SN Computer Science (2020) 1:163
https://doi.org/10.1007/s42979-020-00177-0

ORIGINAL RESEARCH

Contextual Stroke Classification in Online Handwritten Documents with Edge Graph Attention Networks

Jun-Yu Ye[1,2] · Yan-Ming Zhang[1] · Qing Yang[1] · Cheng-Lin Liu[1,2,3]

Received: 7 January 2020 / Accepted: 20 April 2020 / Published online: 12 May 2020
© Springer Nature Singapore Pte Ltd 2020

Abstract
The task of grouping strokes into different categories is an essential processing step in the automatic analysis of online handwritten documents. The technical challenge originates from the variation of the handwriting style, content heterogeneity and lack of prior layout knowledge. In this work, we propose the edge graph attention network (EGAT) to address the stroke classification problem. In this framework, the stroke classification problem is formulated as a node classification problem in a relational graph, which is constructed based on the temporal and spatial relationship of strokes. Then distributed node and edge features for classification are learned by stacking of multiple edge graph attention layers, in which various attention mechanisms are exploited to aggregate information between neighborhood nodes. In the task of text/nontext classification, the proposed model achieves accuracies 98.65% and 98.90% on the IAMOnDo and Kondate datasets, respectively. In the task of multi-class classification, the achieved accuracies are 95.81%, 97.36% and 99.05% on the IAMOnDo, FC and FA datasets, respectively. In addition, we conduct ablation experiments to quantitatively and qualitatively evaluate the key modules of our model.

Introduction

With the rise of pen-based interfaces such as smart phones, tablets and electronic whiteboards in recent years, the need of automatic understanding of online handwritten documents is becoming more and more urgent. In the pipeline of analysis of online documents, a crucial step is to separate strokes into different categories to which different recognition engines can be applied. Therefore, the task of stroke classification is an essential processing step for text recognition, document retrieval or diagram recognition.

Stroke classification is difficult because of the variation of handwriting styles, complex 2D structure and the lack of prior layout knowledge. In the past, many research efforts have been made on the task of text and nontext classification as textual content is usually considered to carry semantic information and textual elements present regularities of stroke features [25]. As there are inherent contextual relationships between strokes, such as spatial and temporal relationships, the stroke classification problem is usually formulated under the structured prediction framework and addressed by various structured prediction methods [8, 14, 25, 33, 35].

Conditional random fields (CRFs) [20] and recurrent neural networks (RNNs) [12] are two mainstream structured prediction methods to model the stroke classification problem. Though the CRF is flexible and easy to integrate different types of contextual information, it is lacking in feature learning process and its computational cost and performance are severely affected by the structure of the probabilistic

This article is part of the topical collection "Document Analysis and Recognition" guest edited by Michael Blumenstein, Seiichi Uchida and Cheng-Lin Liu.

✉ Cheng-Lin Liu
liucl@nlpr.ia.ac.cn

Jun-Yu Ye
junyu.ye@nlpr.ia.ac.cn

Yan-Ming Zhang
ymzhang@nlpr.ia.ac.cn

Qing Yang
qyang@nlpr.ia.ac.cn

1 National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, People's Republic of China

2 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

3 CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, People's Republic of China

graph [19]. While the RNN has strong capacity to model the dependencies between temporal adjacent strokes, it is not easy to accommodate other types of contextual relationship because of the sequential nature of RNN.

In this paper, we present a new framework to address the online stroke classification problem. Essentially, we formulate stroke classification as the node classification problem in a relational graph and solve it with attention-based graph neural network (GNN). Specifically, we build one graph for each document, in which each node represents a stroke and edges are constructed based on the temporal and spatial relationship between strokes. Unary and pairwise geometric features are extracted as the raw node features and edge features in the graph. These components are fed into the edge graph attention network (EGAT), which is stacking by multiple edge attention layers to learn distributed node and edge representations for classification.

The main contributions of this work are summarized as follows:

1. We formulate the online stroke classification problem as the node classification problem in graph and propose to address it under the GNN framework.
2. We propose to use various attention mechanisms on unary and pairwise geometric features with edge update procedure to learn distributed node features and edge features.
3. Using the proposed edge graph attention network (EGAT), we achieve state-of-the-art performance on four public online handwritten document datasets, including IAMOnDo, Kondate, FC and FA. In addition, we perform comprehensive ablation experiments to evaluate the key modules of our model quantitatively and qualitatively.

This paper is an extension to a previous conference paper [34]. The extension is in several respects: the proposal of the edge update module, more experimental results on different datasets and more detailed ablation study.

The rest of the paper is organized as follows. Section 2 reviews related works in online stroke classification and GNN. Section 3 describes the details of the proposed method. Section 4 presents the experimental results, and Section 5 draws concluding remarks.

Related Work

Online Stroke Classification

The task of grouping strokes into different categories in online documents is a fundamental processing step in many document analysis systems. A basic task is to separate strokes into text/nontext categories, and many methods have been proposed to solve this problem [2, 8, 15, 16, 25, 33, 35]. These methods can be divided into different categories depending on how contextual information is exploited. In the following, we first review methods for isolated stroke classification and then structured prediction methods that exploit temporal or spatial relationship of strokes.

In the category of isolated stroke classification, Jain et al. [16] proposed an approach strictly relying on the local features without considering interactions between strokes. They used a linear classifier for classification and achieved 97% accuracy on their own dataset. Indermühle et al. [15] used a support vector machine classifier (SVM) with the same set of features on the IAMOnDo dataset and got the accuracy 91.3%. Considering that the category of strokes has some inherent ambiguity in shape, e.g., a small circle could be classified into a text stroke (representing the letter 'o') or a nontext stroke (representing a circle), Peterson et al. [24] proposed to extract local context features from surrounding strokes. It is shown that the accuracy could be significantly boosted by the local context features, and many structured prediction methods are also built on local context features [8, 25, 33].

The most obvious contextual information existing in online documents is the temporal dependency between successive strokes, and so, it is natural to consider stroke classification as a sequence labeling problem. Bishop et al. [2] proposed to use a hidden Markov model (HMM) for modeling the interactions between temporally successive strokes. A neural network with 11 input features is used to model the emission probability and the transition probabilities are estimated from data. It was shown that the HMM model significantly outperforms isolated stroke classification. A shortcoming of HMM is the assumption of conditional independence between observations [20]. Ye et al. [33] proposed a combined model of neural networks and CRF to overcome the above problem. Both the unary and pairwise potential functions in the CRF are modeled by neural networks, and all the parameters are learned simultaneously. Except for probabilistic graph models, RNN is also a powerful tool for sequence modeling. Indermühle et al. [14] proposed a mode detection method based on bidirectional long short-term memory (BLSTM). A stream of feature vectors is extracted from points of the pen trajectory, and BLSTM is applied to these features to capture the bidirectional temporal context features. Phan et al. [25] also proposed an approach based on BLSTM to address the text/nontext classification problem. They extracted global and local context features for strokes and fed those features into multiple BLSTMs for final predictions, achieving state-of-the-art accuracy on the text/nontext classification problem on the IAMOnDo and Kondate dataset.

Besides temporal contextual information, the spatial relationship between stroke is more informative. Zhou et al. [35] proposed to integrate spatial interactions between strokes with a Markov random field (MRF) model, and the potential functions are computed from the SVM classifier. Moreover, Delaye et al. [8] presented a system based on CRF integrating multiple types of contextual information. They proposed to exploit temporal, spatial, intersecting, lateral and stroke continuation relation to boost the system and performed ablation experiments to demonstrate the effect of each relation.

Graph Neural Networks

There has been an increased interest in adapting convolutions to tackle graph-based problems since the success of convolutional neural networks (CNN) in image recognition. Graph neural network (GNN) is a model that shares the similar concept with CNN but instead operates the convolution in graph structure. Defferrand et al. [7] proposed a graph convolutional operation as an analogue to standard convolution used in CNN. As convolution operation in spatial domain is equivalent to multiplication in the frequency domain, convolution operation applied in graph Laplacian is equivalent to filtering in the graph spectral domain. By applying Chebyshev polynomials to graph Laplacian, Kipf et al. [18] approximate the polynomials with a re-normalized first-order adjacency matrix and achieve comparable results on graph node classification tasks. To better capture the different influences brought by different neighbor nodes, Velickovic et al. [27] adapt the self-attention mechanism [26] into graph learning and propose the graph attention network (GAT). The introduction of self-attention mechanism in GAT integrates the node feature information for aggregating neighborhood nodes and improves the performance on many node classification tasks. Though GAT has great expressive power for graph feature learning, its message passing procedure only relies on the node features and the edge features are not fully incorporated. Variant models on how to exploit edge features are proposed to fill this gap. Gong et al. [11] proposed a unified framework to update the node features and edge features simultaneously in one layer and use doubly stochastic normalization technique to control the norm of the edge features and stabilize the training process. This work is most related to our method. Instead, our method uses the standard batch normalization technique, which is easier to implement in the common deep learning library.

Proposed Method

In this work, we formulate the online stroke classification problem as the node classification problem in relational graph. In the following, we first introduce the problem definition of stroke classification and then describe the pipeline of our model.

Problem Formulation

We are given a set of labeled online documents, in which each document $\mathbf{x}^{(i)}$ is formed by a sequence of strokes $\{\mathbf{x}_t^{(i)}, t = 1, \ldots, T_i\}$ and each stroke has one associated label $\mathbf{y}_t^{(i)} \in \{1, \ldots, L\}$, where L is the number of classes. Our goal is to learn a model from the training set S that can predict strokes in test documents with high accuracy.

Proposed Model

Figure 1 illustrates the framework of our model. Our model consists of three modules: construction of the relational graph, extraction of node and edge features, and edge graph attention network.

Graph Construction

In this work, an online handwritten document (a set of strokes) is represented as a relational $G(V, E)$, where each node $i \in V$ represents a stroke and each edge $(i, j) \in E$ represents the interaction between a pair of stroke (i, j). To build the graph, we integrate two types of contextual information, namely temporal context and spatial context.

Since the online nature of the data permits us to see the document as a sequence of strokes, the temporal neighborhood captures the interaction between strokes that are written successively in the document. To exploit this dependency, we define our temporal neighborhood system by considering every pair of temporally adjacent strokes. On the other hand, the spatial context is also a crucial source of knowledge, as spatially adjacent strokes tend to have same labels. We define our spatial neighborhood system by considering that two strokes $\mathbf{x}_i$ and $\mathbf{x}_j$ are spatial adjacent if their minimal distance $d(\mathbf{x}_i, \mathbf{x}_j)$ is below a given threshold T_1.

Figure 2 illustrates a sample of strokes and its associated document graph.

Feature Extraction

Each node and each edge of the document graph are associated with some features describing their geometric properties. We extract geometric and local context features of each

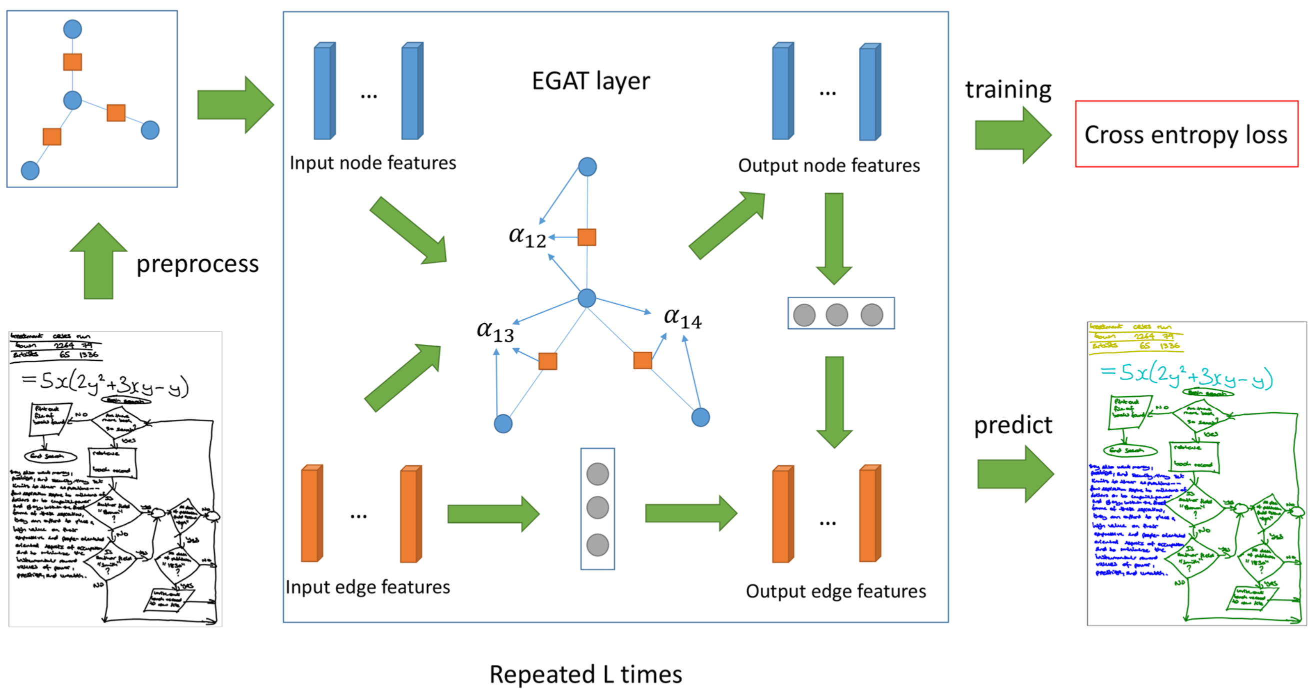

Fig. 1 An illustration of the training and prediction process for one input document. An input document is first preprocessed to generate node features, edge features and the document graph; then, these components are fed into networks stacking with multiple edge graph attention layers to learn context-aware features. During training, the model is learned by minimizing the cross-entropy loss. When performing prediction, the class with maximum probability is chosen for each stroke

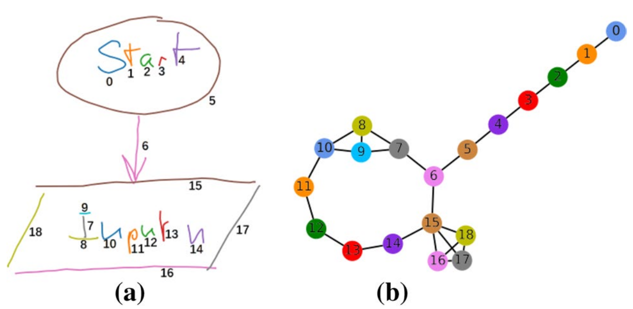

Fig. 2 An illustration of the strokes and its associated relational graph

stroke as node features and pairwise geometric features of pairs of strokes as edge features.

Node Features. We extract 13 contour-based shape features and 10 local context features following the similar procedure in [8, 25, 33] as node features. The shape features are extracted from each stroke directly, while the context features are extracted by considering its interactions with neighboring strokes. Table 1 lists all the node features.

Edge Features. We extract 19 pairwise stroke features as edge features in the graph for evaluating different importance of contextual information carried by different neighbors. These features compute the distance between pairs of strokes with various distance metrics and can be regarded as the indicators to measure similarity between two strokes. Therefore, it can be used to learn gates for controlling the message passing routine between nodes, which is beneficial for nodes to collect useful contextual information. Table 2 lists all the edge features.

Feature Normalization In order to make the density function of features closer to a Gaussian distribution, we process the features with power transform with the power coefficient 0.5.

$$x' = \mathrm{sign}(x)\sqrt{|x|}. \tag{1}$$

Furthermore, we normalize the values of each feature based on the mean μ and standard deviation σ (which are calculated based on the set of strokes in training documents) to standardize the feature values into the same scale. The normalized feature value is computed as

$$x'' = (x' - \mu)/\sigma. \tag{2}$$

Table 1 Node features extracted from stroke x_k

#	Description
1	Trajectory length of x_k
2	Area of the convex hull of x_k
3	Duration of the stroke
4	Ratio of the principal axis of x_k
5	Rectangularity of the minimum area bounding rectangle of x_k
6	Circular variance of points of x_k around its centroid
7	Normalized centroid offset along the principal axis
8	Ratio between first-to-last point distance and trajectory length
9	Accumulated curvature
10	Accumulated squared perpendicularity
11	Accumulated signed perpendicularity
12	Width of x_k, normalized by the median stroke height in the document
13	Height of x_k, normalized by the median stroke height in the document
14	Number of temporal neighbors of x_k
15	Number of spatial neighbors of x_k
16	Average of the distances from x_k to time neighbors
17	Standard deviation of the distances from x_k to time neighbors
18	Average of lengths of time neighbors
19	Standard deviation of lengths of time neighbors
20	Average of the distances from x_k to space neighbors
21	Standard deviation of the distances from x_k to space neighbors
22	Average of lengths of space neighbors
23	Standard deviation of lengths of space neighbors

Table 2 Edge features extracted from a pair of stroke x_i, x_j

#	Description
1	Minimum distance between 2 strokes
2	Minimum distance between the endpoints of 2 strokes
3	Maximum distance between the endpoints of 2 strokes
4	Distance between the centers of the 2 bounding boxes of 2 strokes
5	Horizontal distances between the centroids of 2 strokes
6	Vertical distances between the centroids of 2 strokes
7	Off-stroke distance between 2 strokes
8	Off-stroke distance projected on *X*- and *Y*-axes
9	Temporal distance between 2 strokes
10	Ratio of off-stroke distance to temporal distance
11	Ratio of off-stroke distance on *X*-, *Y*-axes to temporal distance
12	Ratio of area of the largest bounding box of 2 strokes to their union
13	Ratio of widths of the bounding boxes of 2 strokes
14	Ratio of heights of the bounding boxes of 2 strokes
15	Ratio of diagonals of the bounding boxes of 2 strokes
16	Ratio of areas of the bounding boxes of 2 strokes
17	Ratio of lengths of 2 strokes
18	Ratio of durations of 2 strokes
19	Ratio of curvatures of 2 strokes

Edge Graph Attention Network

The edge graph attention network is constructed by stacking *edge graph attention layers*. The input to each edge graph attention layer is a set of node features $\mathbf{H} = \{\mathbf{h}_i, i \in V\}, \mathbf{h}_i \in \mathbb{R}^C$ and a set of edge features $\mathbf{F} = \{\mathbf{f}_{ij}, (i,j) \in E\}, \mathbf{f}_{ij} \in \mathbb{R}^D$, where V is the set of nodes, E is the set of edges, and C, D are the numbers of features in each node and each edge. The output of layer is new sets of node features $\mathbf{H}' = \{\mathbf{h}'_i, i \in V\}, \mathbf{h}'_i \in \mathbb{R}^{C'}$ and edge features $\mathbf{F}' = \{\mathbf{f}'_{ij}, (i,j) \in E\}, \mathbf{f}'_{ij} \in \mathbb{R}^{D'}$, where C', D' are the numbers of output node features and edge features. In the following, the node feature update and edge feature update process will be described sequentially.

Node Feature Update In the progress of node update, we exploit two types of attention mechanisms to aggregate information between neighborhood nodes. The first one is the self-attention mechanism as the work [27]. A shared linear transformation is applied to every node by an attention mechanism $a : \mathbb{R}^{C'} \times \mathbb{R}^{C'} \rightarrow R$ to compute the node attention score.

$$s_{ij} = a(\mathbf{W}_h\mathbf{h}_i, \mathbf{W}_h\mathbf{h}_j), \tag{3}$$

where $\mathbf{W}_h \in \mathbb{R}^{C' \times C}$ is a learnable parameter. The attention mechanism a used in this work is the additive attention which is defined as

$$a(\mathbf{W}_h\mathbf{h}_i, \mathbf{W}_h\mathbf{h}_j) = \sigma(\mathbf{v}^T(\mathbf{W}_h\mathbf{h}_i + \mathbf{W}_h\mathbf{h}_j)), \tag{4}$$

where $\mathbf{v} \in \mathbb{R}^{C'}$ is a learnable parameter and $\sigma(\cdot)$ is leaky ReLU activation function.

The second one is the edge attention mechanism that explored in this work. Since the edge features can be regarded as the indicators to measure similarity between strokes, it is beneficial to integrate these similarity measures to the information collection process. So, we compute the edge attention score with edge features through one-layer neural network:

$$s'_{ij} = \sigma(\mathbf{v}_f^T\sigma(\mathbf{W}_f\mathbf{f}_{ij} + \mathbf{b}_f)), \tag{5}$$

where $\mathbf{W}_f \in \mathbb{R}^{C' \times D}, \mathbf{b}_f \in \mathbb{R}^{C'}, \mathbf{v}_f \in \mathbb{R}^{C'}$ are learnable parameters.

After computing the node attention score and edge attention score, the final attention score is the sum of these two scores after normalization:

$$\alpha_{ij} = \text{softmax}_{\beta}(s_{ij} + s'_{ij}) = \frac{\exp(\beta(s_{ij} + s'_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\beta(s_{ik} + s'_{ik}))}, \tag{6}$$

where β is the temperature hyperparameters and $\mathcal{N}_i$ is the neighborhood of node i. When $\beta = 0$, the model degrades to the vanilla graph convolutional network.

The output of node feature is then computed by the weighted combination of the neighborhood features with the attention coefficients:

$$\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_h \mathbf{h}_j\right). \tag{7}$$

In this work, we also introduce the standard technique *multi-head attention* in our model. Specifically, K independent attention mechanisms execute the transformation in equation (7) and their outputs are concatenated:

$$\mathbf{h}'_i = \Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}_h^k \mathbf{h}_j\right), \tag{8}$$

where $\Vert$ represents concatenation.

Edge Features Update. After the update of the node features, we propose a simple method to update edge features for including more contextual information in the edges. Given the output of node update features $\mathbf{H}' = \{\mathbf{h}'_i, i \in V\}$ and the previous edge features $\mathbf{F} = \{\mathbf{f}_{ij}, (i,j) \in E\}$, the update of edge features $\mathbf{f}'_{ij}$ can be expressed as

$$\begin{aligned} \mathbf{r}_{ij} &= \sigma(\mathbf{W}_{\text{node}}[\mathbf{h}'_i \,||\, \mathbf{h}'_j \,||\, |\mathbf{h}'_i - \mathbf{h}'_j|]), \\ \mathbf{t}_{ij} &= \sigma(\mathbf{W}_{\text{edge}} \mathbf{f}_{ij}), \\ \mathbf{f}'_{ij} &= \sigma(\mathbf{W}_{\text{reduce}}[\mathbf{r}_{ij} \,||\, \mathbf{t}_{ij}]), \end{aligned} \tag{9}$$

where $|\cdot|$ is elementwise absolute function, $\mathbf{W}_{\text{node}} \in \mathbb{R}^{3C' \times C'}$, $\mathbf{W}_{\text{edge}} \in \mathbb{R}^{D \times C'}$, $\mathbf{W}_{\text{reduce}} \in \mathbb{R}^{2C' \times D'}$ are learnable parameters.

Finally, to stabilize and accelerate the training, we employ some standard techniques, including residual connection and batch normalization. In this work, all the hyperparameters of hidden layers are the same, and the residual connection is added except the first layer:

$$\begin{aligned} \mathbf{h}'_i &= \text{BatchNorm}(\mathbf{h}_i + \text{NodeUpdate}(\mathbf{h}_i)), \\ \mathbf{f}'_{ij} &= \text{BatchNorm}(\mathbf{f}_{ij} + \text{EdgeUpdate}(\mathbf{f}_{ij}, \mathbf{h}'_i, \mathbf{h}'_j)). \end{aligned} \tag{10}$$

Network Training

We implement the training algorithm for the EGAT model with the dgl library [30] and its pytorch backend [23].

For optimizing the parameters of the networks in training, we use the standard cross-entropy as the loss function, which is defined as

$$L(\mathbf{W}) = -\sum_{i=1}^{N} \sum_{t=1}^{T_i} \log \mathbf{p}_t[y_t^{(i)}]. \tag{11}$$

The weight matrix parameters are initialized with normal samples $\mathcal{N}(0, \sigma^2)$, where $\sigma = \sqrt{\frac{2}{r+c}}$ and r, c are the number of rows and columns in the matrix [10].

Parameter optimization is performed with Adam [17]. We choose the initial learning rate $\eta_0 = 0.005$, and the learning rate decays according to the validation accuracy. The decay rate $\rho = 0.1$. We use early stopping based on the performance on the validation set, and the best parameter is chosen by the best accuracy on the validation set.

The hyperparameters for all experiments are specified in 'Appendix A.2.'

Experiments

To evaluate the performance of the proposed EGAT mode, we perform experiments on four datasets: IAMOnDo, Kondate, FC and FA. For each dataset, we first report the results of our model and compare with previous methods. Then, we present the results of a series of ablation experiments to evaluate the effects of key modules. To reduce the volatility of the system, we conduct each experiment 10 times and report the mean accuracy and standard deviation for each model.

Datasets

IAMOnDo. The IAMOnDo dataset [15] is a publicly available collection of freely handwritten English online documents. The dataset consists of 1000 documents, mixing texts, drawing, diagrams, formulas, tables, lists and marking elements arranged in an unconstrained way. The dataset is split into five disjoint sets, each containing roughly 200 documents. We follow the traditional data partitioning of the dataset [15]. Documents from *set0* and *set1* constitute the training set, *set2* is the validation set, and *set3* is the independent test set. In this dataset, we perform two sets of experiments to evaluate the performance of our model. In the first set of experiments, strokes are classified into text or nontext, and the corresponding ground truth labels are derived as suggested in [13]. In the second set, we consider to separate strokes into graphic/text/table/list/math five classes as in [9].

Kondate The Kondate dataset [22] is a collection of online freeform handwritten Japanese documents. The dataset contains the documents written by 67 writers, 41 pages per writer covering the stroke types of text, formula, figure, ruled line and editing mask, and it has been used in text/nontext classification experiments reported in [25, 35]. We follow the data partitioning of the dataset as in [25], 210 pages are used for training, 100 pages for validation and 359 pages for testing.

FC The FC dataset [1] is an online handwritten flowchart dataset. The dataset contains the documents drawn by 46 writers. In the original partition, 248 diagrams are used for training and 171 programs for testing. We further split the training data with 220 diagrams for training and 28 diagrams for validation.

FA The FA dataset [5] is an online handwritten finite automata diagram dataset. The dataset contains the documents acquired from 25 people. In this dataset, 132 diagrams are used for training, 84 diagrams for validation and 84 diagrams for testing, respectively.

The statistics of the datasets are presented in 'Appendix A.1.'

Evaluation Metrics

In this work, we evaluate our model on text/nontext classification and multi-class classification. For text/nontext classification, the accuracy is used as the evaluation metric, which is defined as:

$$\text{accuracy} = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \delta(\hat{y}_{it} = y_{it})}{\sum_{i=1}^{N} T_i}, \tag{12}$$

where N is the number of documents, T_i is the number of strokes in i-th document. $\hat{y}_{it}$ and y_{it} are the corresponding prediction and ground truth.

For multi-class classification, we also report the accuracy for each class c, which is defined as

Table 3 Performance of different methods for text/nontext stroke classification on the IAMOnDo dataset

Method	Description	Accuracy
Indermühle et al. [15]	Local classification with SVM	91.30
Weber et al. [31]a	Multiple classifiers system	97.00
Indermühle et al. [14]	BLSTM network	97.01
Delaye et al. [8]	CRF with multiple contexts	97.21
PCC [25]	Global and local context integrated	97.96
Ye et al. [33]	Joint training of MLP and CRF	97.96
Phan et al. [25]a	Global and local classifiers ensembled	98.30
GCN	Graph convolutional networks	97.21 ± 0.07
GAT	Graph attention networks	97.87 ± 0.05
EGAT(w/o EU)	EGAT without edge updates	98.59 ± 0.06
EGAT(w/o space)	EGAT without spatial edges	98.52 ± 0.04
EGAT	Edge graph attention networks	98.65 ± 0.05

An "a" indicates ensembling of multiple classifiers

Table 4 Performance of different methods for text/nontext stroke classification on the Kondate dataset

Method	Description	Accuracy
Zhou et al. [35]	MRF with SVM potentials	96.61
GPC19Q_LSTM [25]	Pair classifiers with BLSTM	97.17
GSC26_LSTM [25]	Global context based BLSTM	98.13
LCC [25]	Global and local context integrated	98.65
Phan et al. [25]a	Global and local classifiers ensembled	**99.04**
GCN	Graph convolutional networks	97.76 ± 0.18
GAT	Graph attention networks	98.27 ± 0.17
EGAT(w/o EU)	EGAT without edge update	98.66 ± 0.17
EGAT(w/o space)	EGAT without spatial edges	98.76 ± 0.10
EGAT	Edge graph attention networks	**98.90 ± 0.12**

An "a" indicates ensembling of multiple classifiers

$$\text{accuracy}[c] = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \delta(y_{it} = c)\delta(\hat{y}_{it} = y_{it})}{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \delta(y_{it} = c)}. \tag{13}$$

Comparison with Previous Methods

Text/Nontext Classification

Tables 3 and 4 illustrate the results of our model for text/nontext classification on IAMOnDo dataset and Kondate dataset, respectively, with previous top-performance methods for comparison. On the IAMOnDo dataset, our model

Table 5 Performance of different methods for multi-class stroke classification on the IAMOnDo dataset

Method	Graphics	Text	Table	List	Math	Overall
Loopy CRF [9]	–	–	–	–	–	79.22
Delaye et al. [9]	95.85	97.25	77.64	74.73	84.28	93.46
GCN	93.09 ± 0.74	97.97 ± 0.34	70.31 ± 1.41	52.73 ± 2.82	74.48 ± 1.80	91.11 ± 0.27
GAT	95.12 ± 0.37	98.19 ± 0.21	77.59 ± 2.31	69.66 ± 1.21	82.22 ± 2.91	93.51 ± 0.33
EGAT(w/o EU)	95.75 ± 0.47	98.21 ± 0.37	88.85 ± 0.98	73.85 ± 1.25	86.78 ± 1.35	95.14 ± 0.38
EGAT(w/o space)	93.86 ± 0.57	97.42 ± 0.40	81.58 ± 1.10	72.68 ± 0.78	85.53 ± 1.05	93.42 ± 0.20
EGAT	**97.11 ± 0.38**	**98.35 ± 0.24**	**89.70 ± 1.86**	**76.15 ± 1.78**	**88.43 ± 1.53**	**95.81 ± 0.29**

Table 6 Performance of different methods for multi-class stroke classification on the FC dataset

Method	Text	Arrow	Data	Decision	Process	Terminator	Connection	Overall
Lemaitre et al. [21]	97.8	79.6	84.7	84.0	85.7	58.9	80.0	91.1
Carton et al. [6]	97.2	83.8	84.3	90.9	90.4	69.8	80.3	92.4
Wu et al. [32]	98.8	87.4	87.6	89.7	91.8	91.6	73.3	94.9
Bresler et al. [5]	99.0	85.3	95.6	90.8	93.7	89.7	93.3	95.2
Wang et al. [29]	99.0	92.1	84.4	89.9	93.5	78.9	79.3	95.8
Wang et al. [28]	**99.3**	90.5	94.6	92.5	94.7	78.5	91.9	96.2
Bresler et al. [4]	99.2	87.5	95.3	88.2	**96.3**	90.7	**94.1**	96.3
Bresler et al. [3]	**99.3**	88.7	**96.4**	90.9	95.2	90.2	**94.1**	96.5
GCN	97.74 ± 0.48	89.44 ± 1.13	88.67 ± 1.29	92.92 ± 1.26	82.30 ± 3.37	85.96 ± 1.26	85.19 ± 1.90	93.99 ± 0.23
GAT	98.01 ± 0.34	88.21 ± 2.18	88.57 ± 3.29	92.27 ± 1.67	82.39 ± 2.55	87.09 ± 5.01	87.56 ± 3.80	94.00 ± 0.30
EGAT(w/o EU)	98.39 ± 0.27	93.21 ± 1.50	93.43 ± 1.37	94.18 ± 1.26	88.72 ± 2.02	90.02 ± 1.59	87.56 ± 2.67	95.98 ± 0.43
EGAT(w/o space)	98.24 ± 0.50	89.55 ± 2.14	90.38 ± 1.32	90.73 ± 2.39	74.98 ± 5.58	75.52 ± 3.38	65.56 ± 6.59	93.33 ± 0.82
EGAT	98.84 ± 0.20	**96.39 ± 0.50**	94.40 ± 1.02	**95.19 ± 1.60**	92.94 ± 1.80	**93.23 ± 1.53**	92.89 ± 2.20	**97.36 ± 0.31**

Table 7 Performance of different methods for multi-class stroke classification on the FA dataset

Method	Label	Arrow	Initial arrow	State	Final state	Overall
Bresler et al. [5]	99.1	89.3	78.5	95.2	96.1	94.5
Bresler et al. [3]	**99.8**	94.9	85.0	96.9	99.2	97.4
Wang et al. [29]	99.0	97.7	–	91.6	96.5	97.8
Bresler et al. [4]	99.7	98.0	**98.6**	**98.3**	**99.2**	99.0
GCN	94.04 ± 0.80	91.75 ± 1.04	–	84.18 ± 2.16	87.81 ± 3.55	92.13 ± 0.62
GAT	99.01 ± 0.39	97.89 ± 0.48	–	92.30 ± 1.07	94.81 ± 2.93	97.87 ± 0.46
EGAT(w/o EU)	98.99 ± 0.35	98.78 ± 0.35	–	97.04 ± 0.32	96.77 ± 0.67	98.64 ± 0.26
EGAT(w/o space)	98.68 ± 0.33	98.43 ± 0.35	–	96.69 ± 0.70	85.73 ± 2.90	97.63 ± 0.35
EGAT	99.49 ± 0.15	**99.08 ± 0.10**	–	97.18 ± 0.36	97.42 ± 0.67	**99.05 ± 0.13**

achieves 98.65% accuracy with the single model and significantly outperforms all the single models based on CRF or BLSTM with 97.96% accuracy. Moreover, our model achieves 0.35% improvements of accuracy over the model in [25], which is an ensemble of global and local BLSTM networks. Compared with the method in [8] that has the similar concept to explore multiple context information with CRF, our model is shown to be more effective to exploit the interaction between contextual strokes. On the Kondate dataset, our model achieves 98.90% accuracy, outperforming the previous single model based LCC [25] by 0.25%.

Multi-class Classification

Tables 5, 6 and 7 compare the results of our model for multi-class classification on IAMOnDo, FC and FA datasets, respectively. On the IAMOnDo dataset, our model achieves the accuracy 95.81%, which is 2.35% than that of the model in [9], and is superior on each class, especially on the class 'Table' and the class 'Math.' In [9], Delaye et al. proposed a hierarchical CRF framework and used distance learning to construct the tree CRF structure. It shows that their model based on CRF could not support the arbitrary graph structure and a massive performance decrease occurred when the CRF graph is no longer a tree. On the contrary, our model can build the graph with arbitrary graph structure, and the increasing number of edges brings performance improvements.

On the FC dataset, the accuracy of our model is 97.36% and is 0.86% higher than that of the previous best method [3]. On the FA dataset, our model yields 99.05% accuracy, which is on par with the previous best method [4] with accuracy 99.0%.

Ablation Study

To identify the critical factors in the success of our proposed model, we conduct ablation study in four different aspects. Specifically, we evaluate the effect of the attention mechanisms, edge update procedure, integration of spatial edges and the number of attention layers. Table 8 presents the methods and its associated modules. The experimental results of these compared methods for text/nontext classification are given in Tables 3 and 4, and the results for multi-class classification are given in Tables 5, 6, and 7.

Table 8 Methods and its associated modules. 'SA' denotes self-attention, 'EA' denotes edge attention, 'EU' denotes edge update, 'space' denotes spatial edges

Method	SA	EA	EU	space
GCN				✓
GAT	✓			✓
EGAT(w/o EU)	✓	✓		✓
EGAT(w/o space)	✓	✓	✓	
EGAT	✓	✓	✓	✓

Evaluation of Attention Mechanisms

To evaluate the effect of different attention mechanisms used in the model, we implement the vanilla GCN model without any attention mechanisms by setting the temperature β to 0 in Equation (7). Then we add the self-attention mechanism and edge attention mechanism progressively to see the performance change. In text/nontext classification, the vanilla GCN achieves accuracies 97.21% and 97.76% on the IAMOnDo and Kondate datasets, respectively. With the self-attention mechanism, the accuracy of model GAT increases to 97.87% and 98.27% with improvements 0.66% and 0.51%. In multi-class classification, the vanilla GCN model achieves accuracies 91.11%, 93.99% and 92.13% on the IAMOnDo, FC and FA datasets, respectively. The model GAT yields 2.40% and 5.74% improvements on the IAMOnDo and FA datasets, with accuracies 93.51% and 97.87%, respectively. On the FC dataset, the GAT yields accuracy 94.00%. These results show that the self-attention mechanism can effectively boost the performance of the graph model just by integrating node features into the message passing procedure.

Next, we further add the edge attention mechanism to the model. In text/nontext classification, the model EGAT(w/o EU) yields accuracies 98.59% and 98.66% on the IAMOnDo and Kondate datasets, with improvements 0.72% and 0.39%, respectively. In multi-class classification, the model EGAT(w/o EU) yields accuracies 95.14%, 95.98% and 98.64% on the IAMOnDo, FC and FA datasets, with improvements 1.63%, 1.98% and 0.77%, respectively. These results show that pairwise stroke features play an important role in the stroke classification task and have good complementarity with node features. Moreover, it proves that the proposal of the edge attention mechanism can effectively exploit the information in edge features.

Evaluation of Spatial Edges

To evaluate the effect of spatial edges, we build the document relational graph without the spatial edges and only temporal edges are considered. In text/nontext classification, the accuracy of model EGAT(w/o space) yields accuracies 98.52% and 98.76% on the IAMOnDo and Kondate datasets, respectively, which are slightly lower than those of the EGAT model, by 0.13% and 0.14%, respectively. In multi-class classification, the accuracies of model EGAT(w/o space) are 93.42%, 93.33% and 97.63% on the IAMOnDo, FC and FA datasets, respectively, which drops severely compared to the EGAT model, by 2.00%, 4.03% and 1.42%, respectively. From the results, we can see that the only usage of temporal information is sufficient to build a high-performance system for text/nontext classification and adding spatial edges has little effect to the system performance. This observation is also consistent with the work [8]. On the contrary, the spatial information plays an important role in the multi-class classification scenario, especially for the class 'Table' of the IAMOnDo dataset, classes 'Arrow,' 'Process,' 'Terminator,' 'Connection' of the FC dataset and class 'Final state' of the FA dataset.

Evaluation of Edge Update

From Table 2, we can see that the raw edge features mainly contain the information about distances and differences between two local strokes and do not have much node information. So we design the edge update module for capturing more node information in the edges and expanding receptive fields of edge features to better control the node feature aggregation procedure. In text/nontext classification, the model EGAT(w/o EU) yields accuracies 98.59% and 98.66% on the IAMOnDo and Kondate datasets, respectively, which are slightly lower than those of the EGAT model by 0.06% and 0.24%, respectively. In multi-class classification, the model EGAT(w/o EU) achieves accuracies 95.14%, 95.98% and 98.64% on the IAMOnDo, FC and FA datasets, respectively, decreasing by 0.67%, 1.38% and 0.41% compared to the EGAT model. This shows that raw edge features are sufficient indicators for judging whether two strokes should belong to the same class or not in the text/nontext classification scenario. But in the multi-class scenario, learning distributed edge features can capture more class dependencies between strokes and improve the classification performance significantly.

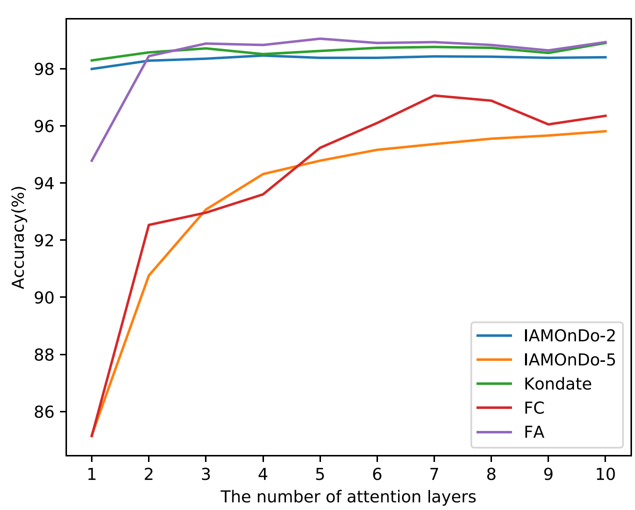

Fig. 3 The performance of different number of attention layers on different datasets

Evaluation of the Number of Attention Layers

Like convolutional layers used in computer vision or natural language processing, every attention layer in graph network model gathers contextual information from the adjacent nodes, and the model with more attention layers tends to learn features with larger size of receptive fields. To evaluate the effect of number of attention layers, we show in Fig. 3 the accuracies of the EGAT model on different datasets with respect to different number of attention layers. We can see that the performance increases with increasing number of layers progressively before a threshold and nearly saturate after that. The performance change is more significant in the multi-class scenario compared to the text/nontext scenario. We hypothesize that the text/nontext classification task is a task that needs more local information and local smoothing, so the model with a small number of attention layer also performs well. But in the multi-class scenario, larger receptive field is required for classification into different semantic objects as the local geometric features do not bring enough contextual information.

Qualitative Analysis

Figure 4 presents examples of stroke classification results on the IAMOnDo, Kondate, FC and FA datasets. The results are obtained by ground truth stroke labels, the EGAT(w/o space) and the EGAT model. In the figure, we can see that the model with spatial edges produces the more consistent and reasonable results and it also demonstrates the necessity of integrating spatial information qualitatively.

Computational Complexity

We perform experiments on a computer with Intel(R) Core(TM) i7-6800K CPU(3.40GHz) and GeForce GTX 1080Ti. Tables 9, 10, 11 present the number of parameters, training time and test speed on different datasets. In the text/nontext classification task of IAMOnDo dataset, the inference time of our model is 0.44s on the entire test

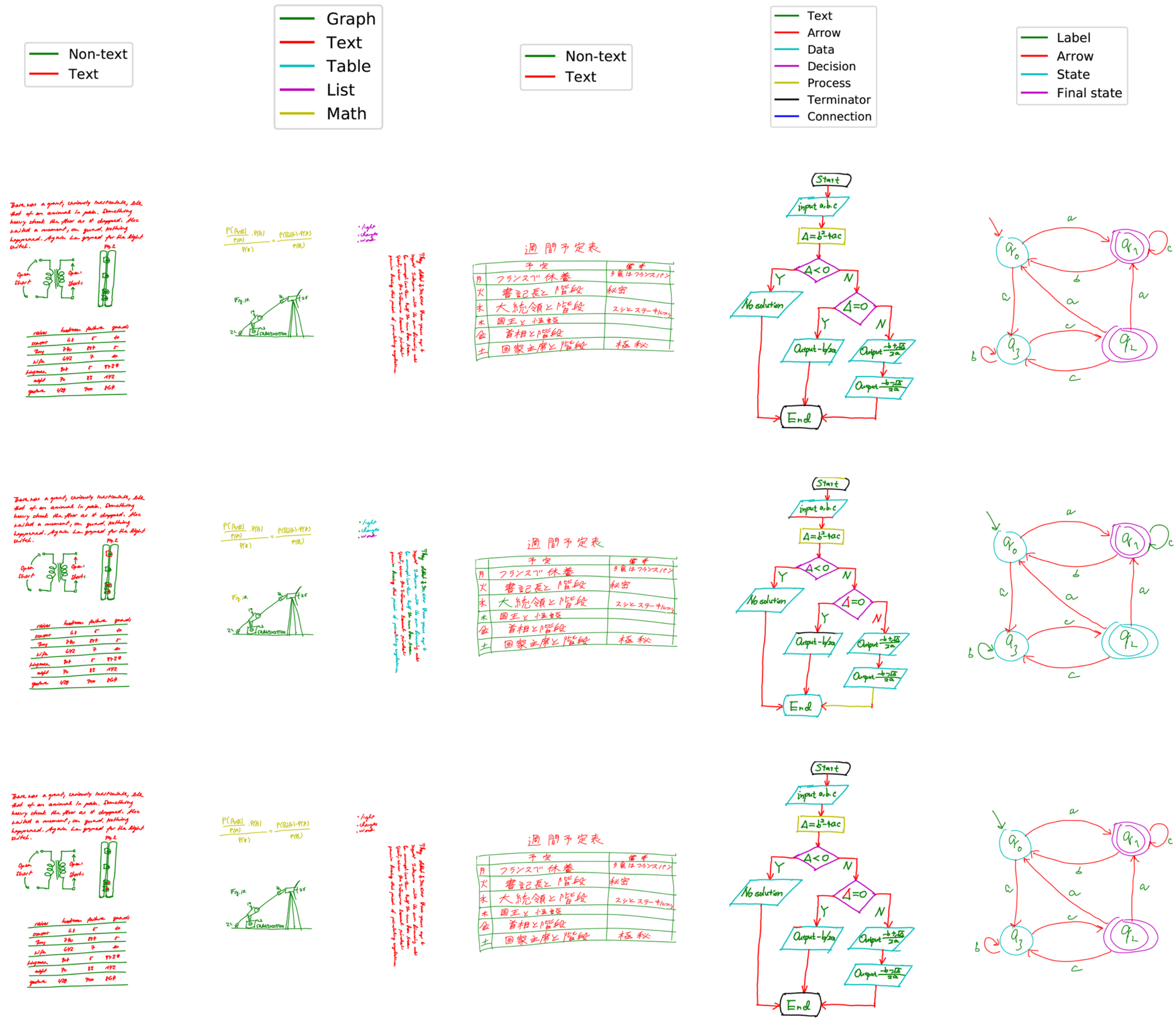

Fig. 4 Examples from IAMOnDo, Kondate, FC and FA datasets. The first row shows the ground truth stroke labels of documents (different classes in different colors). The second row shows the predicted results by the EGAT(w/o space) model. The third row shows the predicted results by the EGAT model

Table 9 The number of parameters, training time and test speed on different datasets

Dataset	Parameters	Training time(s)	Test speed(s/doc)
IAMOnDo-2	0.76M	600	0.0021
IAMOnDo-5	1.63M	1680	0.0033
Kondate	1.11M	120	0.0016
FC	0.15M	180	0.0020
FA	0.15M	300	0.0036

set, which is faster than the PCC model in [25] by 3.23s, NN-CRF model in [33] by 0.69s and Delaye et al. [8] by 190s. Overall, the proposed EGAT model is shown to be computationally efficient in both training and testing phases.

Conclusion

In this paper, we present a novel framework for stroke classification in online handwritten documents. The representation of strokes is learned by graph convolution

Table 10 Statistics of IAMOnDo, Kondate, FC and FA datasets: number of documents, strokes and strokes per category

Dataset		Training	Validation	Test
IAMOnDo	Docs	403	200	203
	Strokes	143348	68725	70927
	Text	116801	57821	57959
	Nontext	26547	10904	12968
	Graphic	37496	15481	17488
	Text	79812	39796	40469
	Table	13044	6562	6883
	List	6337	3474	3115
	Math	6659	3412	2972
Kondate	Docs	210	100	359
	Strokes	41190	18525	71846
	Text	34406	15725	61384
	Nontext	6784	2800	10462
FC	Docs	220	28	171
	Strokes	20865	2490	15696
	Text	13130	1430	9629
	Arrow	3416	466	2735
	Data	1208	174	924
	Decision	942	113	696
	Process	1596	232	1131
	Terminator	409	49	446
	Connection	164	26	135
FA	Docs	132	84	84
	Strokes	6792	4059	4125
	Label	3261	2189	2077
	Arrow	2655	1325	1501
	State	454	279	287
	Final state	422	266	260

and attention mechanisms in a relational graph, which is constructed by exploiting temporal and spatial contextual information. We evaluate the effects of different techniques used in the message passing routine, demonstrating the effects of attention mechanisms, spatial edges and the edge update procedure. With this model, we achieve state-of-the-art performance on four online handwritten datasets IAMOnDo, Kondate, FC and FA.

In the future, there are several potential directions that can be explored. In view of the model, to explore more diverse types of context information can benefit the system performance. In view of document analysis, stroke classification can be integrated with other tasks such as text line segmentation in a unified framework using graph network representation.

Acknowledgements This work has been supported in part by the National Key Research and Development Program Grant 2018YFB1005000 and the National Natural Science Foundation of China (NSFC) Grants 61773376 and 61721004.

Compliance with Ethical Standards

Conflict of interest This work has no conflict of interest with any personal or funding parties.

Appendix

Dataset Statistics

This section is supplementary to section 4.1 and presents the statistics of each dataset.

Table 11 Hyperparameters for all experiments

Hyperparameter	IAMOnDo-2	IAMOnDo-5	Kondate	FC	FA
Number of node output features (C')	32	32	32	32	8
Number of edge output features (D')	19	19	19	19	19
Number of attention heads (K)	8	8	8	4	8
Number of attention layers	5	10	10	7	10
Dropout rate	0.2	0.2	0.1	0.1	0.05
Batch size	16	16	16	8	8
Initial learning rate	0.005	0.005	0.005	0.005	0.005
Decay rate	0.1	0.1	0.1	0.1	0.1
Number of patience round	10	20	10	20	20
Temperature	0.5	0.5	0.5	0.5	0.5

Hyperparameters

This section is supplementary to Section 3.4 and presents the chosen hyperparameters for all experiments. For all edge attention layers, the hyperparameters of each layer $((C', D', K)$ are kept the same. We tune the hyperparameters on the validation set by random search.

References

1. Awal AM, Feng G, Mouchere H, Viard-Gaudin C. First experiments on a new online handwritten flowchart database. In: Document Recognition and Retrieval, vol. 7874, p. 78740A. International Society for Optics and Photonics 2011.
2. Bishop CM, Svensen M, Hinton GE. Distinguishing text from graphics in on-line handwritten ink. In: International Conference on Frontiers in Handwriting Recognition, 2004;142–147.
3. Bresler M, Prusa D, Hlavác V. Detection of arrows in on-line sketched diagrams using relative stroke positioning. Winter Conf Appl Comput Vis. 2015;10:610–617.
4. Bresler M, Prusa D, Hlavác V. Online recognition of sketched arrow-connected diagrams. Int J Doc Anal Recogn. 2016;19(3):253–267.
5. Bresler M, Van Phan T, Prusa D, Nakagawa M, Hlavác V. Recognition system for on-line sketched diagrams. In: International Conference on Frontiers in Handwriting Recognition, 2014;563–568 .
6. Carton C, Lemaitre A, Coüasnon B. Fusion of statistical and structural information for flowchart recognition. In: International Conference on Document Analysis and Recognition, 2013;1210–1214.
7. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, 2016;3844–3852.
8. Delaye A, Liu CL. Contextual text/non-text stroke classification in online handwritten notes with conditional random fields. Pattern Recogn. 2014;47(3):959–968.
9. Delaye A, Liu CL. Multi-class segmentation of free-form online documents with tree conditional random fields. Int J Doc Anal Recogn. 2014;17(4):313–329.
10. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics, 2010;249–256.
11. Gong L, Cheng Q. Exploiting edge features for graph neural networks. In: Conference on Computer Vision and Pattern Recognition, 2019;9211–9219.
12. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–1780.
13. Indermühle E. Analysis of digital ink in electronic documents. Ph.D. thesis, University of Bern 2012.
14. Indermühle E, Frinken V, Bunke H. Mode detection in online handwritten documents using blstm neural networks. In: International Conference on Frontiers in Handwriting Recognition, 2012;302–307.
15. Indermühle E, Liwicki M, Bunke H. Iamondo-database: an online handwritten document database with non-uniform contents. In: International Workshop on Document Analysis Systems, 2010;97–104.
16. Jain AK, Namboodiri AM, Subrahmonia J. Structure in on-line documents. In: International Conference on Document Analysis and Recognition, 2001;844–848.
17. Kingma D, Ba J. Adam: A method for stochastic optimization. In: International Conference on Learning Representation 2015.
18. Kipf T, Welling M. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations 2017.
19. Koller D, Friedman N. Probabilistic graphical models: principles and techniques. New York: MIT press; 2009.
20. Lafferty J, McCallum A, Pereira FC. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning, 2001;282–289.
21. Lemaitre A, Mouchère H, Camillerapp J, Coüasnon B. Interest of syntactic knowledge for on-line flowchart recognition. In: International Workshop on Graphics Recognition, pp. 89–98. Springer, 2011.
22. Mochida K, Nakagawa M. Separating figures, mathematical formulas and japanese text from free handwriting in mixed online documents. Int J Pattern Recognit Artif Intell. 2004;18(07):1173–1187.
23. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
24. Peterson EJ, Stahovich TF, Doi E, Alvarado C. Grouping strokes into shapes in hand-drawn diagrams. In: AAAI Conference on Artificial Intelligence, 2010; 974–979.
25. Phan TV, Nakagawa M. Combination of global and local contexts for text/non-text classification in heterogeneous online handwritten documents. Pattern Recogn. 2016;51:112–124.
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
27. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. In: International Conference on Learning Representation 2018.
28. Wang C, Mouchère H, Lemaitre A, Viard-Gaudin C. Online flowchart understanding by combining max-margin markov random field with grammatical analysis. Int J Doc Anal Recogn. 2017;20(2):123–136.
29. Wang C, Mouchere H, Viard-Gaudin C, Jin L. Combined segmentation and recognition of online handwritten diagrams with high order markov random field. In: International Conference on Frontiers in Handwriting Recognition, 2016, pp. 252–257.
30. Wang M, Yu L, Zheng D, Gan Q, Gai Y, Ye Z, Li M, Zhou J, Huang Q, Ma C et al. Deep graph library: Towards efficient and scalable deep learning on graphs. In: International Conference on Learning Representation 2019.
31. Weber M, Liwicki M, Schelske YT, Schoelzel C, Strauß F, Dengel A. Mcs for online mode detection: Evaluation on pen-enabled multi-touch interfaces. In: International Conference on Document Analysis and Recognition, 2011, pp. 957–961.
32. Wu J, Wang C, Zhang L, Rui Y. Offline sketch parsing via shapeness estimation. In: International Joint Conference on Artificial Intelligence, 2015, pp. 1200–1206.
33. Ye JY, Zhang YM, Liu CL. Joint training of conditional random fields and neural networks for stroke classification in online handwritten documents. In: International Conference on Pattern Recognition, 2016, pp. 3264–3269.
34. Ye JY, Zhang YM, Yang Q, Liu CL. Contextual stroke classification in online handwritten documents with graph attention networks. In: International Conference on Document Analysis and Recognition, 2019, pp. 993–998.
35. Zhou XD, Liu CL. Text/non-text ink stroke classification in japanese handwriting based on markov random fields. Int Conf Document Anal Recogn. 2007;1:377–381.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.