CASIA-onDo: A New Database for Online Handwritten Document Analysis

Yu-Ting Yang[1,2][0000−0002−4561−7245], Yan-Ming Zhang[1][0000−0001−6718−5589], Xiao-Long Yun[1,2][0000−0001−9353−2900], Fei Yin[1][0000−0002−6412−9140], and Cheng-Lin Liu[1,2][0000−0002−6743−4175]

[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
{yuting.yang, ymzhang, xiaolong.yun, fyin, liucl}@nlpr.ia.ac.cn

Abstract. In this paper we introduce an online handwritten document database (CASIA-onDo), serving as a standard database for the development and evaluation of methods in the field of online handwritten document layout analysis. It consists of 2,012 documents including a total of 841,159 online strokes. The database, covering Chinese and English languages, was produced by 200 writers. Six types of contents occur in the documents, namely text, formulas, diagrams, tables, figures, and lists. The distribution of different types is close to the actual situation. Benefiting from detailed annotations, CASIA-onDo can support different tasks of layout analysis under online or offline settings. Firstly, based on the semantic level annotation, it can be used for many classification tasks such as text/non-text classification, table/non-table classification, multi-class stroke classification and so on. Secondly, based on the instance level annotation, it can be used for segmentation tasks such as text line separation and formula segmentation. Thirdly, based on the various writing styles, it can be used for handwriting recognition and writer clustering tasks. In addition, we perform preliminary experiments to provide a benchmark on this database with a state-of-the-art method. More techniques can be evaluated on this challenging database in the future.

Keywords: Online handwritten document · Document layout analysis · Stroke classification · Database.

1 Introduction

Nowadays, pen-based and touch-based interfaces are widely used, producing a large number of handwritten documents with mixed objects. Automatic understanding of online freehand documents, aiming to convert handwritten inputs into digital formats, has been an active and challenging research field. There are a great many difficult tasks in the process of document analysis. Firstly, different types of contents need to be separated by stroke classification. Secondly, different objects have to be extracted by instance segmentation. Thirdly, these different instances can be fed to specialized recognition engines, including those for text lines, diagrams, formulas, and tables. The first two tasks are referred as document layout analysis and are the key for an automatic document processing system.

A standard database of online handwritten documents is important to facilitate research in the field. It should be well-constructed and large enough to provide reliable evaluation metrics for different tasks. Previously released online handwriting databases include text databases (UNIPEN [1], IRONOFF [2], IAMonDB [3], HANDS-Kondate [4], SCUT-COUCH [5] and CASIA-OLHWDB [6]), diagram databases (FC_A [7], FC_B [8], FC [9] and CASIA-OHFC [10]), math expression databases (CROHME [11,12,13]). Since these databases were designed for specific recognition tasks, they typically contain data of one type and thus cannot be used for document layout analysis. The only databases that cover entire document objects are Kondate [14] and IAMonDo [15]. However, Kondate and IAMonDo databases only have small number of documents and simple layout structures, and are also limited in content types and language types. Therefore, it is essential to have a more comprehensive database to foster the research of document layout analysis.

In this paper, we release the CASIA-onDo database, which is the largest online handwritten multi-contents document database so far, with detailed annotation information and complex structure. All this data will be freely public to the academic community and released with this paper[3]. The database consists of 2012 documents containing handwritten text, figures, diagrams, formulas, tables and lists arranged in an unconstrained way. The database is designed primarily for the development of algorithms for document layout analysis. Semantic level and instance level annotations are provided. With semantic level annotations, the database can be applied to semantic segmentation tasks such as multi-class stroke classification, text/non-text classification and table/non-table classification. With instance level annotations, it is suitable for instance segmentation tasks such as text line separation and formula segmentation. It is potentially useful for text recognition, formula recognition, table analysis and diagram recognition by adding corresponding annotations in the future. We also perform multiple semantic segmentation experiments on the new database based on EGAT [22] model, and provide initial experimental results as a baseline.

The rest of this paper is organized as follows. Section 2 reviews related works on handwritten document databases and methods on layout analysis research. Section 3 elaborates the design of the CASIA-onDo database. Section 4 presents the statistics and usage of the database. Section 5 presents our experimental results with a novel automatic stroke feature extraction method. Section 6 draws conclusions.

[3] http://www.nlpr.ia.ac.cn/databases/CASIA-onDo/index.html

2 Related Work

This part mainly discusses databases and approaches for online handwritten document layout analysis, and so we summarize the related works from these two aspects.

Most previous databases of online handwriting focused on handwritten text only, such as UNIPEN [1], IRONOFF [2], CASIA-OLHWDB [6] and IAMonDB [3]. Databases containing different content types are Kondate [14] and IAMonDo [15]. IAMonDo is a collection of handwritten English online documents. It consists of 1000 documents, mixing text, diagrams, formulas, tables, figures and lists. Kondate is made up of 669 freehand Japanese online documents and contains text, formula, figure, ruled line and editing mask. Up to now, there is no database that contains Chinese documents. What'more, the layout and the contents in these databases are simple, for example, there is no in-line formulas and complex tables.

Tasks in document analysis are traditionally modeled as structured prediction problems, which can be divided into three mainstream branches: conditional random fields (CRF)[16,17], recurrent neural networks (RNN)[18,19], and graph neural networks (GNN) [21,22,23]. Among them, GNN is most powerful in modeling complex structure. Ye *et al.* [21,22] formulated stroke classification problem as node classification problem in the relational graph. For multi-class classification, EGAT [22] achieves 95.81% on IAMonDo database. Further, the promoted network [23] performed node clustering and node classification jointly to solve the text line grouping and stroke classification problem.

3 Database Design

3.1 Types of Contents

We suppose that the online handwritten documents are mainly created in the context of note taking in the class. Six different content types are considered when producing this database. Details can be seen from the following list:

Text block: Text blocks are paragraphs or text lines composed of Chinese or English characters. Note that text placed in diagrams, lists and tables does NOT fall into this category.

Formula: Formulas are math expressions composed of numbers, variables, operators and functions. Formula may have 2-dimensional structures such as subscripts or superscripts, and can appear in text lines, in lists or in diagrams and can be on a single line or multiple lines.

Diagram: Diagrams are flowcharts composed of symbols, connecting lines and text. They are commonly used in documents to represent algorithms and workflows.

Table: Table consists of table lines and content in the table units. They can be regular or complex. Regular tables have aligned table lines, while complex tables may have invisible table lines or unaligned structures.

List: Lists are featured by a sign at the beginning of each term, for example numbers, letters, asterisk, etc. Items in a list can be placed horizontally or vertically.

Figure: Figures include objects such as circuit diagrams in Physics, line charts in mathematics, freehand sketches, etc.

3.2 Templates of Contents

To control the distribution of different types of contents, we make templates for contributors to copy. Plenty of text paragraphs, lists, formulas, figures, tables and diagrams are collected from the Internet, and then mixed together to form templates. We made a total of 1000 templates. Considering the real situation of notes in science and engineering, we insert a large number of formulas and tables into the templates. The figures are mainly obtained from the graphs that commonly appear in mathematics and physics, such as circuit diagrams and line charts. The list is used to simulate the options in multiple-choice questions.

When generating the templates, some criteria must be applied to guarantee the desired distribution, which are summarized as follow:

Standardization The distribution of content types in the templates should meet with the reality as much as possible.

Diversity The template should cover both Chinese and English and have different layouts. Each type should have various representations and scales.

Complexity This is mainly reflected in the diverse layout, inline formula and tables without ruling.

More specifically, the quantitative rules which are applied during generation of one template are listed below:

- A template contains at least 3 content types.
- In all texts, the ratio of Chinese to English is 8:2.
- With equal probability, either a random table, or a random figure or a random list is added.
- With equal probability, the table is without ruling, with just horizontal ruling, or with a fully ruled grid.
- The list contains 2-7 items.
- All contents can be in random direction.

3.3 Data Acquisition

In order to cover diverse writing styles as far as possible, we asked 200 writers to copy the documents. Generally, each individual template was copied by 2 different writers, and each writer drew 10 different documents. While 12 sheets were copied by three writers, so there are 2012 documents in total. We compare the statistic of CASIA-onDo with aforementioned document databases in Table 1.

We used Huawei tablets with stylus to collect the documents. It is accurate and can deliver time and pressure information besides the coordinates of the digital ink. Every stroke corresponds to a sequence of 5-dimensional points which

Table 1: Online handwritten document databases overview.

Database	Classes	Partition	Writers	Templates	Documents	Strokes
Kondate [14]	3	Train	67	-	210	41190
		Valid		-	100	18525
		Test		-	359	71846
IAMonDo [15]	5	Train	200	400	400	141421
		Valid		200	200	68725
		Test		212	212	70927
CASIA-onDo	6	Train	200	700	1400	588884
		Valid		100	200	81693
		Test		200	412	170582

contain the information of (x, y) coordinates, time, pressure and the state of pen tip.

In the instructions, the writers are allowed to rearrange the content and keep their own sketchy styles. There are no further constraints and supervision on how to create documents on them. This is done with the purpose of reflecting the workflow in a realistic context. Fig. 1 shows how different writers copied the content of a template to the document.

3.4 Annotation

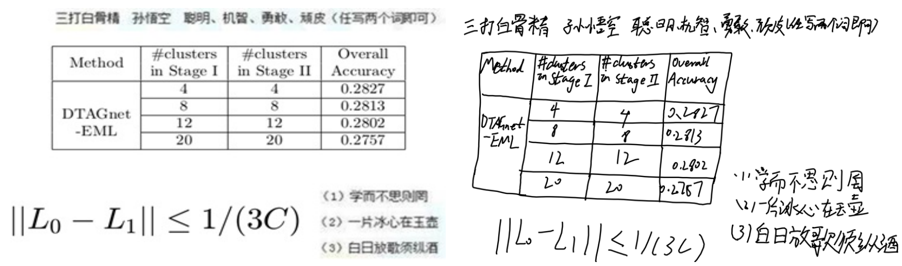
We provide two levels of annotations for each stroke: the semantic class and instance ID of its associate symbol. The contents of texts and formulas have not been provided.

For semantic annotation, we provide 11 semantic labels in total rather than 6 labels listed in Sec. 3.1, so as to facilitate multiple usage of the dataset. In particular, the formula class is divided into four subclasses: in-line formula, inter-line formula, in-list formula and in-diagram formula. The diagram class is divided into symbol within diagram, text within diagram, and formula within diagram. The table class is divided into table line and text within the table.

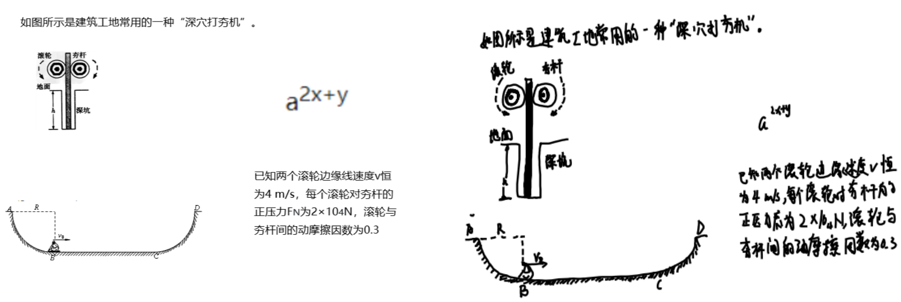
In annotating formulas, isolated numbers or variables (such as $8, a$) are not marked as formulas, while math expressions with 2-dim layout (such as $8^2, a_2$) and special mathematical symbols (such as $\pi, \sum, \infty$) are marked as formulas regardless of their length. In annotating lists, as long as there is a symbol in front of each item, regardless of the length and number, it is marked as a list. For easy viewing, each annotation is given a symbol as shown in Table 2. The colored annotation of document is shown in Fig. 2.

For instance level annotation, each text line is treated as an instance with a unique ID. For the remaining five categories, different entities have different IDs.

We choose the InkML(*Ink Markup Language*) [20] language to store the documents, which can represent information flexibly. This is mainly achieved through two elements. The first one is the **trace** element. This XML tag represents a stroke s. It contains a stroke ID and a sequence of trajectory points with a



（a）**A template given to writer A, and the document created by the writer.**



（b）**A template given to writer B, and the document created by the writer.**

Fig. 1: Templates and corresponding handwritten documents.

Table 2: Annotations and corresponding symbols.

Annotation	Symbol	Annotation	Symbol
TEXT	A	LINEOUT_MATH_EXPRESSION	G
LIST	B	LINEIN_MATH_EXPRESSION	H
LISTIN_MATH_EXPRESSION	C	TABLE	I
DIAGRAM	D	TABLEIN_TEXT	J
DIAGRAMIN_TEXT	E	FIGURE	K
DIAGRAMIN_MATH_EXPRESSION	F		

variable-length m.

$$s = \{[x_1, y_1, p_1, s_1, t_1], [x_2, y_2, p_2, s_2, t_2], \cdots, [x_m, y_m, p_m, s_m, t_m]\}, \tag{1}$$

where x_i and y_i mean xy-coordinates, p_i is the pressure on the pen tip, s_i indicates the pen state (down or up) of the i-th point and t_i is time. The second one is the **traceGroup** element, which records a collection of strokes belonging to the same category. The label exists in its child element **annotationXML** in the format of "label_id", where "label" represents which content type the group of strokes belong to, and "id" is used to distinguish different instances.

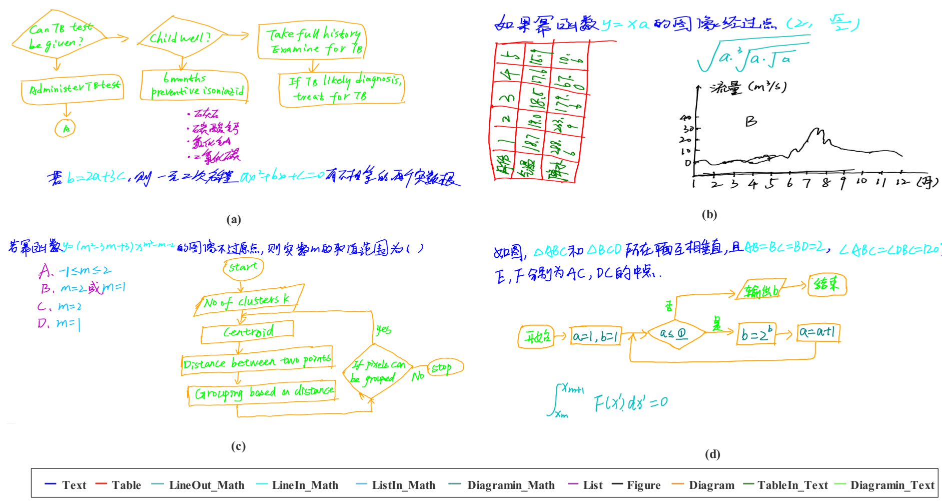


Fig. 2: Visualization of colored annotation.

4 Recommendations of Usage

The CASIA-onDo database has some favorable merits: (1) It records rich online information; (2) It involves various content types; (3) It has both semantic and instance level annotations; (4) The documents are stored writer by writer.

For using the database for research, we recommend standard partitioning into training, validation and test sets, and propose some research scenarios.

4.1 Data Structure and Statistics

The documents are randomly divided into three disjoint subsets——a training set, a validation set and a test set at the ratio of 7:1:2. Therefore, there are 1400/200/412 documents in the training/validation/test set, respectively. The statistics details of the CASIA-onDo dataset are shown in Table 3.

The database consists of 2,012 documents with 841159 individual online strokes. The entire database contains 5046 text lines, 1180 tables, 1254 lists, 745 diagrams, 1328 figures and 7406 formulas.

4.2 Research Scenarios

Based on the CASIA-onDo database, some typical research tasks of handwritten document analysis can be performed. Our recommendations are as follows.

1) *Stroke classification*: Our database is made up of a large number of strokes, grouped into six categories. For different levels of classification, user can use the annotations from Table 2 to generate appropriate labels neatly. The corresponding relationship between annotations and labels is shown in Table 4.

Table 3: Statistics of strokes and instances in the CASIA-onDo database.

	Train		Valid		Test	
	strokes	instances	strokes	instances	strokes	instances
Text	200924	3501	29452	531	56844	1014
Formula	58787	5154	7729	735	17677	1517
Diagram	66944	497	11672	88	22794	160
Table	144876	837	18049	113	38376	230
Figure	38518	937	4881	121	11397	270
List	78835	879	9910	116	23494	259
Total	588884	11805	81693	1704	170582	3450

2) *Text separation and formula segmentation*: Our database contains 5046 text lines and 7406 formulas in total. What's more, all the instances are separated in the ground-truth. It is convenient for training and evaluating text line segmentation and formula segmentation algorithms.

3) *Writer identification*: In our database, all the documents are stored in writer-specific files and each writer has 10 handwritten pages. We can perform experiments to judge whether two documents are from the same writer or not (writer verification) or classify a page to a nearest reference page of known writer (writer identification).

Table 4: Annotations according to different levels of classification.

Label / Symbol / Task	A	B	C	D	E	F	G	H	I	J	K
Text/non-Text	1	1	0	0	1	0	0	0	0	1	0
Table/non-Table	0	0	0	0	0	0	0	0	1	1	0
Diagram/non-Diagram	0	0	0	1	1	1	0	0	0	0	0
Formula/non-Formula	0	0	1	0	0	1	1	1	0	0	0
List/non-List	0	1	1	0	0	0	0	0	0	0	0
Multi-class	0	1	3	2	2	3	3	3	4	4	5

5 Experimental Evaluation

To validate the functionalities of our newly created CASIA-onDo database, we systematically evaluate a state-of-the-art method EGAT [22] based on both hand-crafted features and automatic features as baselines on different tasks from Table 4. Furthermore, we introduce a novel automatic stroke feature extraction method, which will be described in detail.

5.1 Stroke Classification Framework

EGAT [22] is a method that models the stroke classification problem as node classification in a document graph based on graph attention networks. It contains the following three modules:

Graph construction EGAT formalizes a document into a relational graph, in which nodes represent strokes and edges represent interaction between strokes. There are two kinds of contextual information in the graph, namely spatial context and temporal context. Spatially adjacent strokes are more likely to have the same labels. Intuitively, the spatial neighbors are found by selecting the nearest k_s strokes in the document by k nearest search. Temporal adjacent strokes captures the sequence of writing. Temporal neighbors are defined as the strokes whose temporal distance is less than k_t from the current stroke. Here, both k_s and k_t are hyperparameters.

Feature extraction In the original EGAT, 13 contour-based shape features and 10 local context features are extracted from each stroke as node features, and 37 pairwise relation features are extracted from two connected nodes (strokes) as edge features. Since hand-crafted features may suffer from limited descriptive ability, in this work, we come up with an automatic stroke feature extraction method with three steps.

Preprocessing. Given a stroke in Equation 1, to depict the direction of writing trajectory at each time step i, the original point (x_i, y_i, s_i) is processed into a 9-dimensional vector by calculating the first- and second-order derivatives:

$$[x_i, y_i, \Delta x_i, \Delta y_i, \Delta' x_i, \Delta' y_i, s_i^1, s_i^2, s_i^3], \tag{2}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$, $\Delta' x_i = x_{i+2} - x_i$, $\Delta' y_i = y_{i+2} - y_i$. The last three values $[s_i^1, s_i^2, s_i^3]$ is the one-hot encoding of the pen state s_i. [1,0,0], [0,1,0] and [0,0,1] represent pen-down, ongoing and pen-up, respectively.

Extracting node feature. The stroke made up of encoded points is then input into a stroke feature extraction network based on LSTM. The architecture of the network is shown in Fig. 3. Multiple bi-directional LSTM layers are stacked and each layer has forward and backward units. The vectors from the last LSTM layer is passed through a max pooing layer and a fully connected layer.

Extracting edge feature. We design a method to capture the spatial relation between two strokes. The trajectories of two strokes are resampled to equal length adopting the equal interval interpolation method. Then the coordinates of sampling points between two strokes are subtracted as the original edge feature.

Edge graph attention network EGAT is a graph neural network which improves the classic graph attention network model [21] with a novel attention mechanism. Basically, EGAT consists of several stacked edge graph attention layers. A set of stroke and edge features and a set of edges are fed into the first layer. Then the representation of strokes is updated by exploiting temporal and spatial contextual information from the neighborhood by graph convolution with attention mechanisms. Please refer to the original paper [22] for more details.

5.2 Experimental Setup

In graph construction, k_t and k_s are both set to 8. For stroke feature extraction, the bi-directional LSTM consists of 3 layers. Each layer has 50 forward and

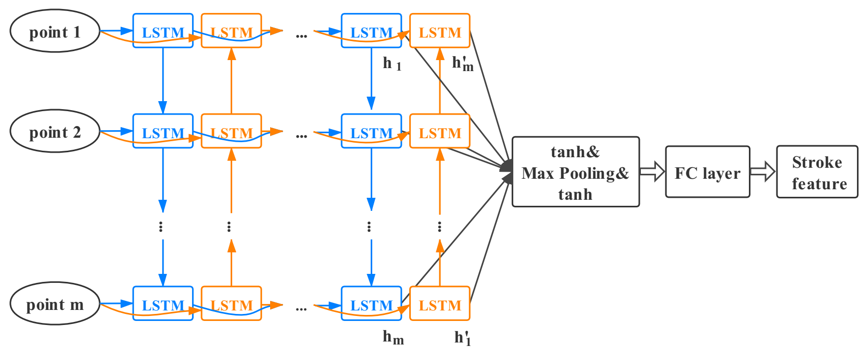


Fig. 3: Stacked BiLSTM for stroke feature extraction.

50 backward LSTM units, so the stroke feature is a 100-dimensional vector. Xavier Normalization is adopted to initialize the parameters. For edge feature extraction, the number of sampling points is 20, so the dimension of edge feature is 40. EGAT employs 3 stacked attention layers. The dimension in hidden layer is set to 32 and the dimension of the output features is equal to the number of categories. 8 attention heads are utilized for concatenation.

When using hand-crafted feature, we utilize Adam optimizer to train the model and batch size is set to 16. For the end-to-end model, the AdamW optimizer is chosen instead and batch size is 4. In both cases, the initial learning rate α is set to 0.001. The model is trained for at most 500 epochs, and early stopping is performed when validation loss does not decrease for ten consecutive epochs. We implement the method with the DGL library and its PyTorch backend. All experiments are conducted with a TITAN RTX GPU.

5.3 Evaluation Metrics

We evaluate EGAT on two-class classification and multi-class classification problems defined in Table 4. For two-class classification, the accuracy is defined as:

$$Accuracy = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \sigma(\hat{y}_{it} = y_{it})}{\sum_{i=1}^{N} T_i} \tag{3}$$

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

For multi-class classification, the accuracy for each class c is defined as:

$$Accuracy[c] = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \sigma(y_{it} = c)\sigma(\hat{y}_{it} = y_{it})}{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \sigma(y_{it} = c)} \tag{4}$$

5.4 Results and Analysis

1) *Quantitative Results*: Table 5 and Table 6 show two-class and multi-class stroke classification accuracy on CASIA-onDo, respectively. As it can be seen from

the tables, multi-class classification is much harder than two-class classification. In addition, our proposed automatic features achieve better performance than hand-crafted features.

Table 5: Two-class stroke classification accuracy on CASIA-onDo (%).

Task	Hand-crafted feature			Automatic feature		
	Positive	Negative	Total	Positive	Negative	Total
Text/non-Text classification	98.53	88.28	96.12	98.55	90.76	96.72
Table/non-Table classification	94.34	98.67	97.69	90.31	98.97	97.02
Diagram/non-Diagram classification	92.34	99.29	98.36	96.94	98.91	98.65
Formula/non-Formula classification	86.08	98.52	97.08	84.04	98.88	97.16
List/non-List classification	69.51	97.71	93.30	75.73	97.29	93.91

Table 6: Multi-class stroke classification accuracy on CASIA-onDo (%).

Feature	Text	Formula	Diagram	Table	Figure	List	Total
Hand-crafted	89.59	76.94	98.17	94.50	91.26	76.13	88.79
Automatic	93.04	89.43	96.07	94.35	91.17	68.55	89.87

	Text	**Formula**	**Diagram**	**Table**	**Figure**	**List**
Text	93.04	0.16	0.21	1.04	0.20	3.92
Formula	7.42	89.43	0.08	0.54	0.43	2.09
Diagram	0.79	0.50	96.07	0.91	1.29	0.43
Table	3.41	0.33	0.42	94.35	0.52	0.96
Figure	2.75	1.03	1.86	2.06	91.17	1.13
List	26.76	3.11	0.20	1.14	0.24	68.55

Fig. 4: The confusion matrix of end-to-end stroke classification result (%).

2) *Qualitative Results*: In Fig. 5, we show some visual results of end-to-end stroke classification on CASIA-onDo, where each color corresponds to a content type. According to statistics, most of the samples enjoy good results.

3) *Failure Analysis*: There are two difficulties of stroke classification on CASIA-onDo database, one is in-line formula, the other is list. In-line formulas are interspersed between text, which is common in class notes. It is easy to confuse the beginning and end of the formula with the surrounding text, resulting in errors. Lists, specially long ones, are also easily confused with text lines, indicating that the model does not learn the intrinsic feature of lists well. We show the confusion matrix in Fig. 4 and some failure cases in Fig. 6.

6 Conclusion and Outlook

This paper describes a database of online handwritten documents containing text blocks, diagrams, formulas, tables, figures and lists. The documents are

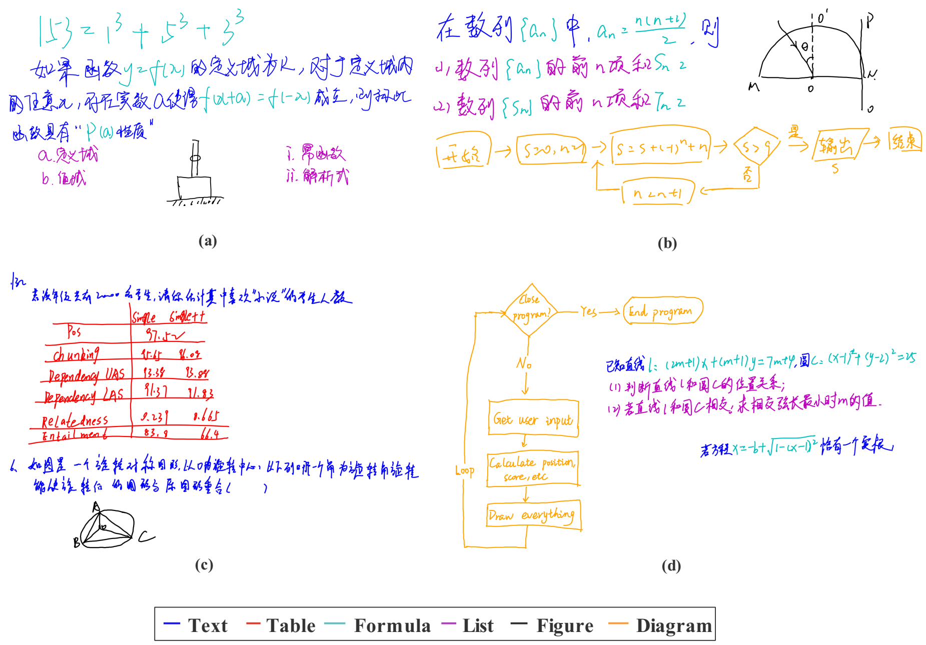


Fig. 5: Visualization of multi-class stroke classification results.

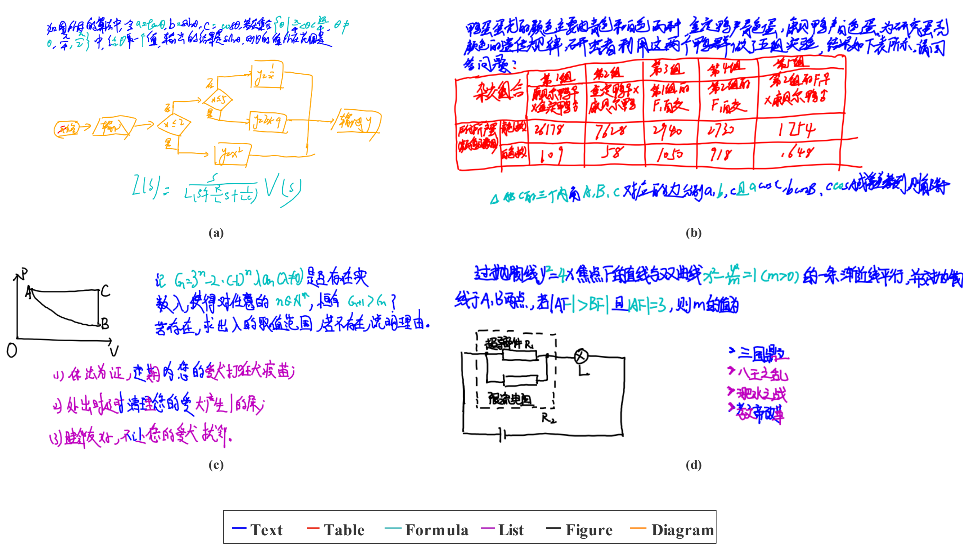


Fig. 6: Some typical examples of misclassification.

written in an unconstrained manner and reflect documents generated in a realistic context. With 200 different writers, many different styles in writing and drawing are included. The distribution of the content types is kept stable over all individual documents. The database can serve as a basis for tasks such as document layout analysis, text line segmentation, formula detection, as well as handwriting recognition, etc.

We evaluate a document layout analysis method EGAT on CASIA-onDo. The results serve as a baseline for future methods to be developed on this database. To overcome the shortcomings of hand-crafted features, we propose an automatic feature extraction method and conduct contrast experiments to demonstrate the efficiency.

There are still many challenging problems need to be further explored, such as precise distinction between inline formula and text and distinction between list and text. These issues are of great research value and important application prospects.

Acknowledgement

This work has been supported by the National Key Research and Development Program under Grand No. 2020AAA0109700 and the National Natural Science Foundation of China (NSFC) under Grant No. 61773376.

References

1. Guyon, I., Schomaker, L., Plamondon, R., et al.: UNIPEN project of on-line data exchange and recognizer benchmarks. In: 12th International Conference on Pattern Recognition, vol. 2, pp. 29–33. IEEE, New York (1994).
2. Viard-Gaudin, C., Lallican, P. M., Knerr, S., et al.: The ireste on/off (ironoff) dual handwriting database. In: 5th International Conference on Document Analysis and Recognition, pp. 455–458. IEEE, New York, USA (1999).
3. Liwicki, M., Bunke, H.: IAM-OnDB-an on-line English sentence database acquired from handwritten text on a whiteboard. In: 8th International Conference on Document Analysis and Recognition, pp. 956–961. IEEE, New York, USA (2005).
4. Nakagawa, M., Onuma, M.: Online handwritten Japanese text recognition free from constrains on line direction and character orientation. In: 7-th International Conference on Document Analysis and Recognition, pp. 519–523. IEEE, New York, USA (2003).
5. Jin, L., Gao, Y., Liu, G., et al.: SCUT-COUCH2009—a comprehensive online unconstrained Chinese handwriting database and benchmark evaluation. International Journal on Document Analysis and Recognition **14**(1), 53–64 (2011)
6. Liu, C. L., Yin, F., Wang, D. H., et al.: CASIA online and offline Chinese handwriting databases. In: International Conference on Document Analysis and Recognition, pp. 37–41. IEEE, (2011)
7. Awal, A. M., Feng, G., et al.: First experiments on a new online handwritten flowchart database. In: Document Recognition and Retrieval XVIII, vol. 7874, p. 78740A. International Society for Optics and Photonics, USA (2011)

8. Bresler, M., Průša, D., Hlaváč, V.: Online recognition of sketched arrow-connected diagrams. International Journal on Document Analysis and Recognition **19**(3), 253–267 (2016)
9. Bresler, M., Van Phan, T., Prusa, D., et al.: Recognition system for on-line sketched diagrams. In: 14th International Conference on Frontiers in Handwriting Recognition, pp. 563–568. IEEE, New York, USA (2014).
10. Yun, X. L., Zhang, Y. M., Yin, F., et al.: Instance GNN: A Learning Framework for Joint Symbol Segmentation and Recognition in Online Handwritten Diagrams. IEEE Transactions on Multimedia, (2021).
11. Mouchere, H., Zanibbi, R., Garain, U., et al.: Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014. International Journal on Document Analysis and Recognition **19**(2), 173–189 (2016)
12. Mouchere, H., Viard-Gaudin, C., Zanibbi, R., et al.: ICFHR2014 competition on recognition of on-line handwritten mathematical expressions. In: 14th International Conference on Frontiers in Handwriting Recognition, pp. 791–796. IEEE, New York, USA (2014).
13. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., et al.: ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions. In: 15th International Conference on Frontiers in Handwriting Recognition, pp. 607–612. IEEE, New York, USA (2016)
14. Mochida, K., Nakagawa, M.: Separating figures, mathematical formulas and japanese text from free handwriting in mixed online documents. International Journal of Pattern Recognition and Artificial Intelligence, **18**(07), 1173–1187 (2004).
15. Indermühle, E., Liwicki, M., Bunke, H.: IAMonDo-database: an online handwritten document database with non-uniform contents. In: 9th International Workshop on Document Analysis Systems, pp. 97–104. Association for Computing Machinery, New York, USA (2010).
16. Delaye, A., Liu, C. L.: Contextual text/non-text stroke classification in online handwritten notes with conditional random fields, Pattern Recognition, **47**(3), 959–968 (2014).
17. Ye, J. Y., Zhang, Y. M., Liu, C. L.: Joint training of conditional random fields and neural networks for stroke classification in online handwritten document. In: International Conference on Pattern Recognition, pp. 3264–3269. IEEE, (2016)
18. Indermühle, E., Frinken, V., Bunke, H.: Mode detection in online handwritten documents using BLSTM neural networks. In: International Conference on Frontiers in Handwriting Recognition, pp. 302–307. IEEE, (2012)
19. Van, Phan. T., Nakagawa, M.: Combination of global and local contexts for text/non-text classification in heterogeneous online handwritten documents, Pattern Recognition, 112–124 (2016)
20. Watt, S. M., Underhill, T., Chee, Y. M., et al.: Ink markup language (InkML). W3C Proposed Recommendation, 10 (2011).
21. Ye, J. Y., Zhang, Y. M., Yang, Q., et al.: Contextual stroke classification in online handwritten documents with graph attention networks. In: International Conference on Document Analysis and Recognition, pp. 993–998. IEEE, New York, USA (2019)
22. Ye, J. Y., Zhang, Y. M., Yang, Q., et al.: Contextual stroke classification in online handwritten documents with edge graph attention networks. SN Computer Science, **1**, 1–13(2020).
23. Ye, J. Y., Zhang, Y. M., Yang, Q., et al.: Joint stroke classification and text line grouping in online handwritten documents with edge pooling attention networks. Pattern Recognition, (2021).