Expert Systems With Applications 172 (2021) 114588

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Sense the pen: Classification of online handwritten sequences (text, mathematical expression, plot/graph)☆

Junaid Younas [a,b,*], Muhammad Imran Malik [c,d], Sheraz Ahmed [b], Faisal Shafait [c,d], Paul Lukowicz [a,b]

[a] University of Kaiserslautern, Germany
[b] German Research Institute for Artificial Intelligence, Kaiserslautern, Germany
[c] School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), H-12 Islamabad, Pakistan
[d] National Centre of Artificial Intelligence, National University of Sciences and Technology, Islamabad, Pakistan

ARTICLE INFO

Keywords:
Digital pen
Digital screen
Online handwriting
Classification
Machine-learning
Database

ABSTRACT

This paper has a threefold contribution. First, it presents a novel online handwriting database captured using a digital/sensor pen (Apple pencil) and digital/sensor screen (iPad). The captured data are continuous streams of multi-dimensional points, analyzed and processed to classify handwritten sequences into plain text, mathematical expressions, and plots/graphs. Second, a new feature set for online handwritten sequence classification is proposed. The said feature set is used to establish a benchmark for the proposed dataset using various machine-learning classifiers. Third, an ablation study is performed to look into the performance of the proposed feature set compared to the existing feature sets. Here, the proposed feature set has outperformed all the existing feature sets in various evaluation metrics. Furthermore, the proposed dataset and the feature set have been made publicly available along with the benchmark evaluation to enable further research in the field.

1. Introduction

We are living in a digital age where different sensors or sensor-based devices surround us. Display and sensor technologies, when combined, provide new ways for users to interact with their surroundings. For handwriting, in particular, the increasing influence of digital devices (e.g., digital ink and mobile devices like iPad, tablets, etc.) has attracted the research community's attention (Twyman & Heward, 2018; Yang et al., 2018; Koile & Singer, 2006; Dunn & Sweeney, 2018; Hulls, 2005; Griechisch et al., 2014). Use of these devices paved the way for writing-behavior analysis like handwriting classification (classifying the handwritten samples/sequences into text, graphics, or formulas, etc.) (Delaye & Lee, 2015; Bahlmann et al., 2002; Younas et al., 2018; Indermuhle et al., 2012), handwriting recognition (recognizing what is written - optical character recognition) (Liwicki & Bunke, 2006, 2009; Mandal et al., 2018), and writer identification(identifying the writer of handwritten text/signature) (Kholmatov & Yanikoglu, 2005; Liwicki et al., 2006; Said et al., 1998; Sesa-Nogueras et al., 2016).

Handwriting is broadly categorized as (a) offline and (b) online. Offline handwriting is usually produced on paper with a traditional pen. In offline handwriting, temporal information cannot be tracked as automatic systems are provided with photographs of handwriting alone (only spatial information). Whereas writing on digital displays or writing with digital pens on special/ordinary paper is termed as online handwriting. In online handwriting, pen movements are recorded as a continuous stream of points, and therefore, temporal information about handwriting is also available with the spatial information. Note that every person has its specific writing style, which may vary when writing different modalities, like plain text, or mathematical expression, or plots/graphs, etc. This makes handwriting classification, whether online or offline, quite challenging and interesting for the research community.

Handwriting classification is important from a historical perspective as well. During the first half of the 20th century, handwriting classification systems were developed as a bio-metric tool. For example, Milwauke police (B.Livingston, 1959) and Nottingham police (Moore, 1945) adopted handwriting classification systems to help in criminal investigation and to keep records of citizens. Furthermore, German police used handwriting as a bio-metric feature during the second world

☆ This work was supported by Higher Education Commission (HEC) of Pakistan and Forschungsinitiative under the project "AI Enhanced Cognition and Learning".
* Corresponding author at: University of Kaiserslautern, Germany.
E-mail address: junaid.younas@dfki.de (J. Younas).

https://doi.org/10.1016/j.eswa.2021.114588
Received 4 February 2020; Received in revised form 5 January 2021; Accepted 5 January 2021
Available online 12 January 2021
0957-4174/

war (Moore, 1945). Till the 90′s, mostly handwritten templates were used for writing classification and writer identification using predefined feature sets (Smith, 1954; B.Livingston, 1959). Later, traditional handwriting devices coupled with sensors broadened the research scope, particularly for online handwriting, and commercial systems for online handwriting analysis (classification and recognition) were reported (Younas et al., 2018; Delaye & Lee, 2015; Schelske et al., 2011). Today, handwriting classification systems find applications in education, banking, postal services, and forensic science. For example, online handwriting classification systems serve as a basis to analyze the performance of students while attempting solutions to different tasks (writing mathematical expressions, or plain text, or plotting graphs) (Twyman & Heward, 2018; Yang et al., 2018; Koile & Singer, 2006; Dunn & Sweeney, 2018; Hulls, 2005). Similarly, in the banking sector and forensic science, automatic handwriting classification systems can facilitate segmentation/extraction of different modalities (like handwritten text or mathematical expressions) from documents, which could further facilitate experts in performing document verification (Malik et al., 2013; Sharma & Sundaram, 2017; Diaz et al., 2019; Malik & Liwicki, 2012). Moreover, handwriting classification can serve as an important step to improve the performance of handwriting recognition systems by classifying data first and then passing it to handwriting recognition systems.

In this paper, we present a feature set comprised of 49 features for online handwritten sequence classification. The presented feature set, to the best of the authors' knowledge, hasn't been used for the task of handwriting classification previously. We also demonstrate its significance in developing a novel approach to classify handwritten sequences in plain text, mathematical expressions, and plots/graphs. Furthermore, we present a novel database for online handwriting classification and recognition recorded by sensor pen (i.e., Apple pencil) and touch screen display (i.e., iPad). The collected database consists of 12,139 sequences of plain text, mathematical expressions, and plots/graphs. We benchmark the proposed dataset for online handwritten sequence classification by reporting results on well-known machine-learning algorithms. We note that a few datasets, originally collected for online handwriting recognition, have been used for handwriting classification (Indermuhle et al., 2010; Liwicki & Bunke, 2006). These datasets, however, are either better suited for recognition, e.g., IM-OnDB - collected as handwritten notes of English text on a whiteboard for mode detection, i.e., identifying handwritten strokes at every point in document creation (e.g., IM-OnDo). In comparison the dataset presented in this paper contains handwritten sequences that could be readily used for online classification of handwritten modalities into text, mathematical expressions, and/or plots/graphs.

The rest of the paper is structured as follows. Section-II presents an overview of recent work in the online handwriting classification domain. In section-III, precise information about the presented feature set and used machine-learning classifiers are covered. Section-IV covers the methodology of collecting the database and presents salient features of data, followed by evaluation protocol and parameters to tune the machine-learning classifiers for optimal performance. Results are furnished and discussed in section-V, which also covers the strengths and weaknesses of the presented feature set and a comparison with the existing feature sets. Finally, section-VI concludes the paper and provides some hints for future research in this direction.

2. Related work

In this section, we present a historical perspective and a detailed overview of the feature sets available for the classification of online handwritten data. We also cover publicly available datasets for online handwritten documents. In 1954, Smith et al. (Smith, 1954) presented a feature set to classify handwriting for the very first time. Their presented feature set consisted of six features: speed, size, slant, and spacing as developed factors, pressure and form (defined as idiosyncrasies of handwriting) as unconscious behaviors. In 1959, Livingston et al. (B. Livingston, 1959) presented a handwriting and pen-printing classification system to identify law violators. They highlighted 12 factors of printed style lettering done with pencils, pens, or other writing instruments, which can be used to classify an individual's handwriting.

Bouletreau et al. (1997) presented a new family of synthetic parameters based on the fractal behavior of writing for handwriting analysis. They presented four fractal parameters for the classification of writing into different families. They termed their approach as a preliminary step in the process of handwriting classification. Delaye & Lee (2015) presented a flexible framework to segment online handwritten documents, i.e., text-lines, non-text objects, and mathematical symbols. Their presented approach is based on single-linkage agglomerate clustering built upon a feature set for pairwise distance definition. They also present a combination of features to improve online handwritten document segmentation.

Writer identification and classification based on writing styles are presented by Schomaker et al. (1994). Pen-tip displacement data is recorded and then, based on velocity, is segmented into strokes. Each stroke is represented by a 1-d feature vector. These feature vectors are used to train the Kohonen network to classify writers. They used discriminant analysis and clustering techniques to classify writing styles into different families.

Bahlmann et al. (2002) present a new Gaussian dynamic time warping kernel (GDTW) by combining support vector machines (SVMs) and dynamic time warping (DTW) for the classification of online handwriting. Their presented approach creates class boundaries based on discrimination rather than relying on modeling assumptions. They evaluated their approach on the UNIPEN handwriting data. Ahmad et al. (2004) presented the development of a hybrid model for online handwriting classification. Their system can classifying digits, lowercase letters, and uppercase letters using SVMs with RBF kernel. They reported results on UNIPEN and IRONOFF handwriting datasets.

With the publication of IAM-onDo dataset (Indermuhle et al., 2010), text and non-text classification tasks got the attention of the research community, as this database contains contents with formal and informal text, diagrams, tables, drawings, and figures. Delaye & Liu (2013) presented an automatic page segmentation method to extract graphical objects from online documents. Their method is based on hierarchical conditional random fields (CRFs). Delaye & Liu (2014) presented a text/non-text classification system based on CRFs for online handwritten documents. They first calculated the CRFs for text and non-text stroke labeling, then integrated context information to improve the classification results. Phan & Nakagawa (2014) presented a deep-learning based classifier for classifying online documents into text and non-text parts. They used recurrent neural networks (RNNs) and long short term memory (LSTM) networks to evaluate their system on Japanese ink documents database Kondate and IAM-OnDo database.

Weber et al. (2011) presented a system to classify ink traces into either text or graphics for mode detection. They also presented a set of features for online handwriting classification and recognition tasks. They benchmark their presented feature set using machine-learning classifiers on the IAM-OnDo database. Indermuhle et al. (2012) presented a BLSTM based neural network approach for text and non-text stroke detection. Individual strokes are transformed into feature vectors, used to train and test the presented model. They also reported results on the IAM-OnDo database.

Younas et al. (2018) presented an approach for the classification of plain text, mathematical expressions, and graphs. Their presented approach is based on feature-engineering and an ensemble of basic machine-learning classifiers. They used a distance-based threshold to segment raw data into sequences. Every sequence is then transformed to a 26-(dimensional) feature vector before passing it to the classifier. They used the ensemble method to generate the final classification result.

3. Methodology

In this section, we present a detailed overview of data collection, which later contributed towards the compilation of the dataset. Feature extractors are used to transform input sequences into feature vectors, which are then used to train the machine-learning classifiers. An overview of the proposed method is shown in Fig. 1. Details of classifiers along with subsequent modules used are given in the following subsections.

3.1. Data collection and Pre-processing

The data collection process starts with an iOS-based application for iPads, which provides the functions for creating new documents, storing data, managing, and exporting existing documents to other devices for further processing. Document templates are used to create new documents, which pre-define the structure of the document. Each new document is assigned a unique Id, thus allowing multiple copies of a single template. Tasks are distributed along the pages depending on the nature of the task (text, mathematical expressions, or plots), allowing users to navigate back and forth. Apple pencil is used to perform these tasks, as shown in Fig. 2a. Writing data is recorded at a rate of 240 points per second. These points contain information about the pencil's location on the touchscreen, the force of the touch, altitude & azimuth angle, and time.[1] The writing of a user is rendered in view by linear interpolation between successive points. There is an option to use the eraser, which enables the user to undo writing mistakes.

We also discuss steps to refine the collected data into a proper dataset to be further used by the research community without reinventing the wheel. When a person writes, data are stored as a continuous stream of points in a *.csv* file. In addition to handwritten sequences, data also contain document Id and page information. So, preprocessing was done to cleanse the data, segment it into strokes and sequences. Every segmented sequence represents a single word or expression. The segmented sequences are of variable length and may compose of a single or multiple strokes. After preprocessing, the data look the same as was written on the iPad, as shown in Fig. 2b.

3.2. Feature extractor

Different sets of features have been presented in the literature (Liwicki & Bunke, 2009; Liwicki et al., 2006; Bahlmann, 2006; Otte et al., 2012) for handwriting recognition and classification. We present a new feature set, which includes some of the existing features. A

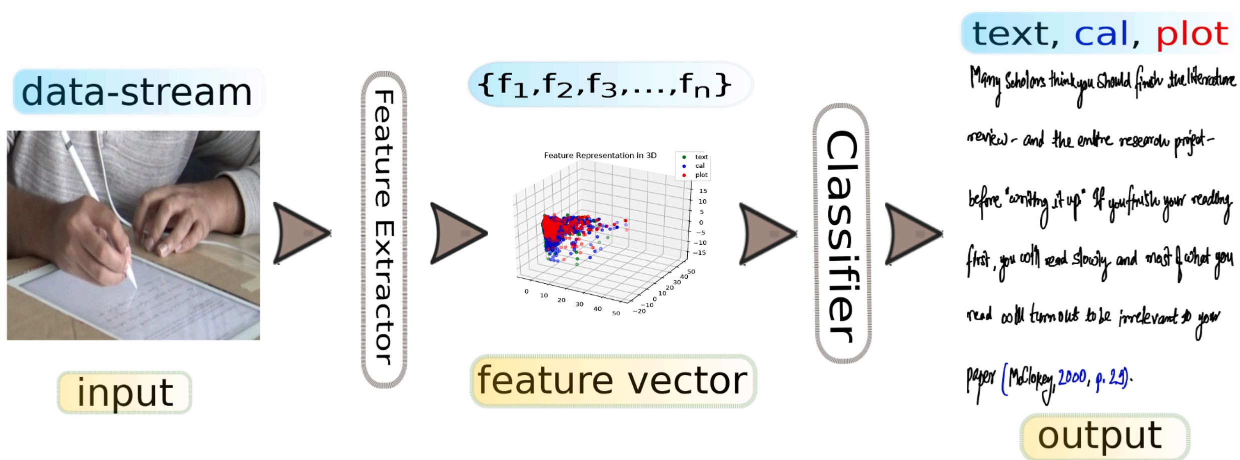
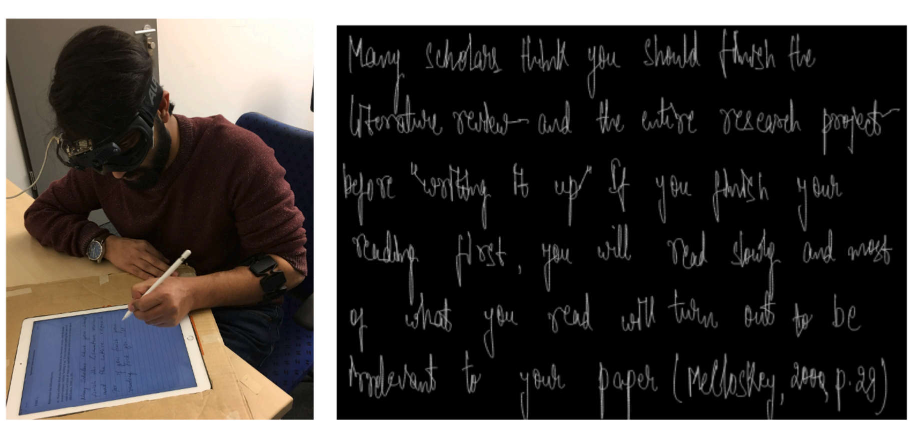


Fig. 1. System overview.



(a) Experimental setup. (b) Output sequences.

Fig. 2. Data collection setup.

[1] See https://developer.apple.com/documentation/uikit/uiview documentation of the UIView class.

Table 1
Overview of the proposed feature set.

Liwicki and Bunke (2006)	Younas et al. (2018)	Additional features
Speed	Sequence length	Sequence distance
Vicinity aspect	Sequence time	Sequence height & width
Vicinity curliness	Sequence displacement	Stroke count
Slope	Velocity	Average stroke distance
Linearity	Force range	Max. & Min. stroke distance
	Mean force	Max. & Min. stroke length
	Force variance	Max. & Min. stroke time
	Variance & Std. t	Mean stroke length
	x,y skew	Mean stroke time
	x,y kurtosis	Mean slope
	Variance & Std. x, y	Max. & Min. force
	Variance & Std. of direction angles	Force std.
	Variance & Std. of slope	Variance & Std. x-values
		Variance & Std. y-values

comprehensive overview of existing and new features is presented in Table 1. This feature set is used to transform online handwritten sequences into feature vectors. Sequences are recorded at the rate of 240 Hz, where at every point, p_i is recorded with information of time-stamped (x,y) coordinates, angles, and force, defined as $p_i = (x_i, y_i, f_i, \ldots\ldots, t_i)$.

A stroke starts with the pen-down movement of the Apple pencil writing on the iPad and ends with the next pen-up movement. Thus, a stroke is defined by a sequence of points, $s_i = [p_1, p_2, \ldots\ldots, p_n]$, for the time interval, $t_i = [t_1, t_2, \ldots, t_n]$, when the pencil-tip is in contact with iPad, whereas a sequence refers to a word or expression. A sequence can be composed of a single or multiple strokes, $seq_i = [s_1, s_2, s_3, \ldots, s_n]$. Every sequence is considered an independent entity and is transformed into feature vectors, $v_i = [f_{1i}, f_{2i}, f_{3i}, \ldots, f_{ni}] \in R^f$, which were later used to train machine-learning classifiers. Our presented feature set consists of the following 49 features contributing to achieve state-of-the-art performance using machine-learning classifiers.

- Length of a sequence, L_s (1), total length of strokes present in a sequence.

$$L_s = \sum_{i=1}^{n} \mathrm{len}(s_i) \tag{1}$$

- Time of a sequence, T_s (2), total time in seconds taken to complete a sequence.

$$T_s = [t_n - t_1]_s \tag{2}$$

- Sequence displacement, s (3), the shortest possible distance in pixels of pencil movement for a given segmented sequence.

$$s = \sqrt{x^2 + y^2} \tag{3}$$

- Sequence distance, D (4), the sum of displacement of consecutive points present in a sequence.
- Sequence height and width, (5), (6), sequence height is defined by the difference of maximum and minimum value of y-values present in the sequence, height = $\max[y_i]$ - $\min[y_i]$, while sequence width is difference of x-values, width = $\max[x_i]$ - $\min[x_i]$.
- Sequence slope or gradient, m (7), slope or gradient is measure of steepness and direction of line.

$$m = \frac{\Delta y}{\Delta x} \tag{4}$$

- Speed (8), the rate at which given sequence is produced.
- Velocity (9), rate of change of the displacement for a given sequence.
- Strokes count (10), number of times pen made contact with screen to complete a sequence.
- Average stroke distance (11), average stroke distance is calculated by averaging the total distance with the number of strokes present in a sequence.
- Maximum and minimum distance of stroke (12), (13), maximum and minimum distance of strokes, which are present in a given sequence.
- Maximum and minimum stroke length (14), (15), maximum and minimum length of strokes that are present in a given sequence.
- Maximum and minimum stroke time (16), (17), maximum time taken to produce a stroke in sequence as well as minimum time for a stroke.
- Mean stroke length (18), average length of strokes present in a sequence.
- Mean stroke time (19), average time taken to produce strokes of a sequence.
- Mean slope (20), average slope of strokes present in a sequence.
- Vicinity aspect (21), the aspect of the trajectory of a given sequence.

$$\frac{\Delta y - \Delta x}{\Delta y + \Delta x} \tag{5}$$

- Vicinity curliness (22), the length of a given sequence divided by max (Δx,Δy).
- Linearity (23), we define linearity of a sequence by average squared distance of strokes present in the sequence to the straight line.
- Maximum and minimum of force (24), (25), maximum and minimum of the pen force used to produce a given sequence.
- Range of force (26), the difference of maximum force value and minimum force value for a given sequence.
- Mean force (27), average force applied for a given sequence.
- Variance and standard deviation of force (28), (29).
- Variance and standard deviation (30), (31) of the rate of change during segmented sequence Δt.
- x,y-skew (32), (33), skewness is measure of amount and direction of departure from horizontal symmetry for a given sequence.
- x,y-kurtosis (34), (35), kurtosis is measure of height and sharpness of central peak for a given sequence.
- Variance of x, y (36), (37), the rate of change of pixels in both horizontal and vertical direction.
- Variance and standard deviation (38), (39) of x-values.
- Variance and standard deviation (40), (41) of y-values.
- Standard deviation of x, y (42), (43).
- Variance of direction angles of a given sequence (44), (45), measure of variance sin and cosine angles between consecutive pixel for a given sequence.
- Standard deviation of direction angles (46), (47).
- Variance and standard deviation of gradient of a given sequence (48), (49).

3.3. *Classifiers*

This section covers a detailed analysis of machine-learning classifiers used for evaluation. We used different classifiers based on bagging and boosting algorithms. Bagging algorithms are simple ensemble techniques that merge various classification models using voting strategies like average voting, majority voting, etc. Observations are chosen differently for individual models using the bootstrap process, which helps in achieving better generalization. Bagging classifiers are elaborated generically in Fig. 3. On the other hand, boosting algorithms are ensemble techniques that build models using a sequential learning process where observations are chosen based on classification error. Sequential learning results in better performance at every subsequent

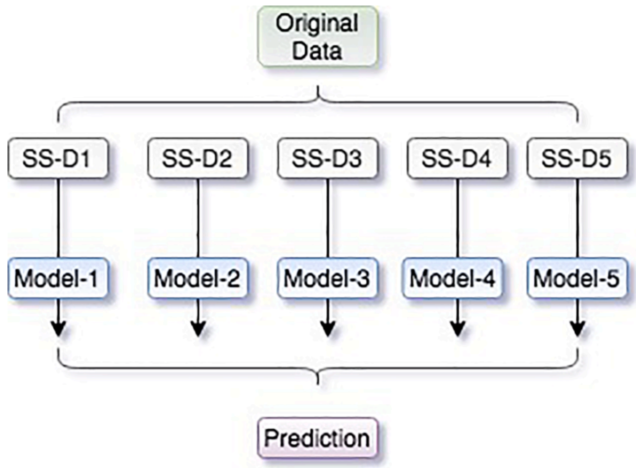


Fig. 3. Bagging classifier.

step. The following subsections explain various classification models used in this ablation study.

3.3.1. Random forest

Random forest classifier (RF) is considered a very effective machine-learning algorithm for predictions. RF is a meta estimator (Breiman, 2001) that follows the bagging technique. It uses decision trees (DT) as a basic building block. Multiple decision trees are combined to form a forest named as random forest. Each decision tree in the forest uses random sub-samples of training data and is built independently. Distribution is same for all the trees present in the forest. For classification results, RF uses the majority vote method to produce more diverse and robust results.

3.3.2. Bagging classifier

A bagging classifier (Breiman, 1996) is an ensemble algorithm that fits base classifiers, each on a random subset of data. It aggregates averages of individual predictions using a popularity vote method to estimate the final result. A bagging classifier is a way to reduce the variance of base estimators by introducing randomization, resulting in a significant performance boost.

3.3.3. Extra tree classifier

An extra tree (ET) classifier (Geurts et al., 2006) is a *meta*-estimator based on the bagging technique. It builds an ensemble of the unpruned decision or regression trees. The main difference of extra tree classifier with other ensemble methods is rather than using a random subset of data, and it uses complete data to build individual trees. Secondly, it splits the nodes by choosing cut-points entirely at random. The final prediction is achieved by aggregating the individual outputs using majority voting in classification problems and averaging them in regression problems.

3.3.4. Gradient boosting classifier

Gradient Boosting classifier or (GBM) (Friedman, 2002) is an ensemble machine-learning algorithm using the boosting technique. GBM is a sequential learning model that builds an additive model in a forward stagewise fashion. Every model at the subsequent step learns from the errors of the model at the previous step. Error is minimized by defining loss functions, i.e., mean squared error (MSE). Predictions are updated by using gradient descent and applying learning rate. Final predictions are made where the error is minimum and predicted values are closed to actual ones.

3.3.5. Recurrent neural networks

Recurrent Neural Networks are known for sequence data handling because of their ability to handle temporal information using self-connected hidden layers. The hidden layer implements input, forget, and output gates to regulate the dependencies. Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Gulcehre et al., 2014) are commonly used RNNs for handwriting recognition and classification. LSTM units implement memory gates to store the memory at different stages, enabling them to carry the early-stage features to later stages, allowing longer-distance dependencies. GRUs also keep the temporal information without implementing memory gates, making them adaptive to different time scales to store the dependencies. Memory gates solve the problem of vanishing gradients in back-propagation, resulting in improved performance.

4. Dataset & evaluation protocol

4.1. Overview

20 participants (14 males, 6 females) took part in the study. 18 participants were right-handed, and 17 participants(12 males, 5 females) had first-time writing experience on digital devices, iPads in this case. These participants were students from different disciplines and different geographical regions, i.e., Germany, Pakistan, India, Cuba, Venezuela, and the USA. The data collection was constraint-free as there were no time restrictions and all participants were free to write text, mathematical expressions, and/or plot graphs the way they wanted.

During experiments, participants solved different exercises based on instruction material provided to them. Exercises consist of text reproduction, creative writing, copying mathematical expressions, solutions to fundamental calculus problems, and drawing some easy graphs. Exercises were kept simple and elaborated so that every participant can understand them. The difficulty level increased as participants progressed with the solutions.

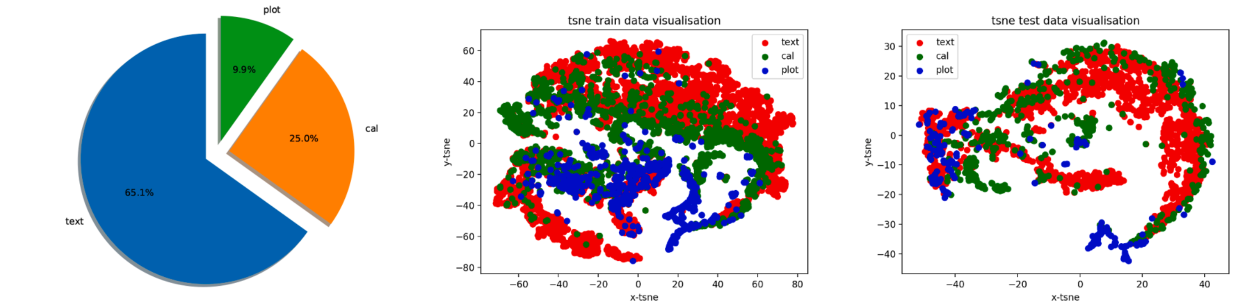
4.2. Database contents

Considering the classroom environment, note creation is a common activity, and the best way to monitor and track the progress is by online handwriting analysis. These notes mostly consist of plain text, whether it is structured, i.e., list, caption, part of a table, or unstructured, i.e., normal text, mathematical expressions that include numerical representation, formulas, axis-markers, and graph/plots. First of all, strokes are extracted from raw sequences. Every stroke is visualized individually, and then sequences are created out of these strokes manually to minimize the influence of segmentation error. Every sequence is annotated as text, mathematical expression, and plot/graph to generate the ground-truth. Our presented dataset consists of 12,139 labeled sequences. Dataset distribution is presented in Fig. 4a. 65% of sequences belong to text class, 25% of the sequences from mathematical expression class, and remaining 10% from plot/graph class. Dataset, namely onTabWriter containing stroke information and labeled sequence information is publicly available[2].

4.3. Feedback

After completing writing tasks, participants were requested to fill a feedback form. Based on feedback, 75% of participants found attempting solutions to calculus problems as the most difficult. In comparison, 60% of participants felt more stressed in solving mathematical expressions than producing text and drawing graphs. 70% of participants felt more comfortable copying text and solutions while remaining like

[2] http://bit.ly/2KWNEue.



(a) Class distribution in dataset (b) TSNE visualization of train data (c) TSNE visualization of test data

Fig. 4. Contents present in database with their representation and distribution in train and test set.

Table 2
Personal preference to write on traditional notebooks versus tablets.

Task	Traditional Notebook (%)	Tablets (%)
All	30	70
Text writing	35	65
Mathematical expressions	40	60
Graphs	50	50

creative tasks. We also asked participants about their preferences, provided both regular writing notebooks or digital devices. As depicted, for every task, the majority of the writers preferred writing on tablets, and statistics are presented in Table 2. Besides, almost every participant reported gradual improvement in writing ease and comfort with more writing practice.

4.4. Evaluation protocol

In this section, we will discuss the data split used to train and test machine-learning classifier, along with the analysis of metrics to report results. The database is analyzed in person-dependent settings where train and test data are split in a way that both contain writing data from all participants. We also report results in person-independent setup where data is split into train and test set so that a participant's data can be used either in the training phase or testing it, but cannot in both. Person-independent setup helps to establish the generalization of our presented approach. 4:1 split is used to transform the dataset into train and test sets. 80% of data is used in the training and optimization phase, and the rest 20% is used to test the model and report results. t-Distributed Stochastic Neighbour Embedding (t-SNE) visualization of train and test data is shown in Fig. 4b & c, respectively.

We used the scikit library (Pedregosa et al., 2011) to train and test our machine-learning classifier. Accuracy defines the correctness of a model and the most commonly used metric to report the results. When input data is biased or polarised, precision and recall become more relevant metrics. Therefore, we report results as the most relevant metrics in the online handwriting research community, i.e., accuracy, precision, recall, and f1-score. Precision is defined as the ability of a system to distinguish a positive sample from a negative one, whereas recall is the competence of a system to classify positive samples, mathematically defined as follows:

$$Precision = \frac{correctDetections}{totalDetections} \tag{6}$$

$$Recall = \frac{correctDetections}{totalsamples} \tag{7}$$

As we have close numbers for precision and recall for different classifiers, f1-score is used to represent the results, which is the harmonic mean of precision and recall. f1-score is mathematically defined as:

$$f1-score = 2\frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

4.5. Optimization parameters

Random forest, extra tree, bagging classifier, and gradient boosting classifiers are used in this study to establish the significance of the proposed feature set. Decision trees are used as a base estimator. All parameters are found after empirical evaluation. The number of trees is set to 199. We used the maximum available features to train and test all of our models. Criterion used for RF is 'entropy', for ET 'gini', and for GBM 'mse'. 'balanced' weight mode is used to address the partiality and bias in the data. We also tried different variants of LSTM and GRU networks. Both networks are three layers deep with [64, 128, 256] hidden units for each layer. A Learning rate of 0.001 is used with the batch size of 100. All networks are trained for 30 epochs.

5. Results and discussion

5.1. Results

In person-dependent data split and using bagging classifier, a

Table 3
Comparison of person-dependent performance of ML classifiers on different feature sets.

Feature set	Classifiers	Overall result %			
		Accuracy	Precision	Recall	f1-score
Younas et al. (2018)	Bagging classifier	79.4	79.1	79.4	79.2
	Extra tree	80.2	79.9	80.2	80.5
	Gradient boosting	79.8	79.6	79.8	79.7
	Random forest	77.9	78.5	77.9	78.2
Liwicki and Bunke (2006)	Bagging classifier	86.4	86.2	86.4	86.3
	Extra tree	85.5	85.2	85.5	
	Gradient boosting	85.2	85.1	85.2	85.3
	Random forest	85.5	85.5	85.5	85.5
Proposed	Bagging classifier	90.0	89.8	90.0	89.9
	Extra tree	90.3	90.2	90.3	90.2
	Gradient boosting	90.5	90.4	90.5	90.4
	Random forest	90.1	89.9	90.1	90.0
	GRU	89.7	89.1	89.7	89.1

successful classification rate of 90.0% is achieved with precision 89.8 and recall of 90.0, as reported in Table 3. Bagging classifier produced the highest score for text class by correctly classifying 95.9% sequences with the precision and recall of 92.0 and 95.9. Mathematical expressions are classified with the precision of 86.2, recall is 79.4, and classification accuracy is 79.4. Numbers reported for plot/graph using bagging classifier are 78.4, 84.6, and 78.4 percent in terms of accuracy, precision, and recall, respectively.

When it comes to RF classifier, overall accuracy, precision, and recall, each calculates to 90.1, 89.9, and 90.1. Classwise accuracy is 96.3, 78.9, and 79.9 percent, as shown in Table 5. The precision score is 92.4, 86.7, and 82.5 with recall of 96.0, 78.9, and 79.9 for text, mathematical expressions, and plot/graph class, respectively. Extra tree classifier shows marginally lesser performance than GBM classifier with the accuracy, precision, and recall score of almost 90.3. Text classification score is the best for extra tree classifier with the accuracy of 97.2%, while results for mathematical expressions class are the lowest with the accuracy of 77.7%, and graph/plot class is classified with the accuracy of 79.5%. Precision & recall scores for text, mathematical expressions, and graph class are 91.1 & 97.2, 90.4 & 77.7, and 86.9 & 79.5, respectively.

GBM produces the overall best results with an accuracy score of 90.5% and outperforms all its counterparts. The precision and recall score for GBM is 90.4 and 90.5. Mathematical expressions and plot/graph class are predicted with the classification accuracy of 81.3 and 79.9 percent, precision of 86.1 and 84.5, and recall of 81.3 and 79.9. The classification rate of 95.8 is achieved by GBM for text classification with the precision & recall of 93.0 & 95.8.

In person-dependent setup, all classifiers perform convincingly well, producing overall classification results with overall accuracy and f1-score in range of 90 ± 0.5.Gradient boosting classifier produced overall the state-of-the-art results, as reported in Table 3. Similarly, GBM outperforms its counterparts with the highest success rate in classifying individual classes i.e., text, and mathematical expressions with the f1-score of 94.1, and 83.6 as shown in Table 5. All of our presented models produce results with the f1-score of 90 ± 0.5, in comparison to the best results with f1-score of 80.5 achieved by Younas et al. (Younas et al., 2018) feature set using extra tree classifier. Similarly using Liwicki et al. (Liwicki and Bunke, 2006) feature set yields the f1-score of 86.3 for the best results using bagging classifier in person-dependent set-up. Our presented approach also outperforms their method in per class computed results, as shown in Table 5.

In person-independent data split, extra tree classifier produced the best results with an overall accuracy of 88.9% and f1-score of 88.8. The precision score of the extra tree classifier is 88.7 with the recall of 88.9. Bagging classifier surprisingly doesn't perform well in person-independent setup with overall classification accuracy and f1-score of 81.0 and 82.0. Overall precision and recall score for bagging classifier is 83.1 and 81.0. Accuracy, precision, recall, and f1-score of random forest is 88.6, 88.5, 88.6, and 88.5, respectively. With GBM, the same is not true as in person-dependent setup, as shown in Table 4. 87.3% of the sequences were predicted correctly with the precision of 87.4, recall of 87.3 and f1-score of 87.3. Younas et al. (2018) feature set achieved the best score of 78.3 in terms of accuracy with f1-score of 78.0 using random forest classifier and Liwicki and Bunke (2006) best results are 82.5 and 82.4 for accuracy and f1-score using gradient boosting classifier.

Text class is 94.4 times correctly classified by extra tree classifiers with the precision of 90.8 and recall of 94.4. RF classifier predicts text class with 92.9, 92.0, and 92.9 in terms of accuracy, precision, and recall, respectively. There is a significant decrease in numbers of text class prediction for bagging classifier, as reported in Table 6. Text classification accuracy is 82.3%, with precision and recall score of 90.3 and 82.3, respectively. GBM produced results with the accuracy of 91.2%, and precision & recall is 92.3 & 91.2 for text class. Mathematical expression class is correctly classified at the rate of 76.6, 77.5, 76.4, and

Table 4
Comparison of person-independent performance of ML classifiers on different feature sets.

Feature set	Classifiers	Overall result %			
		Accuracy	Precision	Recall	f1-score
Younas et al. (2018)	Bagging classifier	78.1	77.7	78.1	77.9
	Extra tree	78.2	78.0	78.2	78.1
	Gradient boosting	77.0	76.4	77.0	76.7
	Random forest	78.3	77.8	78.3	78.0
Liwicki and Bunke (2006)	Bagging classifier	82.0	82.0	82.0	82.0
	Extra tree	80.8	80.7	80.8	80.7
	Gradient boosting	82.5	82.3	82.5	82.4
	Random forest	81.2	81.4	81.2	81.3
Proposed	Bagging classifier	81.0	83.1	81.0	82.0
	Extra tree	88.9	88.7	88.9	88.8
	Gradient boosting	87.3	87.4	87.3	87.3
	Random forest	88.6	88.5	88.6	88.5
	GRU	87.4	87.7	87.4	87.5

79.0 with precision of 73.8, 80.7, 81.4, and 75.9 for bagging classifier, RF, ET, and GBM, respectively. The recall rate of mathematical expressions is 76.6, 77.5, 76.4, and 79.0, respectively. Plot/graph class is classified by bagging classifier with the accuracy of 79.4, precision & recall score is 51.6 & 79.4, results for RF classifier are 84.9, 82.7, and 84.9, respectively. Accuracy of GBM and extra tree classifier to predict plot class is 80.0% and 79.0% with the precision & recall of 81.0 & 80.0 and 88.3 & 79.0.

We also evaluated the presented feature set using deep-learning methods. LSTM and GRU models were trained and tested for both person-dependent and person-independent setup. The best results are achieved by using 256 hidden units with the GRU network. In person-dependent setup, the overall accuracy of 89.7% with the precision, recall, and f1-score of 89.1, 89.7, and 89.1, respectively, is acheived. Results are 87.4, 87.7, 87.4, and 87.5 in terms of accuracy, precision, recall, and f1-score, respectively, for person-independent setup. In person-dependent setup, GRU network achieved the best results among all the classifiers for graph/plot class with an f1-score of 86.9. Similarly, in person-independent setup, GRU delivered the best results for the mathematical expression class with an f1-score of 85.0. Results furnished in Tables 5 & 6 demonstrate the competitive results for both the machine-learning and the deep-learning method, which establishes the relevance and importance of presented feature set for online-handwriting classification.

5.2. Discussion

We trained the same classifiers on Liwicki & Bunke (2006) and Younas et al. (2018) proposed feature sets using the proposed dataset for a detailed and fair comparison. Our proposed feature set achieved the best results in both person-dependent and person-independent setup and outperformed its counterparts in all metric scores. The computational load for the proposed feature set on all the classifiers is also minimal, making it ideal for real-time use in low-cost systems. Moreover, results achieved by our proposed feature set are also superior for every class, i.e., text, mathematical expressions, and plotting graphs classification by a margin.

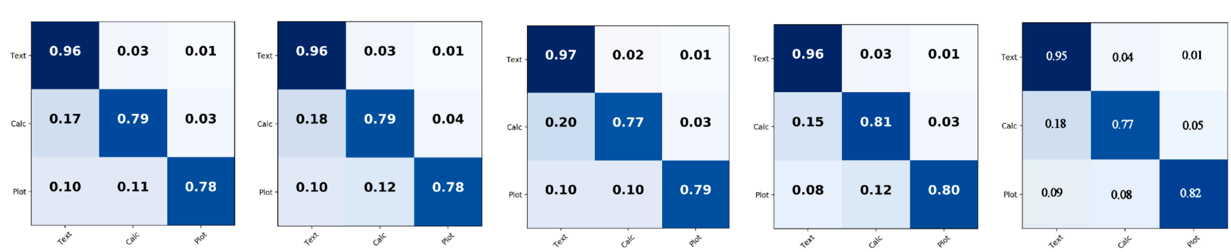
When there is a clear pattern and structure in the writing of a participant, ideal results, regardless of the sequence class, are produced, as shown in Fig. 7b, with ground truth file in Fig. 7a. Fig. 5 shows that every classifier showed distinct performance on classifying text, because writing text exhibits a clear pattern and distinguished writing behavior.

Table 5
Class-wise detailed results of different feature sets for Machine-learning classifiers in person-dependent set up on newly proposed dataset.

Feature set	Classifiers	Text %				Mathematical expressions%				Graph%			
		Accuracy	Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score
Younas et al. (2018)	Bagging classifier	88.7	85.6	88.7	87.1	68.2	68.8	68.5	68.2	67.1	74.4	67.1	70.6
	Extra tree	90.9	84.6	90.9	87.6	67.6	71.5	67.6	69.5	66.1	77.3	66.1	69.4
	Gradient boosting	88.6	86.1	88.6	87.3	70.0	69.7	70.0	69.8	67.3	74.4	67.3	70.7
	Random forest	83.8	88.0	83.8	85.8	70.0	65.9	70.0	67.9	71.3	67.6	71.3	69.4
Liwicki and Bunke (2006)	Bagging classifier	93.7	89.9	93.7	91.8	71.9	81.5	71.9	76.4	74.8	73.0	74.8	73.9
	Extra tree	94.4	88.1	94.4	91.1	66.5	82.3	66.5	73.6	74.8	73.3	74.8	74.0
	Gradient boosting	92.0	89.8	91.8	90.8	73.2	78.1	73.2	75.6	72.8	71.3	72.8	72.0
	Random forest	92.6	90.1	92.6	91.3	70.9	79.8	70.9	75.1	76.0	68.7	76.0	72.2
Proposed	Bagging classifier	95.9	92.0	95.9	93.9	79.4	86.2	79.4	75.1	78.4	84.6	78.4	72.2
	Extra tree	97.2	91.1	97.2	94.1	77.7	90.4	77.7	82.7	79.5	86.9	79.5	81.4
	Gradient boosting	95.8	93.0	95.8	94.1	81.3	86.1	81.3	83.6	79.9	84.5	79.9	83.0
	Random forest	96.0	92.4	96.0	94.1	78.9	86.7	78.9	82.6	79.9	82.5	79.9	81.2
	GRU	95.2	91.7	95.2	93.4	77.0	95.2	77.0	85.1	82.2	93.4	82.2	86.9

Table 6
. Class-wise detailed results of different feature sets for Machine-learning classifiers in person-independent set up on new proposed dataset.

Feature set	Classifiers	Text %				Mathematical expressions%				Graph%			
		Accuracy	Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score
	Bagging classifier	91.9	80.0	91.9	85.5	57.8	73.6	57.8	64.7	69.5	77.1	69.5	73.1
Younas et al. (2018)	Extra tree	94.7	78.6	94.7	85.9	55.5	75.8	55.5	64.1	65.5	79.7	65.5	71.9
	Gradient boosting	90.0	80.4	90.0	84.9	57.6	71.4	57.6	63.8	69.1	72.7	69.1	70.9
	Random forest	89.8	82.7	89.8	86.1	60.7	72.6	60.7	66.1	72.1	71.8	72.1	71.9
	Bagging classifier	88.4	87.2	87.8	88.4	70.1	68.8	70.1	69.4	65.3	76.1	65.3	70.3
Liwicki and Bunke (2006)	Extra tree	88.2	85.7	88.2	86.9	66.6	65.9	66.6	66.2	62.7	81.1	62.7	70.7
	Gradient boosting	89.3	87.5	89.3	88.4	71.1	69.6	71.1	70.3	61.7	76.0	61.7	68.1
	Random Forest	86.5	87.7	86.5	87.1	70.4	67.3	70.4	68.8	69.6	70.3	69.6	69.9
	Bagging classifier	82.3	90.3	82.3	86.1	76.6	73.8	76.6	75.2	79.4	51.6	79.4	62.5
proposed	Extra tree	94.4	90.8	94.4	92.6	76.4	81.4	76.4	78.8	79.0	88.3	79.0	83.4
	Gradient boosting	91.2	92.3	91.2	91.7	79.0	75.9	79.0	77.4	80.0	81.0	80.0	80.5
	Random forest	92.9	92.0	92.9	92.5	77.5	80.7	77.5	79.1	84.9	82.7	84.9	83.8
	GRU	90.5	92.8	90.5	91.6	80.1	90.5	80.1	85.0	83.1	91.6	83.1	87.1

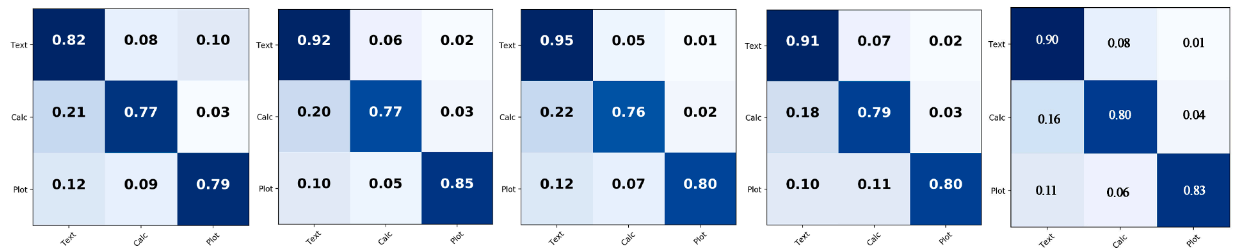


(a) Bagging Classifier (b) Random Forest (c) Extra Tree (d) Gradient Boosting (e) LSTM

Fig. 5. Normalized confusion matrices for person-dependent results.

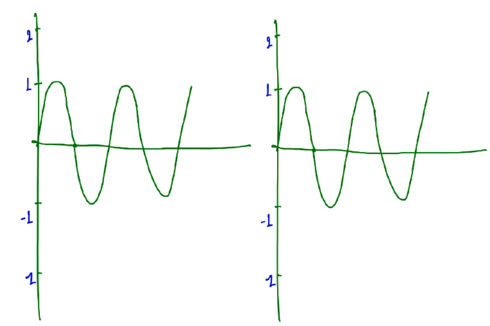
Few text sequences produced by writing single or few strokes are confused either with mathematical expressions or plot/graph, classification results in comparison to the ground-truth file are shown in Fig. 9a & b (see Fig. 6).

The proposed feature set makes every classifier fully adept of classifying free-style and constraint-free online handwritten sequences into text, mathematical expression, and/or plots/graphs. Furthermore, it can classify minority class sequences among the majority class sequences, i.



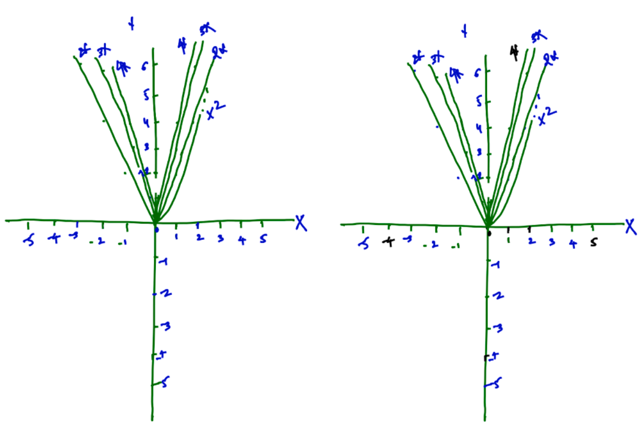
(a) Bagging Classifier (b) Random Forest (c) Extra Tree (d) Gradient Boosting (e) LSTM

Fig. 6. Normalized confusion matrices for person-independent results.



(a) Ground truth. (b) Output

Fig. 7. A perfect classification result (green color annotates plot, while blue color is labelled as mathematical expressions). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Ground truth. (b) Classifier's result.

Fig. 8. An example of plot classification with annotated ground-truth (complex scenario) (green color annotates plot and blue color is for mathematical expressions, while black color highlights text).

e., single mathematical expression correctly classified within a text block, as shown in Fig. 9b. As discussed earlier, every sequence is considered as an individual sequence despite its position or context in the text. Results establish the significance of the presented feature set for online handwritten sequences. As every classifier in an ablation study performs equally well to produce state-of-the-art results both in person-dependent and person-independent data split.

Producing mathematical formulas, their derivations and solutions have a very close resemblance to text. Therefore, mathematical expressions class gets confused with the text class, as shown in Fig. 10a & b. Mathematical expressions and text writing are very different from plot class; therefore, there are lesser confusions between these classes, as results also demonstrate. Plot/graph sequence visually show a clear pattern, but markers and ticks are single stroke sequences and are treated as independent sequences. We don't use any context information; therefore, few confusions with text and/or mathematical expressions, as shown in Fig. 8a & b.

The feature set presented by Liwicki & Bunke (2006) was mainly for handwriting recognition but has been adopted for handwriting classification. Meanwhile, the paraphernalia for online handwriting recording is much advanced over the decade, i.e., the introduction of sensor pens, smart-pens to write on papers and digital screens. These devices record pen pressure, force, time, and angles along with (x,y) coordinates, which can help to develop better systems for handwriting classification. The ablation studies results on the presented feature set, a combination of existing and new features institute a better understanding of the problem at hand, as shown in Tables 3 & 4. Furthermore, the relevancy of features in the presented feature set is also demonstrated by the machine-learning classifier's results compared to the deep-learning classifier. As handwriting classification is very different task than handwriting recognition, therefore a new feature set is presented with the focus on online handwriting classification will help the research community to further push the boundaries in this direction.

We also discuss the significance of features present in our feature set. To start with, individual features are evaluated for their impact as a whole as well as for every class, as shown in Fig. 11a. By evaluating individual features, we get an insight into the performance and importance of individual features along with in-class comparison. Moreover, we also rank the features based on their importance and significance in the given set, starting from maximum and dropping the least important feature at every subsequent step, as shown in Fig. 11b. The most important features present in the proposed feature set are maximum stroke distance, variance of y, vicinity curliness, variance of y-values, etc. The least important features in the rankings are minimum force, sequence time, average change in directional angles, etc.

Once we have insights about the impact of individual features that can be further utilized to evaluate and find out important features for individual classes, as shown in Fig. 12. Simple peaks in the individual feature impact graph provide the overall important features along with important features for text, calculation, and plot class. Ten features were found to be most contributing to text classification, which include stroke, sequence, and time-related features, and results are visualized in Fig. 12a. Similarly, 14 features influenced most in the classification of mathematical expressions and results are shown in Fig. 12b. For classification of plot/graph class, 12 features are marked as high impact, and results are shown in Fig. 12c.

Common features in all the classes are related to stroke and sequence distance, length, time, standard deviation, and variance along horizontal and vertical axis. We combine these individual features to form a superset of important features and results are presented in Fig. 12d. By using selected features based on their rankings, we achieved nearly the

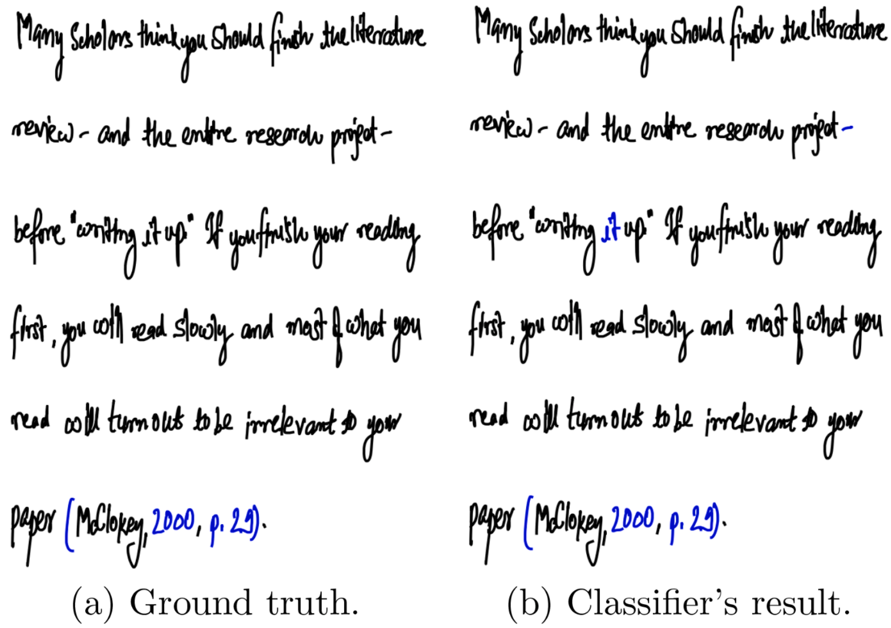


Fig. 9. An example of text classification along with ground-truth (black color annotates text, while blue color is labelled as mathematical expressions).

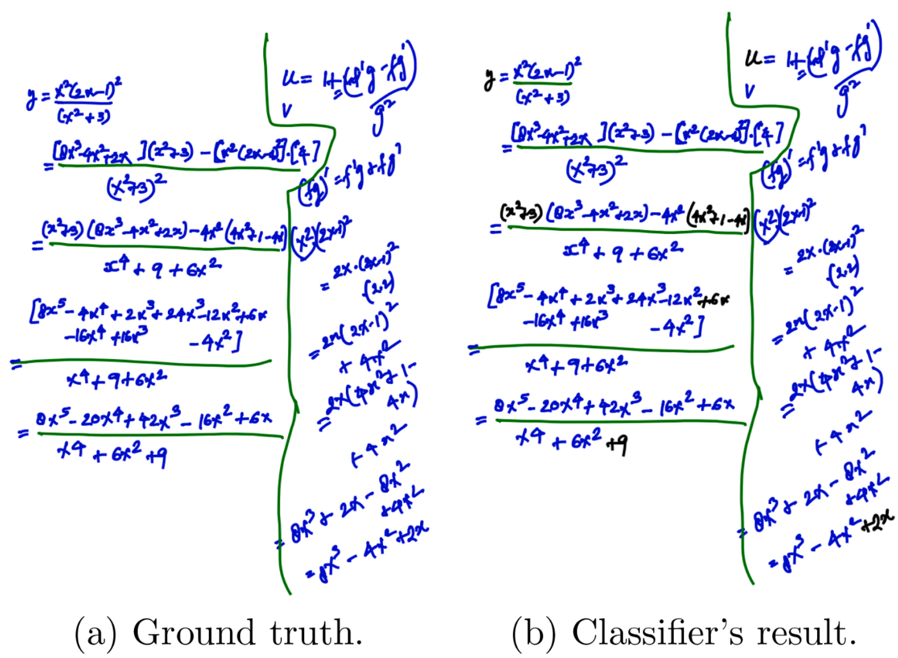


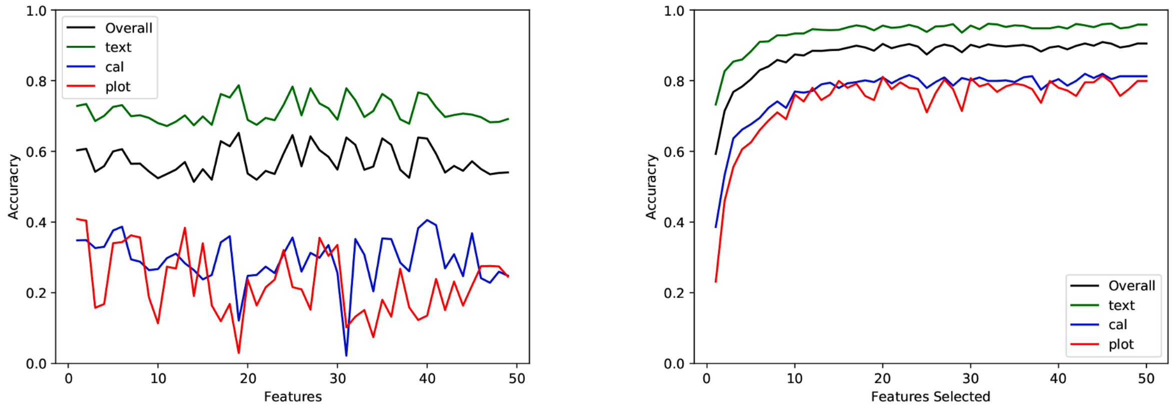
Fig. 10. Classification of mathematical expressions class in complex scenario (blue color annotates mathematical expressions and green color annotates plot, while black color is labelled as text.

same performance as using the complete feature set. Details of selected features are provided in Table 7, which contains few features from Liwicki & Bunke (2006), Younas et al. (2018), and newly proposed features.

We also evaluate existing feature sets on our proposed dataset to establish the utility, superiority, and a fair comparison with our presented feature set. Although the feature set presented by Liwicki and Bunke (2006), Liwicki et al. (2006) focused on handwriting recognition, even then our proposed feature set yields not only overall superior results but also surpasses their results for text detection. It also outperforms results produced by using existing feature sets to classify mathematical expression and plot/graph class by a margin. We evaluate the existing feature set on the best performing classifier in this study, yet every classifier produced better results on the newly proposed feature set.

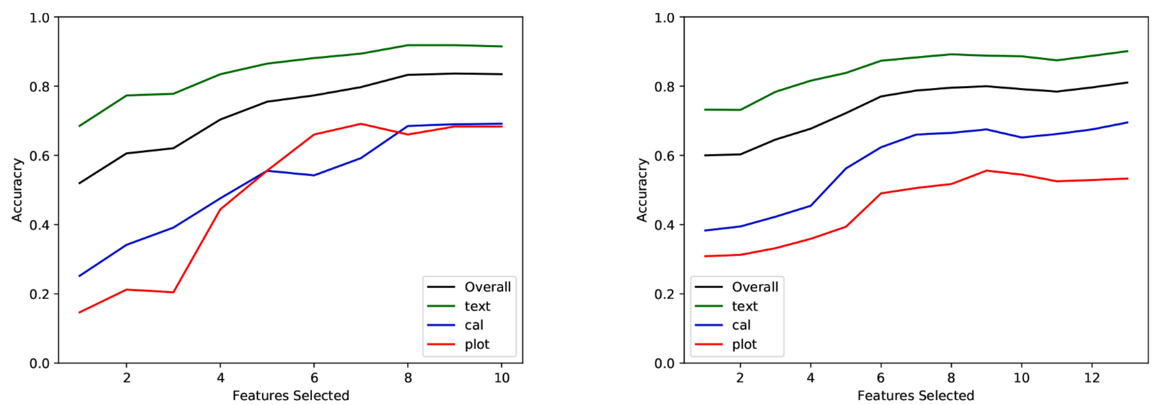
6. Conclusion

In this paper, we propose a novel feature set to classify online handwritten sequences into text, mathematical expression, and plot/graphs. Our presented feature set produced the state-of-the-art results on unseen data using various machine learning classifiers. The proposed feature set incorporates stroke information along with sequence information. We also present a new public dataset for constraint-free online handwriting classification using a sensor pen and touch display. This dataset is challenging as the contributors were allowed to write in

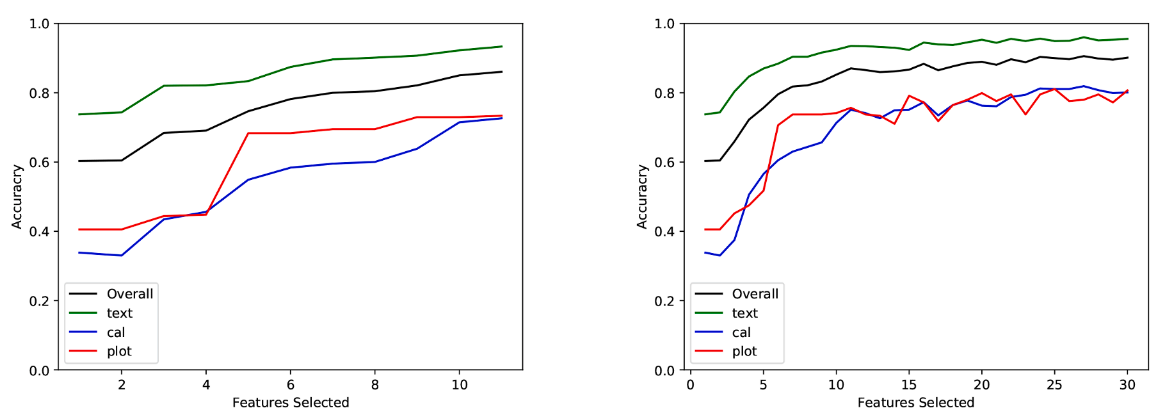


(a) Individual feature contribution (b) Feature impact evaluation based on their ranking

Fig. 11. Overview of our presented feature set.



(a) Top performing features for Text (b) Top performing features for Calculation



(c) Top performing features for Plot (d) Top performing features combined

Fig. 12. Evaluation of top performing features from proposed feature set.

freestyle - where sequences lack clear patterns - causing difficulty for classifiers.

Although we present the results on deep-models, we recommend exploring deep-learning models in combination of statistics-based models as a prospect of this work. A comparison of our presented features with the features produced by deep neural networks will be very interesting and highly encouraged. Our presented dataset can also be used to analyze the writing behavior and classroom performance of

Table 7
Selected features based on their relevance.

Liwicki and Bunke (2006)	Younas et al. (2018)	Additional features
Speed	Sequence displacement	Sequence distance
Vicinity aspect	Velocity	Sequence height
Vicinity curliness	Mean force	Max. & Min. stroke distance
Slope	Variance & Std. t	Max. & Min. stroke length
	x,y skew	Max. stroke time
	x,y kurtosis	Mean slope
	Variance x; y	Max. force
	Variance of direction angles	Force std.
		Variance x-values
		Variance & Std. y-values

students. We recommend using the context information for blockwise classification of data, which could significantly increase classifiers' performances by incorporating contextual information.

CRediT authorship contribution statement

Junaid Younas: Conceptualization, Methodology, Data curation, Writing - original draft, Investigation, Visualization, Writing - review & editing, Software. **Muhammad Imran Malik:** Validation, Writing - review & editing. **Sheraz Ahmed:** Conceptualization, Supervision. **Faisal Shafait:** Supervision. **Paul Lukowicz:** Resources, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Ahmad, A. R., Khalia, M., Viard-Gaudin, C., & Poisson, E. (2004). On-line handwriting recognition using support vector machine. In TENCON. (Vol. A, pp. 311-314).

Bahlmann, C. (2006). Directional features in on-line handwriting recognition. Pattern Recognition, 39(1), 115–125.

Bahlmann, C., Haasdonk, B., & Burkhardt, H. (2002). On-line handwriting recognition with support vector machines - a kernel approach. In *In 8th Int. Workshop on Frontiers in Handwriting Recognition* (pp. 49–54).

B.Livingston, O. (1959). A handwriting and pen-printing classification system for identifying law violators. Journal of Criminal Law and Criminology, 49.

Bouletreau, V., Vincent, N., Sabourin, R., and Emptoz, H. (1997). Synthetic parameters for handwriting classification. In (ICDAR), volume 1, pages 102-106 vol.1.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Delaye, A., & Lee, K. (2015). A exible framework for on-line document segmentation by pairwise stroke distance learning. *Pattern Recognition, 48*(4), 1197–1210.

Delaye, A., & Liu, C. (2013). Graphics extraction from heterogeneous on-line documents with hierarchical random fields. *(ICDAR)*, 1007–1011.

Delaye, A., & Liu, C.-L. (2014). Multi-class segmentation of free-form on-line documents with tree conditional random fields. *International Journal on Document Analysis and Recognition (IJDAR), 17*(4), 313–329.

Diaz, M., Ferrer, M. A., Impedovo, D., Malik, M. I., Pirlo, G., & Plamondon, R. (2019). A perspective analysis of handwritten signature technology. *ACM Computing Surveys, 51*(6), 117.

Dunn, J., & Sweeney, T. (2018). Writing and iPads in the early years: Perspectives from within the classroom. *British Journal of Educational Technology, 49*(5), 859–869.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367–378.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3-42.

Griechisch, E., Malik, M. I., & Liwicki, M. (2014). On-line signature verification based on kolmogorov-smirnov distribution distance. In *In 14th Int. Conf. on Frontiers in Handwriting Recognition* (pp. 738–742).

Gulcehre, C., Cho, K., Pascanu, R., & Bengio, Y. (2014). Learned-norm pooling for deep feedforward and recurrent neural networks. In T. Calders, F. Esposito, E. Hullermeier, R. Meo, editors, Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I, volume 8724 of Lecture Notes in Computer Science (pp. 530–546). Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. 9(8), 1735–1780.

Hulls, C. C. W. (2005). Using a Tablet PC for classroom instruction. In Frontiers in education 35th annual conference.

Indermuhle, E., Liwicki, M., & Bunke, H. (2010). Iamondo-database: An on-line handwritten document database with non-uniform contents. In: DAS '10 (pp. 97–104). ACM.

Indermuhle, E., Frinken, V., & Bunke, H. (2012). Mode detection in on-line handwritten documents using BLSTM neural networks. In *In 13th Int. Conf. on Frontiers in Handwriting Recognition* (pp. 302–307).

Kholmatov, A., & Yanikoglu, B. (2005). Identity authentication using improved on-line signature verification method. *Pattern Recognition Letters, 26*(15), 2400–2408.

Koile, K., & Singer, D. (2006). Development of a Tablet-PC-based system to increase instructor-student classroom interactions and student learning.

Liwicki, M., & Bunke, H. (2006). HMM-based on-line recognition of handwritten whiteboard notes. *In 10th Int. Workshop on Frontiers in Handwriting Recognition*.

Liwicki, M., & Bunke, H. (2009). Feature selection for HMM and BLSTM based handwriting recognition of whiteboard notes. *IJPRAI, 23*, 907–923.

Liwicki, M., Schlapbach, A., Bunke, H., Bengio, S., Mariethoz, J., & Richiardi, J. (2006). *2006*. DAS: Writer Identification for Smart Meeting Room Systems. Technical report.

Malik, M. I., & Liwicki, M. (2012). From terminology to evaluation: Performance assessment of automatic signature verification systems. *Int. Conf. on Frontiers in Handwriting Recognition*, 613–618.

Malik, M. I., Liwicki, M., & Dengel, A. (2013). Part-based automatic system in comparison to human experts for forensic signature verification. *ICDAR*, 872–876.

Mandal, S., Prasanna, S. R. M., & Sundaram, S. (2018). Gmm posterior features for improving online handwriting recognition. *Expert Systems with Applications, 97*, 421–433.

Moore, J. (1945). Handwriting classification. *The Police Journal*, 39–61.

Otte, S., Krechel, D., Liwicki, M., & Dengel, A. (2012). Local feature based on-line mode detection with recurrent neural networks. *Int. Conf. on Frontiers in Handwriting Recognition*, 533–537.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Machine Learning, 12*, 2825–2830.

Phan, T. V., & Nakagawa, M. (2014). Text/non-text classification in on-line handwritten documents with recurrent neural networks. In *In 14th Int. Conf. on Frontiers in Handwriting Recognition* (pp. 23–28).

Said, H. E. S., Baker, K. D., & Tan, T. N. (1998). Personal identification based on handwriting. In 14th int. conf. on pattern recognition (Vol. 2, pp. 1761–1764).

Schelske, Y. T., Dengel, A., Strau, F., Liwicki, M., Schoelzel, C., & Weber, M. (2011). MCS for on-line mode detection: Evaluation on pen-enabled multi-touch interfaces. *(ICDAR), 00*, 957–961.

Schomaker, L., Abbink, G., and Selen, S. (1994). Writer and writing-style classification in the recognition of on-line handwriting. In IEE European workshop on handwriting analysis and recognition (pp. 1/1-1/4).

Sesa-Nogueras, E., Faundez-Zanuy, M., & Roure-Alcobe, J. (2016). Gender classification by means of online uppercase handwriting: A text-dependent allographic approach. *Cognitive Computation, 8*(1), 15–29.

Sharma, A., & Sundaram, S. (2017). A novel on-line signature verification system based on GMM features in a DTW framework. *IEEE Transactions on Information Forensics and Security, 12*(3), 705–718.

Smith, T. L. (1954). Six basic factors in handwriting classification. *Journal of Criminal Law and Criminology*.

Twyman, J. S., & Heward, W. L. (2018). How to improve student learning in every classroom now. *International Journal of Educational Research, 87*, 78–90.

Weber, M., Liwicki, M., Schelske, Y. T. H., Schoelzel, C., Strauß, F., & Dengel, A. (2011). MCS for Online Mode Detection: Evaluation on Pen-Enabled Multi-touch Interfaces. *International Conference on Document Analysis and Recognition*.

Yang, J., Pan, H., Zhou, W., & Huang, R. (2018). Evaluation of smart classroom from the perspective of infusing technology into pedagogy. *Smart Learning Environments, 5*(1), 20.

Younas, J., Fritsch, S., Pirkl, G., Ahmed, S., Malik, M. I., Shafait, F., & Lukowicz, P. (2018). What am I writing: Classification of on-line hand-written sequences. *Intelligent Environments (Workshops), 23*, 417–426.