. 2019.

[3] Z, Jianshu, D Jun, et al., "Track, Attend, and Parse (TAP): An End-to-End Framework for Online Handwritten Mathematical Expression Recognition." *IEEE Trans. Mul*., vol. 21, no. 1, pp. 221-223, June. 2018.

[4] G Shan, H Wang, W Liang, et al., "Robust Encoder-Decoder Learning Framework towards Offline Handwritten Mathematical Expression Recognition Based on Multi-Scale Deep Neural Network," [online]. Available: https://arxiv.xilesou.top/abs/1902.05376. 2019.

[5] K F Chan, D Y Yeung, "Mathematical expression recognition: a survey", *Int. J. Doc. Anal Recognit*., vol. 3, no. 1, pp. 3-15, Aug. 2000.

[6] D Blostein, A Grbavec, "Recognition of mathematical notation," *Handbook of character recognition and document image analysis.* 1997, pp. 557-582.

[7] Y LeCun, L Bottou, Y Bengio, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*., vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[8] B Q Vuong, Y He, S C Hui, "Towards a web-based progressive handwriting recognition environment for mathematical problem solving," *Expert. Syst. Appl*., vol. 37, no. 1, pp. 886-893, Jan. 2010

[9] A Belaid, J P Hatton, "A syntactic approach for handwritten mathematical formula recognition," *IEEE Trans. Pattern. Anal*., vol. 6, no. 1, pp. 105-111, Jan. 1984.

[10] L Hu, R Zanibbi, "HMM-Based Recognition of Online Handwritten Mathematical Symbols Using Segmental K-Means Initialization and a Modified Pen-Up/Down Feature," in *Proc. Int. Conf. Document. Anal. Recognit*., pp. 457–462, Nov. 2011.

[11] L Hu, Z Richard, "Segmenting handwritten math symbols using AdaBoost and multi-scale shape context features," in *Proc. 12th Int. Conf. Document Anal. Recognit*., pp. 1180-1184, Aug. 2013.

[12] D F Julca-Aguilar, T S N Hirata, "Symbol detection in online handwritten graphics using Faster R-CNN," In *Proc. 13th IAPR Int. Workshop. Document. Anal. Systems* (DAS)., pp.151-156. Jun. 2018.

[13] H Mouchere, C Viard-Gaudin, R Zanibbi, et al., "Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014)," In *Proc. 14th Int. Conf. Frontiers Handwriting Recognit*., pp. 791-7, Sep. 2014.

[14] A Nazemi, et al., "Offline handwritten mathematical symbol recognition utilizing deep learning." [online]. Available: https://arxiv.org/abs/1910.07395. 2019.

[15] I Ramadhan, B Purnama, S Al Faraby, "Convolutional neural networks applied to handwritten mathematical symbols classification," in *Proc. IEEE Int. Conf. ICoICT*., pp. 1-4, May 2016.

[16] L Dong, H Liu, "Recognition of Offline Handwritten Mathematical Symbols Using Convolutional Neural Networks," in *Proc. 6th Int. Conf. Image. Graph.* (ICIG), pp. 149-161. Dec. 2017.

[17] F Alvaro, J A Sanchez, J M Benedi, "Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks," in *Proc. IEEE 22th, Int. Conf. Pattern Recognit*., pp. 2944–2949. Aug. 2014,

[18] K Davila, S Ludi, R Zanibbi, "Using Off-Line Features and Synthetic Data for On-Line Handwritten Math Symbol Recognition," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit*., pp. 323-328, Sep. 2014.

[19] D H Nguyen, A Le Duc, M Nakagawa, "Recognition of Online Handwritten Math Symbols Using Deep Neural Networks," *IEICE T. Inf. Syst*., vol. 99, no. 12, pp. 3110-3118. Dec. 2016.

[20] H Mouchère, C Viard-Gaudin, R Zanibbi, et al., "ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions," In *Proc. 15th Int. Conf. Frontiers Handwriting Recognit*., pp. 279-284, Oct. 2016.

[21] Ink markup language. http://www.w3.org/TR/InkML/. 2017-04-06.

[22] Q B Huang, B Y Zhang, T M Kechadi, "Preprocessing techniques for online handwriting recognition," in *Proc. Int. Conf. Intell. Syst. Design. Applica.*, pp. 793-800, Oct. 2007.

[23] W Yang, L Jin, Z Xie, et al., "Improved deep convolutional neural network for online handwritten Chinese character recognition using domain-specific knowledge," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, (ICDAR). pp. 551-555, Aug. 2013.

[24] Z Zhong, L Jin, Z Xie, "High performance offline handwritten Chinese character recognition using googlenet and directional feature maps," in *Proc. 13th Int. Conf. Document Anal. Recognit*., pp. 846-850, Aug. 2015.

[25] L Z Bai, Q Huo, "A study on the use of 8-directional features for online handwritten Chinese character recognition," in *Proc. 8th Int. Conf. Document Anal. Recognit*., pp. 262-266, Sept. 2005.

[26] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, Nov. 1946.

[27] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," [online]. Available: https://arxiv.gg363.site/abs/1409.1556

[28] K He, X Zhang, S Ren, et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), pp. 770–778, Aug. 2016,

[29] C Szegedy, W Liu, Y Jia, et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*. (CVPR), pp. 1–9, Jun. 2015.

[30] B. Ma, Z. Liu, F. Jiang, Y. Yan, J. Yuan, and S. Bu, "Vehicle Detection in Aerial Images Using Rotation-Invariant Cascaded Forest," *IEEE Access*, vol. 7, pp. 59613 - 59623, May. 2019.

[31] Hu J, Shen L, Sun G, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), pp. 7132–7141, Jun. 2018.

[32] S Ioffe, C Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, pp. 448–456, Jul. 2015.

[33] A Krizhevsky, I Sutskever, E G Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM.* Vol. 60, no.6, pp. 1097-1105. 2017.

[34] J I Goodfellow, D Warde-Farley, M Mirza, et al., "Maxout networks," [online]. Available: https://arxiv.org/abs/1302.4389, 2013.

[35] Y Wen, K Zhang, Z Li, et al., "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*, pp. 499–515, Sep. 2016.

[36] H Mouchère, R Zanibbi, U Garain, et al., "Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014," *J. Document. Anal. Recognit.*, vol. 19, no. 2, pp. 262-266, Mar. 2016.

[37] H Mouchere, C Viard-Gaudin, H D Kim, et al., "Crohme2011: Competition on recognition of online handwritten mathematical expressions," in *Proc. Int. Conf. Document. Anal. Recognit.*, pp. 1497-1500, Sept. 2011.

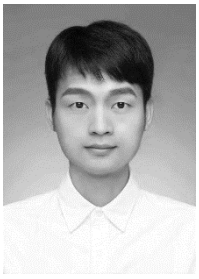
[38] K He, X Zhang, S Ren, et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.* (ICCV), pp. 1026–1034, Dec. 2015,

[39] D Kinga, B J Adam. "A method for stochastic optimization," [online]. Available: https://arxiv.org/abs/1412.6980. 2014.

[40] L. v. d. Maaten, G. Hinton, "Visualizing data using t-SNE", *J. Mach. Learn. Res.*, vol. 9, pp. 2579-2605, Nov. 2008.

[41] Y Cheng, M Yu, et al., "Few-shot learning with meta metric learners," [online]. Available: https://arxiv.xilesou.top/abs/1901.09890. 2019.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2979346, IEEE Access



Dingbang Fang is currently working toward the master's degree in the College of Information Science and Engineering, Huaqiao University, China. His current research interests include object recognition and deep machine learning, especially computer vision.

Chenhao Zhang received the master's degrees in computer science from Central China Normal University, in 2019. His research interests include machine learning and deep learning, especially computer vision