PAPER

Recognition of Online Handwritten Math Symbols Using Deep Neural Networks

Hai DAI NGUYEN[†a)], *Member*, Anh DUC LE[†], *Nonmember*, *and* Masaki NAKAGAWA[†], *Member*

SUMMARY This paper presents deep learning to recognize online handwritten mathematical symbols. Recently various deep learning architectures such as Convolution neural networks (CNNs), Deep neural networks (DNNs), Recurrent neural networks (RNNs) and Long short-term memory (LSTM) RNNs have been applied to fields such as computer vision, speech recognition and natural language processing where they have shown superior performance to state-of-the-art methods on various tasks. In this paper, max-out-based CNNs and Bidirectional LSTM (BLSTM) networks are applied to image patterns created from online patterns and to the original online patterns, respectively and then combined. They are compared with traditional recognition methods which are MRFs and MQDFs by recognition experiments on the CROHME database along with analysis and explanation.

key words: *CNN, BLSTM, gradient features, dropout, maxout*

1. Introduction

Online mathematical symbol recognition is an essential component of any pen-based mathematical formula recognition systems. With the increasing availability of touch-based or pen-based devices such as smart phones, tablet PCs and smart boards, interest in this area has been growing. However, existing systems are still far from perfection because of challenges arising from the two-dimensional nature of mathematical input and the category set with many similar looking symbols. In this work, we just address the problem of mathematical symbol recognition.

Handwritten character pattern recognition methods are generally categorized into two groups: online and offline recognition methods [1]. The former such as Markov Random Field (MRF) [2] is to recognize online patterns which are time sequences of strokes, and a stroke is a time sequence of coordinates of pen-tip or finger-tip trajectory recorded by pen-based or touch-based devices. The latter such as use of Modified Quadratic Discriminant Function (MQDF) and directional feature extraction [3] is to recognize offline patterns which are formed as two-dimensional images captured from a scanner or camera devices. Converting online patterns into offline patterns by discarding temporal and structural information make it possible to apply offline recognition methods for recognizing online patterns. Therefore, we can combine online and offline recognition methods to take advantage of them.

Manuscript received March 10, 2016.
Manuscript revised July 15, 2016.
Manuscript publicized August 30, 2016.

[†]The authors are with the Graduate School of Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, 184–8588 Japan.
a) E-mail: haidnguyen0909@gmail.com

DOI: 10.1587/transinf.2016EDP7102

Long short-term memory recurrent neural networks (LSTM RNNs) have been successfully applied to sequence prediction and sequence labeling tasks [4]. For online handwriting recognition, bidirectional LSTM (BLSTM) networks with a connectionist temporal classification (CTC) output layer using a forward backward type of algorithm have shown to outperform state-of-the-art HMM-based systems [5]. Recently, Alvaro et al. proposed a set of hybrid features that combine both online and offline information and using HMMs and BLSTM networks for classification [6]. BLSTM networks employing raw images as local offline features along pen-tip trajectory significantly outperformed HMMs in the symbol recognition rate. However, the experimental results showed that the proposed local offline features when combined with online features and BLSTM networks did not produce the great improvement. In Sect. 2 we describe the way of using directional (or gradient) features as local offline features following the method of Kawamura et al. [7] and show that the recognition rate is improved by adding more gradient features when combined with BLSTM networks.

With the recent success of Convolutional Neural Networks (CNNs) in many recognition tasks [8], a number of different nonlinearities have been proposed for activation functions of Deep Neural Networks. A nonlinearity that has recently become popular is the Rectified Linear Unit (ReLU) [9], which is a simple activation function $y = \max(0, x)$. More recently, the maxout nonlinearity [10], which could be regarded as a generalization of ReLU, is proposed. Maxout networks, combined with "dropout" [11], have also achieved improvement in computer vision tasks and outperformed the standard sigmoid and ReLU networks [10], [12]. In this paper, the maxout function is employed for both Convolution and Full-Connected layers along with drop out to build up a Deep Maxout Convolutional Network (DMCN).

A combination of multiple classifiers has been shown to be effective for improving the recognition performance in difficult classification problems [12], [13]. Also in online handwriting recognition, classifier combination has been applied [14], [15]. In this work, we simply employ a linear combination of DMCNs and BLSTM networks. Our experiments also show that the best combination ensemble has a recognition rate which is significantly higher than the rate achieved by the best individual classifier and the best combination of the previous methods on the CROHME database.

This paper is an elaborated and updated version from a conference paper [12] with more concise and formal presentation, extensive evaluation with statistical verification and detailed analysis. The rest of this paper is organized as follows: Sects. 2 and 3 present online and offline recognition methods and also mention to recent approaches of Deep Learning on this problem. Sections 4 and 5 report our experimental results and analyses. Section 6 draws conclusions.

2. Online Symbol Recognition Methods

This section describes briefly our previous online recognition method based on MRFs and then presents our new method by BLSTM networks.

2.1 MRF-Based Recognition Method

MRFs as well as HMMs match the feature sequence of an input pattern with states of each class. One advantage of MRFs compared to HMMs is capability of using both unary and binary features while HMMs can only adopt unary features [2].

For online handwritten symbol recognition, feature points must be extracted from the input and each class prior to using MRFs for matching. It can be done by the method by Rammer [16] as illustrated in Fig. 1(a).

Denote feature points of the input as sites $S = \{s_1, s_2, s_3, \dots, s_I\}$ and the states of a class C as labels $L = \{l_1, l_2, l_3, \dots, l_J\}$. The algorithm finds the class C that minimizes the energy function defined as follows:

$$E(O,F|C) = E(O|F,C) + E(F|C) = \sum_{i=1}^{I}[-\log P(O_{s_i}|l_{s_i},C) - \log P(O_{s_i s_{i-1}}|l_{s_i},l_{s_{i-1}},C)] - \log P(l_{s_i}|l_{s_{i-1}},C)] \quad (1)$$

Where l_{s_i} is the label of a class C assigned to s_i, O_{s_i} is the unary feature vector extracted from s_i and $O_{s_i s_{i-1}}$ is the binary feature vector extracted from the combination of s_i and s_{i-1}. For details of this method, the readers can refer to the paper by Zhu et al [2].

2.2 BLSTM-Based Recognition Method

This section outlines the principle of BLSTM RNNs used for the online symbol classification task and local gradient features to improve the accuracy.

2.2.1 Overview of LSTM

LSTM RNNs, which have been successfully applied to sequence prediction problems [4], are RNNs with memory blocks instead of regular blocks. Each block consists of one or more memory cells, along with gates: input, output and forget gates, as illustrated in Fig. 2, which allows the networks to remember information of previous states over a long period of time.

Bidirectional RNNs [17] make them possible to access to past and future context as well. BRNNs use two hidden layers, which are connected to the same output layer, therefore have access to contextual information from both directions. Figure 3 is a simple depiction of BRNNs. Thus, BLSTM networks can be derived from BRNNs by replacing regular units with LSTM blocks.

2.2.2 Feature Extraction for BLSTM Networks

This subsection describes features used with BLSTM networks.

One of problems of online classifiers is how to classify patterns having similar shapes or sampled point order. For example, Fig. 4 illustrates the case of two symbols ’0’ and ’6’. Using a contextual window around each sampled point is an obvious way to overcome this problem. Various ways can be used to extract features from these windows such as the approach presented by Alvaro at el. [6].

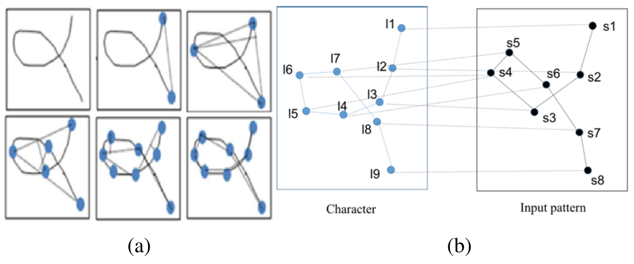

Fig. 1 Feature points extraction and labeling.

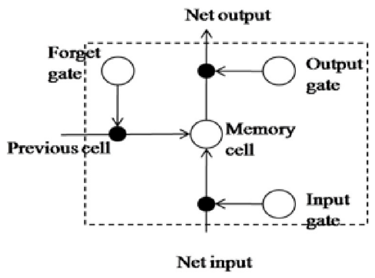

Fig. 2 An LSTM memory block.

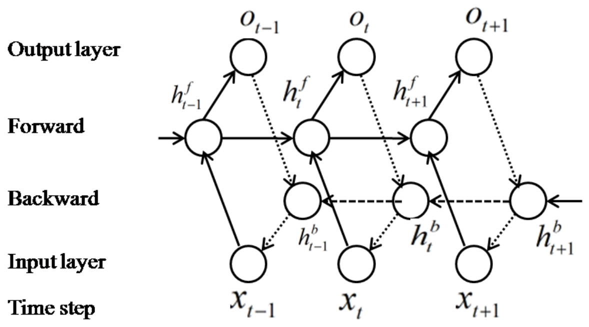

Fig. 3 Structure of a bidirectional network with input i, output o, and two hidden layers for forward and backward processing.

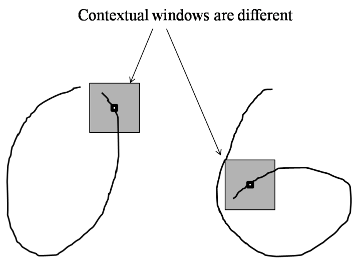

Fig. 4 Contextual window for each sampled point.

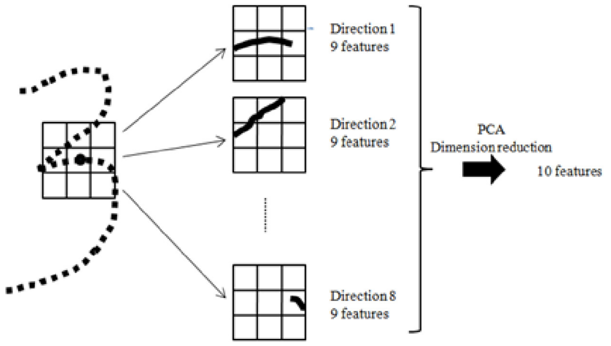

Fig. 5 Gradient feature extraction and PCA for dimension reduction.

They used raw images centered at each point to present the context information and then PCA for dimension reduction. However, the recognition rate has not been improved when used with BLSTM networks for online features since the classifier may not exploit features of raw images adequately. In [12], the gradient features are employed with good performance for character recognition.

Firstly, each online symbol pattern is linearly normalized to standard size (64x64). Then for each point p = (x, y), 6 time-based features are extracted:

- End point (1), otherwise (0).
- Normalized coordinates: (x, y)
- Derivatives: (x', y')
- Distance between point (i, i+1)

As for context information around each point combined with these online features, the gradient directional features are employed. Regarding gradient features: For each point p = (x, y), a context window centered at p is employed. From the context window, gradient directional features are decomposed into components in 8 chain code directions (depicted as Fig. 5). The context window is partitioned into 9 sub-windows of equal-sizes 5x5. The value for each block is calculated through a Gaussian blurring mask of size 10x10 and finally PCA is used to reduce the dimension into 20 dimensions.

3. Offline Symbol Recognition Methods

This section describes preprocessing and summarizes our previous offline recognition method based on MQDFs and then presents our new method based on CNNs.

3.1 Preprocessing

Coordinate normalization is done by interpolating pen-tip coordinates and applying normalization, which is either of the three normalization methods: linear normalization (LN) expanding the pattern to the unit size while keeping the horizontal and vertical ratio (LN), pseudo bi-moment normalization (P2DBMN) or line density projection interpolation (LDPI) [18].

3.2 MQDF Based Recognizer

After preprocessing, directional line segment feature extraction (SEG-FE) [18] is applied to extract 8 directions from 8 x 8 regions. Then, Fisher linear discriminant analysis (FDA) is used to reduce 8 x 8 x 8 dimensions into 160.

Symbol recognition consists of two steps: coarse and fine classification. Coarse classification reduces computation time on datasets with huge number of categories (e.g., Japanese, Chinese) and it nominates a smaller number of candidate classes according to Euclidean distance. Fine classification selects the output from the candidate classes using MQDFs defined as follows:

$$h_i(x) = \sum_{j=1}^{k} \frac{1}{\lambda_i}[\varphi_{ij}^T(x-\mu_i)] + \sum_{j=k+1}^{n} \frac{1}{\sigma^2}[\varphi_{ij}^T(x-\mu_i)] + \log\prod_{j=1}^{k}\lambda_{ij} + (n-k)\log\sigma^2 \tag{2}$$

where x is a feature vector, μ_i is the mean vector of the i^{th} class, k is the truncated dimensionality, λ_{ij} is the j^{th} eigenvalue of the covariance matrix of the i^{th} class and φ_{ij} is the corresponding eigenvector.

According to the previous work [19], the best performance using MQDFs is obtained when n is about 160 and k is about 50. Therefore, n is set as 160 and k as 50 for our experiments, respectively.

3.3 CNN Based Recognizer

As for Deep Learning based approaches, a preprocessed online pattern presented in 3.1 must be transformed to a two-dimensional image before it can be taken as input for the classifier. All normalized patterns are converted to image of size 48x48 with line thickness of 3 pixels.

Convolutional Neural Networks (CNNs) [8] are composed of alternating layers of convolution, pooling and full connected layers. In principle, CNNs consist of two stages: feature extraction and classification. Convolutional and pooling layers play the role of feature extraction and full connected layers play the role of classification. It is worth noting that both stages can be simultaneously trained while they must be separately trained for other classifiers.

There can be multiple filters in a convolutional layer

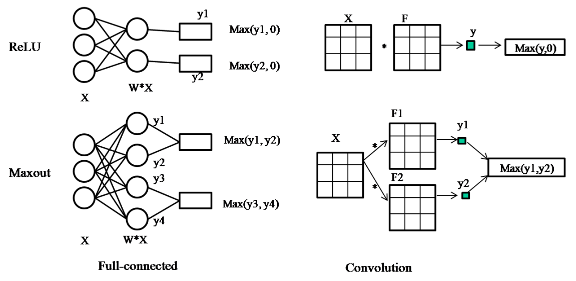

Fig. 6 A single maxout layer with the pool size K = 3.

which work as templates. Feature maps obtained by calculating the inner product of an input image (or feature maps of the previous layer) and corresponding templates measure how well the templates match each part of the image.

Each convolutional layer is usually followed by a pooling layer to reduce its dimensionality by grouping elements into regions and taking the maximum or average value within each region. These layers are important for classification task because extracted features are increasingly invariant to local transformation of the input image.

3.4 Deep Maxout Neural Network (DMCN)

Maxout proposed by Goodfellow et al. [10] has been shown to be more effective to improve the performance of the deep networks than Rectified Linear Unit (ReLU) and other activation functions when used along with Dropout [11]. As for full connected layers, nodes in detection layers are grouped and only maximum value is taken forward to the next layer, as illustrated in Fig. 6(a). Intuitively, maxout acts similarly to maxpooling in CNNs. The only difference is that max pooling is a way of taking the maximum node of a given interest region in the same channel while maxout is taking the maximum value within a group of nodes across channels, as illustrated in Fig. 6(b). An advantage of maxout is fast convergence during training because of smooth flow of back-propagated gradients while the networks still retain the same power of universal approximation as shown in [10].

4. Setting for Experiments

A series of experiments on mathematical symbol datasets are conducted to evaluate and compare the methods. This section reports our settings for the experiments.

4.1 Datasets

CROHME is a contest for online handwritten mathematical expression (ME) recognition, initially organized at ICDAR 2011 [19]. The sample pattern dataset is selected from 5 different MEs databases. The dataset contains 8,836 MEs for training as well as 761 MEs and 987 MEs for testing in the 2013 version and the 2014 version, respectively. In the last and current competition, there are 101 symbol classes. We extracted isolated symbols from them and named the dataset SymCROHME. Table 1 lists the organization of this dataset and Fig. 7 shows samples of various kinds of symbols.

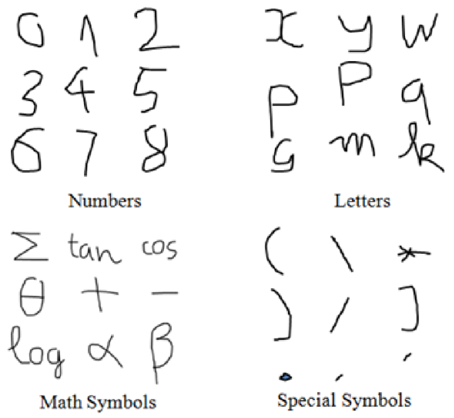

Fig. 7 Four kinds of symbols in SymCROHME.

Table 1 Organization of SymCROHME.

Subset name	**TrainCROHME**	**TestCROHME**	
		2013	**2014**
# symbols	120,341	6,080	9,999

A random 10% of the training set is reserved as the validation set and used for tuning the meta-parameters. In online mathematical symbol recognition, it is useful to consider N best rate (so-called Accuracy top-N in some materials) which is defined as the follow:

$$\text{N best rate} = \frac{\text{\#sample whose correct label lies on top N scores}}{\text{total number of samples}}$$

The reason is that the recognition performance of the whole system is based on not only recognition of isolated symbols but also other components i.e. structure analysis, grammar or context analysis.

4.2 Machine Environment

In order to speed up, the training stage of DMCNs is carried out on a NVIDIA Quadro K600 general purpose graphical processing unit (GPGPU), which contains 4GB of GDDR3 RAM. All the rest of experiments are implemented on an *Intel*(*R*)*Core*^{TM}2 Dual E8500 CPU 3.16GHz with 2.0 GB memory. RNNLIB is used for the implementation of the BLSTM networks [20].

5. Experiments, Results and Consideration

This section presents the results of the experiments and analyses on significant improvements by the neural networks. Paired t-test is used to verify the significance of improvements.

5.1 DMCNs and MQDFs

In this subsection, basic experiments for DMCNs are firstly described. Then, DMCNs and MQDFs are compared for offline recognition.

5.1.1 Basic Experiment of DMCN

To get the best performance, a series of experiments was carried out to determine a good configuration including the number of layers and the number of nodes in each layer for our CNN. Maxout units are adopted to improve the performance compared with ReLU except the first layer as suggested in [21], the bottom layers should be replaced by layers of a smaller number of ReLU units. The configuration is described as follows:

Our models are trained using stochastic gradient descent with a batch size of 64 samples, momentum of 0.95. The update rule for weight w is:

$$\begin{aligned} m_{i+1} &= 0.95 m_i - 0.0005 \epsilon w_i - \epsilon \text{Grad}_i \\ w_{i+1} &= w_{i+1} + m_{i+1} \end{aligned} \tag{3}$$

where i is the iteration index, m is the momentum variable, ϵ is the learning rate, and $Grad_i$ is the average over the ith batch of the derivative of the objective with respect to w, evaluated at w_i. The weights in each layer are initialized from a zero-mean Gaussian distribution with standard deviation 0.01 and the biases in each layer with the constant 0. An equal learning rate is used for all layers, which is adjusted manually throughout training. The heuristics which we follow is to multiply the learning rate by 0.5 when the validation error rate stops improving with the current learning rate. The learning rate is initialized at 0.01. The network is trained for 100 epochs and the training process is stopped when the error rate does not improve for 20 epochs which takes 2 to 3 hours on NVIDIA Quadro K600 4GB GPU.

5.1.2 Comparison of DMCN and MQDFs

Next, several experiments are conducted to compare DMCNs with MQDFs. In offline handwritten character recognition, nonlinear normalization based line density equalization has been proven effective when using with MQDFs, so that the three normalization methods are used as mentioned in Sect. 3, i.e., LN, P2DBMN or LDPI before the feature extraction stage in both the methods. For MQDFs, n is 160 and k is 50.

The results listed in Table 4 show that DCMNs outperform MQDFs significantly when using the linear and P2DBMN normalizations with $p<0.0001$ by two-tailed paired t-tests and slightly but not significantly with $p = 0.66$ when using LDPI on CROHME2013. On CROHME 2014, however, DCMNs outperform MQDFs significantly with any of LN, P2DBMN or LDPI with $p<0.0001$ by two-tailed paired t-tests. Therefore, it is supported that DMCNs outperform MQDFs.

Let us consider why CNNs perform better than MQDFs. The best feature of deep neural networks is depth. The depth of the CNNs allow each of its layers to compute complicated features which represent larger and more complex object parts as illustrated in Fig. 8. Let us focus on offline symbol recognition. In the first step, the edges should be extracted from the image. In the second step, small parts (or edges of edges) should be computed from the edges, such as corners. In the third step, combinations of small parts should be computed from the small parts. The idea is to extract progressively more abstract and specific units at each step. It is likely that character recognition consists of breaking an image up into small parts and increasing their complexity at each layer, which cannot be done with shallow models as MQDFs or SVMs. Also its large number of parameters and units allows it to do so finely, provided that the appropriate network parameters are tuned.

Table 2 Structure of our CNN.

Layer	Name	Size
1	Convolution-ReLU	32 filters size of 3x3
2	Max-Pooling	2x2
3	Convolution-Maxout	32 filters size of 2x2
4	Max-Pooling	2x2
5	Convolution-Maxout	48 filters size of 2x2
6	Max-Pooling	2x2
7	Convolution-Maxout	64 filters size of 2x2
8	Max-Pooling	2x2
9	Full-Connected	512
10	Softmax	101 (# of classes)

Table 3 N-best rates of MQDFs and DMCNs on TestCROHME2013 with three normalization methods.

Norm. methods	Recog. methods	N-best rate (%) 1	3	5
LN	**MQDF**	81.68	94.67	97.22
	DMCN	84.93	97.47	98.78
P2DBMN	**MQDF**	82.40	94.47	97.22
	DMCN	85.18	97.22	98.40
LDPI	**MQDF**	83.92	96.51	98.24
	DMCN	84.21	97.17	98.24

Table 4 N-best rates of MQDFs and DMCNs on TestCROHME2014 with three normalization methods.

Norm. methods	Recog. methods	N-best rate (%) 1	3	5
LN	**MQDF**	81.91	91.23	96.07
	DMCN	89.39	97.83	98.78
P2DBMN	**MQDF**	82.20	93.51	97.26
	DMCN	87.39	97.08	98.30
LDPI	**MQDF**	83.24	94.18	97.75
	DMCN	88.85	97.63	98.70

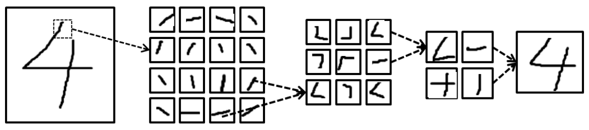

Fig. 8 Deeper layers compute elaborate features.

5.2 BLSTM Networks and MRFs

For online recognition methods, LN is used to normalize online patterns to size 48x48 to compare BLSTM networks and MRFs.

5.2.1 Basic Experiment of BLSTM Network

For LSTM networks, the BLSTM architecture is used in this work. The size of the input layer is 6 for online features and 16 for the combination of online and gradient features. Two hidden BLSTM layers are used with 32 and 128 memory blocks per layer. The output layer is softmax layer, whose size is 101, the number of mathematical symbol classes. The weights of the network are initialized from a zero-mean Gaussian distribution with standard deviation 0.1. The networks are trained using online stochastic gradient descent with the learning rate as 0.0001 and the momentum as 0.9. The training algorithm is stopped when the error rate is not improved for 20 epochs.

In order to evaluate the effectiveness of gradient features for BLSTM networks, experiments are conducted on three BLSTM networks: The first net only uses 6 online features mentioned in Sect. 2.2.2 for each point, the second net only uses gradient features and the final uses combination of online features and gradient features.

BLSTM networks with hybrid features outperform that used with only on-line features or gradient features (it is better than about 0.9 point, which is verified by paired t-tests with p= 1.48 at the risk $\alpha = 0.01$) The reason is because the proposed gradient features include more the context information on each point.

5.2.2 Comparison of BLSTM Network and MRFs

As for MRFs, we keep the setting as in [2]. For example, the coordinates of points and subtracting vector between points are used as unary and binary features, respectively. To estimate $P(O_{s_i} \mid l_{s_i}, C)$ and $P(O_{s_i s_{i-1}} \mid l_{s_i}, l_{s_{i-1}}, C)$ in Eq. (1), Gaussian functions with unary and binary features used as the mean and variance initialized by 1 are used. $P(l_{s_i} \mid l_{s_{i-1}}, C)$ is estimated as follows:

$$P(l_{s_i} \mid l_{s_{i-1}}, C) = \frac{\text{Number of transition from } l_{s_{i-1}} \text{to } l_{s_i}}{\text{Number of sites assigned } l_{s_{i-1}}} \tag{4}$$

$$P(l_{s_1} \mid l_{s_0}, C) = \frac{\text{Number of} s_1 \text{assigned } l_{s_1}}{\text{Number of } s_1} \tag{5}$$

The reader can refer to [2] for more details.

Next, experiments are conducted to compare N-best rates between BLSTM networks and MRFs. Table 7 shows the results. BLSTM networks outperform MRFs in all of 1-best, 3-best and 5-best rates. More specifically, BLSTM networks improve approximately 7 and 6 point of 1-best rate on TestCROHME2013 and TestCROHME2014, respectively. The significance is verified by two-tailed paired t-tests with p<0.0001 for both TestCROHME2013 and TestCROHME2014.

Therefore, it is supported that BLSTM networks outperform MRFs on CROHME2013 and CROHME2014.

Let us consider the reason why BLSTM networks perform better than MRFs in this work and they do better than HMMs [6] on online math symbol recognition. RNNs with LSTM blocks make them possible to integrate the influences of the previous nodes (even the future nodes for Bidirectional RNNs) without suffering from the gradient vanishing problem [4] while MRFs can take only neighboring points into consideration because of Markov assumption, as illustrated in Fig. 9. Moreover, local dependence of each state on only neighboring states makes MRFs less effective than RNNs.

Table 5 Comparative results among different features by BLSTM networks on TestCROHME2014.

Method	Architecture	1-best rate (%)
BLSTM networks with Online features	6 features: end point or not, x, y , x', y', d	86.8
BLSTM networks with gradient features	gradient feature	86.9
BLSTM networks with online+ gradient features	6 online + 20 gradient features	87.7

Table 6 Comparative results between MRFs and BLSTM networks on both testing data.

Dataset	**TestCROHME3013**			**TestCRHOME2014**		
Method	N-best rate(%)			N-best rate (%)		
	1	**3**	**5**	**1**	**3**	**5**
MRF	76.53	90.72	93.29	81.40	93.91	95.94
BLSTM	83.44	95.35	97.09	87.29	95.70	97.61

5.3 BLSTM+DMCN and MRF+MQDF

Combining multiple individual classifiers is proven effective to improve recognition rate for difficult classification prob-

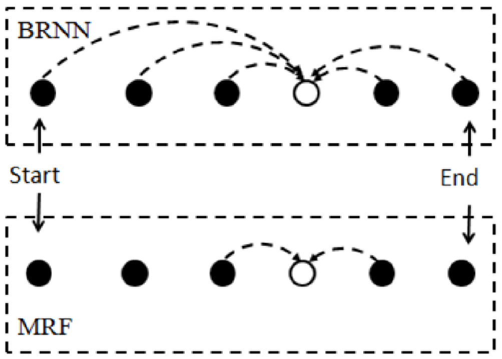

Fig. 9 The global and local dependence of each state in BRNN and MRF, respectively.

lems. Kittler et al. [14] analyze several combination methods both theoretically and practically. In our case, two classifiers are combined so that all the rules: max, min, product, sum and weighted sum except median and majority vote are considered. For weighted sum, a simple weighted average strategy of two classifiers is used for simplicity as the following formulation:

$$f_j^{linear}(u_1, u_2) = \alpha u_1(j) + (1 - \alpha)u_2(j) \tag{6}$$

Where $u_1(j)$ is softmax output of DMCNs or score of MQDFs , and $u_2(j)$ is softmax output of BLSTM networks or score of MRFs on j^{th} class, while is the weighting parameter for combination. The weighting parameter α is estimated on the validation set. More specifically, an experiment was conducted to determine that the parameter α of 0.6 and 0.65 were the best for the accuracy of the combination of MQDFs and MRFs, and the combination of DMCNs and BLSTM networks, respectively. The changes of the recognition rate on the validation set according to are illustrated as in Fig. 10.

Table 7 shows the recognition rates for combining DMCN and BLSTM with the five different combination strategies on two versions of testing sets. Note that we do not need normalization process as suggested in [14] because the output values are in the same range and scale. The results show that weighted sum yields improved recognition rates compared to the rest.

Then, experiments are made to compare the combination of DMCNs and BLSTM networks against that of MQDFs and MRFs. The validation set is used to tune the weighting parameter and the best value of α is obtained as 0.65 or 0.6 for combination of DMCNs and BLSTM networks or MQDFs and MRFs, respectively. The results listed on Tables 8, 9, 10 shows that the combined classifiers make significant improvement when compared with the individual classifiers and the combination of DMCNs and BLSTM networks outperform that of MQDFs and MRFs. In particular, the combination of MQDFs and MRFs improves approximately 2.2 point and 3.1 point on TestCROHME 2013 and TestCROHME 2014 from the best individual, respectively. The combination of DMCNs and BLSTM networks improve approximately 2.5 point and 1.6 point on TestCROHME 2013 and TestCROHME2014 from the best individual, respectively. The significance is verified by two-tailed paired t-tests with p<0.0001. N-best rate to the 3rd or 5th is more than 98% so that it will contribute to improve the entire ME recognition when combined with robust segmentation and structural analysis.

Combining online and offline classifiers takes advantages of both online and offline classification methods. More specifically, there are some characters making online classifiers difficult to discriminate from other symbols. For example, ‘0’ and ‘6’ have similar pen movement and sampled points (Fig. 4) so that it may be difficult for online classifiers like MRFs or LSTM networks to discriminate but they have different shapes then offline classifiers like MQDFs or CNNs can solve this problem easily. Another case is differentiating ‘B’ and ‘β’ which have the similar shapes but completely different pen movement so that online classifiers work. From these points of view, the effectiveness of com-

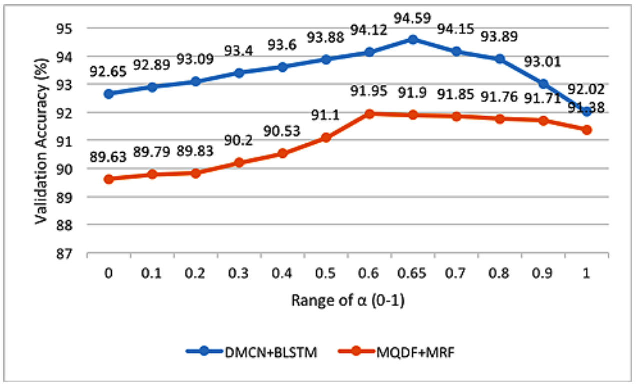

Fig. 10 Experiments on weighting parameter α

Table 7 Comparison of different combination methods

Com. Methods Testing set (year)	Max	Min	Product	Sum	Weighted Sum
2013	85.58	86.5	86.55	86.12	87.35
2014	88.85	90.18	89.96	89.33	91.28

Table 8 Recognition rates by combination of MQDFs and MRFs in comparison with individual classifiers.

Dataset	**TestCROHME2013**	**TestCRHOME2014**
Method	1-best rate(%)	1-best rate(%)
MQDF	83.68	82.89
MRF	76.53	81.40
MQDF+MRF α = 0.6	85.84	85.98

Table 9 1-Best rates of (top 1) by combination of DMCNs and BLSTM networks in comparison with individual classifiers.

Dataset	**TestCROHME2013**	**TestCRHOME2014**
Method	**1-best rate(%)**	**1-best rate(%)**
DMCN	84.93	89.39
BLSTM	83.44	87.29
DMCN+BLSTM (α = 0.65)	87.35	91.28

Table 10 Comparison of MRF+MQDF and BLSTM+DMCN

Dataset	**TestCROHME2013**		
Method	**N-best rate(%)**		
	1	**3**	**5**
MQDF+MRF	85.84	97.40	98.34
DMCN+BLSTM	87.35	97.76	98.51

Dataset	**TestCROHME2014**		
Method	**N-best rate(%)**		
	1	**3**	**5**
MQDF+MRF	85.98	96.90	98.70
DMCN+BLSTM	91.28	98.31	99.12

Table 11 1-Best rate (top 1) of our best system when compared with other systems on both testing sets without non-character class (i.e., 101 classes) (Results from CRHOME competition and F. Alvaro et al. [22], [23]).

Dataset	TestCROHME2013	TestCROHME2014
Systems	**1-best Rate(%)**	**1-best Rate(%)**
I	87.10	91.24
II	unavailable	82.72
III	unavailable	91.04
IV	unavailable	88.66
V	unavailable	85.00
VI	unavailable	84.31
VIII	unavailable	77.25
Ours	**87.35**	**91.28**

Table 12 Some frequent false recognition in CROHME database.

True	Recognized	# Recog. Err	
		TestCROHME2013	TestCROHME2014
.	x	37	75
1	1	53	52
Z	2	15	18
C	c	24	28
P	p	49	15
S	s	14	15
X	x	50	30
V	v	27	14

bining online and offline classifiers is supported.

Moreover, our system is compared with others published in [22], [23]. As shown in Table 11, our system outperforms slightly the system **I** which is currently holding the-state-of-the-art on TestCRHOME 2013 and 2014.

To end this section, some analyses on false recognitions shown in Table 12 are made. Almost of these cases are due to similar shapes of some symbols and confusion between lower and upper characters which is unavoidable because original characters are scaled to fixed size by the normalization methods. However, this problem can be solved by using other components of mathematical formula recognizer e.g. context and grammar analysis.

6. Conclusion

This paper studied Deep learning for recognizing online handwritten mathematical symbols. Experiments on the CROHME databases lead us to the following conclusions: 1) compared with MQDFs, Deep neural networks improve the offline recognition of mathematical symbols because they may extract wider yet specific features; 2) as for online methods, BLSTM networks outperform MRFs because they can access the whole context of the input sequence flexibly; 3) Combining both online and offline recognition methods improves classification performance by taking their advantages. Especially, the combination of BLSTM networks and DMCNs outperform that of MRFs and MQDFs.

There remains some works to improve this study. For deep learning, the size of training patterns is crucial. We have used the open pattern set (CROHME) for fair comparison, but we can still generate artificial patterns and use them for training. Moreover, more sophisticated network architectures and combining methods need be investigated. This classifier is integrated into a mathematical expression recognition system, where the recognition of the whole system will help to solve the problem of similar shaped classes.

Acknowledgments

The authors would like to thank Dr. Seiji Hotta for his advices. We would also thank Dr. Phan Van Truyen and other members in Nakagawa laboratory for giving helpful comments and suggestions.

References

[1] B. Zhu and M. Nakagawa, "Online Handwritten Chinese/Japanese Character Recognition," in Advances in Character Recognition, ISBN: 978-953-51-0823-8, InTech, DOI: 10.5772/51474, pp.51–68, 2012.
[2] B. Zhu and M. Nakagawa, "Online Handwritten Japanese Characters Recognition Using a MRF Model with Parameter Optimization by CRF," Proc. 10th International Conf. Doc. Ana. and Rec., Beijing, China, pp.603–607, Sept. 2011.
[3] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," Proc. IEEE Trans. Pattern Anal. Mach. Intell., vol.9, no.1, pp.149–153, 1987.
[4] A. Graves, "Supervised Sequence Labeling with Recurrent Neural Networks," inTextbook of Studies in Computational Intelligence, Springer, 2012.
[5] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," Proc. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI), vol.31, no.5, pp.855–868, 2009.
[6] F. Alvaro, J.-A.Sanchez, and J.-M.Benedi, "Classification of On-line Mathematical Symbols with Hybrid Features and Recurrent Neural Networks," Proc. 12th International Conf. on Doc. Ana. and Rec., Washington DC, USA, pp.1012–1016, Aug. 2013.
[7] A. Kawamura, K. Yura, T. Hayama, Y. Hidai, T. Minamikawa, A. Tanaka, and S. Masuda, "On-line recognition of Freely Handwritten Japanese Characters using Directional Feature Densities," Proc. 11th International Conf. on Pattern Recognit. (ICPR), vol.2. Conference B: Pattern Recognition Methodology and Systems, The Hague, pp.183–186, Sept. 1992.
[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," Proc. IEEE, vol.86, no.11, pp.2278–2324, 1998.
[9] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," Proc. 27thInternational Conference on Machine Learning (ICML), pp.131–136, Haifa, Israel, 2010.
[10] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv: 1302.4389, 2013.
[11] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," Proc. The Computing Research Repository (CoRR), 2012.
[12] H.D. Nguyen, A.L. Duc, and M. Nakagawa, "Deep Neural Networks on recognizing mathematical symbols," Proc. 3rd IAPR Asian Conference on Pattern Recognit. (ACPR), Kuala Lumpur, Malaysia, 2015.
[13] Y. Huang and C.Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," IEEE

Trans. Pattern Anal. Mach. Intell., vol.16, no.1, pp.66–75, 1994.

[14] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On Combining classifiers,” Proc. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI), vol.20, no.3, pp.226–239, 1998.

[15] H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, and M. Nakagawa, “Hybrid Pen-input Character Recognition System based on Integration of On-line, Off-line Recognition,” Proc. 5th International Conf. on Doc. Anal. and Rec. (ICDAR), Bangalore, pp.209–212, Sept. 1999.

[16] U. Ramer, “An iterative procedure for the polygonal approximation of plan closed curves,” Computer Graphics and Image Processing, vol.1, no.3, pp.244–256, 1972.

[17] M. Schuster and K.K. Paliwal, “Bidirectional recurrent neural networks,” Proc. IEEE Trans. Signal Processing, vol.45, no.11, pp.2673–2681, Nov. 1997.

[18] T.V. Phan, J. Gao, B. Zhu, and M. Nakagawa, “Effects of line densities on nonlinear normalization for online handwritten Japanese character recognition,” Proc. 10th International Conf. on Doc. Ana. and Rec. (ICDAR), Beijing, China, pp.834–838, Sept. 2011.

[19] CROHME: Competition on Recognition of Online Handwritten Mathematical Expression, in: 14th International conference on frontiers in handwriting recognition (ICFHR), Crete, Greece, Sept. 2014.

[20] A. Graves, RNNLIB: A recurrent neural network library for sequence learning problems, http://sourceforge.net/projects/rnnl/, 2013.

[21] P. Swieetojansky, J. Li, and J-T. Huang, “Investigation of maxout networks for speech recognition,” Proc. IEEE International Conference on Acoustic, Speech Signal Process. (ICASSP), Florence, Italy, pp.7649–7653, 2014.

[22] H. Mouchere, C.V. Gaudin, R. Zanibbi, and U. Garain, ICFHR 2014 Competition on Recognition of online Handwitten Mathematical Expression(CROHME 2014), Crete, Greece, Sept. 2014.

[23] F. Alvaro, J.-A. Sanchez, and J.-M.Benedi, “Offline features for classifying Handwritten Math Symbols with Recurrent Neural Network,” Proc. 22nd International Conf. on Pattern Recognit. (ICPR), Stockholm, Sweden, pp.2944–2949, Aug. 2014.

Hai Dai Nguyen received a B.Eng. degree in computer science from Hanoi University of Science and Technology, Vietnam in 2013. Since April 2014, he has been an M.Sc. student in the Department of Computer and Information Sciences at Tokyo University of Agriculture and Technology. He is currently working on machine learning, recognition of handwritten characters, and large-scale deep learning using high-performance computers.

Anh Duc Le received B.Eng. and M.Sc. degrees in computer science from Ho Chi Minh City University of Technology and Tokyo University of Agriculture and Technology2011 and 2014, respectively. Since April 2014, he has been a Ph.D. student in the Department of Electronic and Information Engineering at Tokyo University of Agriculture and Technology. He is currently working on machine learning, recognition of handwritten characters, and mathematical expressions.

Masaki Nakagawa received B.Sc. and M.Sc. degrees from the University of Tokyo in 1977 and 1979, respectively. During the 1977/78 academic year, he enrolled in the computer science course at Essex University in the UK and received an M.Sc.(with distinction) in computer studies in July 1979. He received a Ph.D. in information science from University of Tokyo in December 1988. He has been working at Tokyo University of Agriculture and Technology since April 1979. He is currently a Professor of Media Interaction in the Department of Computer and Information Sciences.