The OnHW Dataset: Online Handwriting Recognition from IMU-Enhanced Ballpoint Pens with Machine Learning

FELIX OTT*, Fraunhofer Institute for Integrated Circuits (IIS), Nuremberg, Germany and Ludwig-Maximilians-University (LMU), Munich, Germany
MOHAMAD WEHBI*, Friedrich-Alexander University (FAU) Erlangen-Nuremberg, Germany
TIM HAMANN, STABILO International GmbH, Heroldsberg, Germany
JENS BARTH, STABILO International GmbH, Heroldsberg, Germany
BJÖRN ESKOFIER, Friedrich-Alexander University (FAU) Erlangen-Nuremberg, Germany
CHRISTOPHER MUTSCHLER, Fraunhofer Institute for Integrated Circuits (IIS), Nuremberg, Germany and Ludwig-Maximilians-University (LMU), Munich, Germany

This paper presents a handwriting recognition (HWR) system that deals with online character recognition in real-time. Our sensor-enhanced ballpoint pen delivers sensor data streams from triaxial acceleration, gyroscope, magnetometer and force signals at 100 Hz. As most existing datasets do not meet the requirements of online handwriting recognition and as they have been collected using specific equipment under constrained conditions, we propose a novel online handwriting dataset acquired from 119 writers consisting of 31,275 uppercase and lowercase English alphabet character recordings (52 classes) as part of the *UbiComp 2020 Time Series Classification Challenge*. Our novel `OnHW-chars` dataset allows for the evaluations of uppercase, lowercase and combined classification tasks, on both writer-dependent (WD) and writer-independent (WI) classes and we show that properly tuned machine learning pipelines as well as deep learning classifiers (such as CNNs, LSTMs, and BiLSTMs) yield accuracies up to 90 % for the WD task and 83 % for the WI task for uppercase characters. Our baseline implementations together with the rich and publicly available `OnHW` dataset serve as a baseline for future research in that area.

CCS Concepts: • **Hardware** → **Emerging tools and methodologies**; **Sensor devices and platforms**; Noise reduction; Digital signal processing; Sensor applications and deployments; • **Human-centered computing** → *Ubiquitous and mobile devices*; • **Computing methodologies** → Neural networks.

Additional Key Words and Phrases: Online handwriting recognition, character dataset, inertial measurement unit, time-series data, sensor-based pen, writer-(in)dependent, multi-stroke gestures, embedded

1 INTRODUCTION

Handwriting involves a representation of the language by structured symbols and applies thoughts and spoken language onto paper. It is used for communication between individuals or for the documentation of thoughts for further use. Handwriting Recognition (HWR) is the process of converting written text into a digitized form

*Both authors contributed equally to this research.

Authors' addresses: Felix Ott, felix.ott@iis.fraunhofer.de, Fraunhofer Institute for Integrated Circuits (IIS), Nuremberg, Germany, Nordostpark 84, Nuremberg, 90411, Ludwig-Maximilians-University (LMU), Munich, Germany; Mohamad Wehbi, mohamad.wehbi@fau.de, Friedrich-Alexander University (FAU) Erlangen-Nuremberg, Germany, Erlangen, 91052; Tim Hamann, tim.hamann@stabilo.com, STABILO International GmbH, Heroldsberg, Germany, Heroldsberg, 90562; Jens Barth, jens.barth@stabilo.com, STABILO International GmbH, Heroldsberg, Germany, Heroldsberg, 90562; Björn Eskofier, bjoern.eskofier@fau.de, Friedrich-Alexander University (FAU) Erlangen-Nuremberg, Germany, Erlangen, 91052; Christopher Mutschler, christopher.mutschler@iis.fraunhofer.de, Fraunhofer Institute for Integrated Circuits (IIS), Nuremberg, Germany, Nordostpark 84, Nuremberg, 90411, Ludwig-Maximilians-University (LMU), Munich, Germany.

that a computer can understand. HWR has been studied for several years, however, it still presents a challenge that requires further research since the need for HWR systems is growing with the extended need for the use of digitized systems [57]. This domain can be categorized into two distinguished types: offline and online HWR.

Optical Character Recognition (OCR) falls into the domain of offline HWR and describes the analysis of the visual representation making use of offline features of the input, where the input of the system is an image containing handwriting. The written paper is scanned into a digitized image through, e.g., a digitizer, a tablet or a camera, then segmented into different segments that could include lines, words, or letters, which then undergo the recognition process [56]. Offline HWR systems have reached near-human performance results and have been successfully implemented in different areas of applications such as signature verification [21], reading bank checks and postal addresses. However, offline HWR cannot be applied for applications that require a real-time recognition (as there is no image of the document immediately after the completion). Furthermore, from simply analyzing the images it is not possible to make use of rich information such as the temporal direction of writing, the writing order, writing speed, and (in some cases) the pressure of writing. Only using the position of the strokes leads to ambiguities if letters overlap [15, 71].

Online HWR (OnHWR) typically uses time in association with different types of spatio-temporal signals. The data may contain a form of positions including information about the displacement of certain input devices, or may include the movement of the input devices on the writing surface. These signals are then processed by a recognition system that orders the strokes by their position and time and that can make use of the geometrical design and dynamic information from the movement of the writer. In many previous work a stylus pen together with a touch screen surface usually serve as input devices. Through the temporal information online HWR systems can be more accurate than offline systems, since similarly shaped characters can be distinguished by knowing the number of strokes that were necessary [56, 71].

One application of HWR systems is the commitment in primary school classes, where the teacher instructs an essay, for example, the pupils write with the sensor-enhanced pen on normal paper, and the text can be converted to a computer-based format automatically and online. The teacher receives immediately a status of the process. Furthermore, the written text can directly be corrected, and decreases the teacher's effort. Currently, no HWR system suffices all requirements for such an application, as such systems are either offline, require a pen that influences the graphomotoric of the writer, or requires for writing on a tablet that is expensive and influences the writing style [48]. The required device for the sensor-pen is just a computer, tablet, or phone with an installed app with a pen-device bluetooth connection that is often available anyways.

For the evaluation of HWR systems and also of the writing-style writer-specific and platform-specific aspects are necessary that have to be considered. The identification of handedness of the writer plays an important role to study and compare left-handed and right-handed writers. Previous work analyzes the handedness on the basis of strokes and slope of letter [63]. The writing performances of dysgraphic and proficient writers are compared by a distinction between the number and duration of two kinds of pauses, i.e., pen stops and pen lifts [55]. For the design of the recording platform pen-based systems can be favored over tablet or keyboard systems, as writing with a pen provides better cognitive processing, i.e., theoretical understanding, critical thinking and memory recollection [3, 65]. Modifications in writing conditions, e.g., a keyboard or a smoother writing surface of a tablet, might influence the writing performance, in particular, those of non-automatized beginning writers such as children as their handwriting movements require visual and graphomotor feedback [26]. Hence, pen-based OnHWR systems on paper have the lowest impact on the graphomotor.

HWR systems require large amounts of training data to acquire the ability of understanding and classifying what the user is writing. However, the data collection process is a time and resource consuming process. Consequently, for the sake of progressing within the specific domain of research, collected datasets are shared within the scientific community. This field has been researched for many years with several databases being published. However, many of these published datasets were collected using specific expensive equipment [1, 47, 51] that

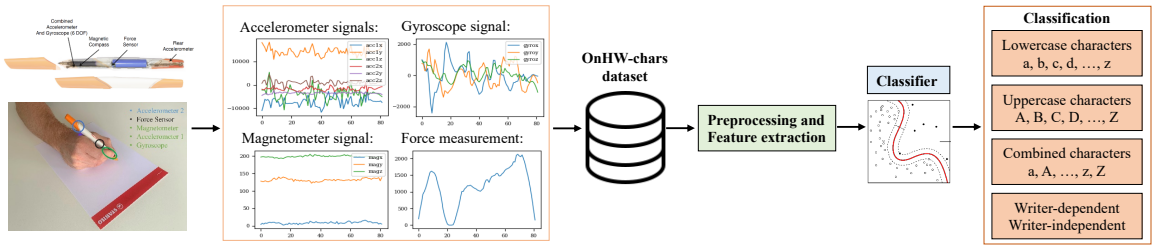


Fig. 1. Complete pipeline. (1) The recording setup is a STABILO DigiPen and a tablet storing sensor data and ground truth labels. (2) The ballpoint pen is enhanced with two accelerometers, one magnetometer, one gyroscope and one force sensor. (3) The `OnHW-chars` database consists of 31,275 uppercase and lowercase letters (52 classes) from 119 writers. (4) Pre-processing and noise filtering is necessary for (5) training the classifier. (6) We present the results for lowercase, uppercase and combined letters, both on writer-dependent (WD) and writer-independent (WI) classification tasks.

make a recognition system unachievable when applying for a real use-case utilization [45], are too small [77], or only address specific aspects [11, 34, 60]. Hence, the availability of a dataset collected with a convenient digital pen is essential for the scientific community. The primary purpose underlying our research is to implement a HWR system that uses a digitizer in the form of a pen that transmits data online during the writing process. To train such a recognizer for efficient recognition, we need a sophisticated dataset.

The main contribution of this paper is to share a large dataset adding a scientific value in the handwriting recognition domain. We present a dataset of alphabet characters written on plain paper in the form of time-series data collected from a digital ballpoint pen equipped with sensors, i.e., a STABILO DigiPen. The collection of data written on normal paper makes it easier to apply a writing recognizer without the need of other more expensive devices or specific writing surfaces. In addition, we implemented (most of) the previously used methods applied for OnHWR, i.e., Machine Learning (ML) classifiers such as k-Nearest Neighbour (kNN) and Support Vector Machine (SVM) and also Deep Learning (DL) methods such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), i.e., Long-Short-Term Memorys (LSTMs), on our dataset. This provides a solid baseline for future research and fosters reproducibility in this research area. Fig. 1 presents the whole pipeline including data recording, pre-processing, classifier training, and evaluation.

The remainder of this paper is structured as follows. Section 2 discusses similar datasets that are currently used in the field. A summary of available offline and online datasets in the handwriting recognition domain is provided, including the type, recording platform and size of data. Section 3 presents our main contribution: a novel dataset for online handwriting recognition. We present the digitizer used for data acquisition along with the detailed description of the collected dataset. Section 4 provides methods used to pre-process data, to extract features from it, and how to apply online recognizers, i.e., we describe our implementation of the ML and DL baseline models that aim at solving the classification problem that this dataset offers. We provide quantitative results and discuss them in Section 5. Section 6 concludes.

2 RELATED WORK

Over the last decade, online handwriting recoginition has shown promising results and high accuracy. The number of datasets and methods to evaluate HWR systems steadily increases. While offline HWR is already very advanced [14, 49, 74], the focus of research moves to gesture recognition [4, 25, 38, 43, 69, 76], human activity recognition [79], and online HWR systems [19, 35, 36, 39, 73].

To train an OnHWR system a dataset needs to fulfill a number of requirements. (1) The dataset should allow for both a WD and WI evaluation, as writing recognition has to be applicable to new writers without re-training the model. Here, the difficulty is to cover many writing styles, i.e., printed or cursive writing, holding of the pen, pressure of the pen, size of the characters, and influence of noise. For that, the age, gender, education and frequency of writing has to be distributed homogeneously over the dataset. (2) As the number of alphabet classes is high and the introduced noise of the sensors can vary to a high degree, we need a large number of writers to guarantee for an optimal learning procedure to be considered for the later evaluation of the classifiers. (3) Finally, the dataset must contain time-series data for online recognition captured at a high frame-rate in order to allow for high classification accuracies.

This section provides a review of offline and online recognition datasets in Sections 2.1 and 2.2 with a focus on the requirements on the dataset for an optimal online writing recognition. To better compare the available dataset we line out a summary in Table 1 which summarizes all offline and online datasets, their corresponding recording platform, the size of the data, and the corresponding evaluation methods. Due to the broad spectrum of associated applications, the diversity of patterns, and our main contribution on OnHWR systems, we split these datasets into digits, characters and words, gestures, and objects, shapes and symbols datasets.

2.1 Offline Datasets and Recognition Systems

The development of offline datasets started in early 1900's. The IAM [49] dataset is one of the most commonly used dataset and provides English words and sentences. The large NIST dataset [22, 75] and its variants SD-19 [29], MNIST and EMNIST [14] contain digits and characters, but suffer from high ceiling effects, i.e., less generalization leads to overfitting. Further datasets cover addresses, e.g., the CEDAR [33] dataset, and outdoor image texts, e.g., the SVT [74] dataset. More offline datasets are listed in Table 1. The classes of our OnHW-chars dataset are the same as the classes from the IAM [49] dataset, but the dataset was acquired on a whiteboard and not on paper. The OnHW-chars dataset is smaller in size compared to the NIST and MNIST datasets, but larger as other visual image datasets [49, 74].

Existing recognition systems differ regarding pattern representation (i.e., image templates, structural representations and feature vectors), drawing constraints, and decision-making processes. The datasets present a large diversity of content with very different properties. We differentiate the datasets between their number of classes, the available amount of training samples per class, and between WD and WI experimental settings. The recognition of some datasets are quite challenging because of the presence of different writing styles and noisy data, while some datasets enable an easier recognition [17]. Most of previous recognition systems focus on writing on an electronic device. This requires an expensive device, and the recognition system cannot be used on normal paper. Hence, we focus on pen-based recognition systems that have integrated sensors.

2.2 Online Datasets

In the following, we describe datasets that are a collection of digits, characters and words more related to our OnHW-chars dataset. The LaViola [40] dataset has been written by 11 persons with a pen on a *TC 1100* tablet and covers trajectory-based digits, characters and mathematical symbols. They used an AdaBoost classifier and yield an accuracy between 90.9 % and 97.19 % for different recognizer configurations. Keshari et al. [37] achieved an accuracy up to 94.57 % on the mathematical expressions utilizing SVMs trained on standard and Chebyshev coefficient features.

The UJIpenchars/UJIpenchars2 [47] datasets contain 62/97 different classes of characters and symbols recorded by 11/60 writers on the *Toshiba Portégé M400 tablet* covering 1,364/11,640 samples. UNIPEN [30] is an ongoing project of collecting handprint and cursive handwriting on a pen-based computer from various alphabets including Chinese and Latin, pen gestures and signatures. The sentences dataset IAM-OnDB [45] covers

Table 1. Overview of state-of-the-art offline and online pen-based handwritten datasets. The writer-dependent (WD) and writer-independent (WI) column indicate the possibility of running WD and WI experiments. As we focus on OnHWR systems, we do not declare reported experiments for offline datasets.

Dataset	Classes	Recording platform	Size			WD/WI	Experiments
			writer	sample	class		
Offline datasets [78]							
NIST [22, 75]	Handwritten digits	Pen	3600	800,000	10	n.d.	n.d.
MNIST	Subset of NIST	Pen	–	70,000	10	n.d.	n.d.
EMNIST [14]	Digits and letters	Pen	–	445,600	36	n.d.	n.d.
Mathematics	Mathematical symbols	Pen	–	60,000	10	n.d.	n.d.
Devangari	Devangari characters	Pen	25	1,800	36	n.d.	n.d.
Arabic Text	Lexicon of words	Pen	–	113,284	–	n.d.	n.d.
Document	Lists, tables, formulas, diagrams and drawings	Handwritten documents	189	941	–	n.d.	n.d.
CEDAR [33]	Characters and digits	Pen	1,500	59,584	–	n.d.	n.d.
CENPARMI	Digits	Pen				n.d.	n.d.
IAM [49]	English word, sentences	Pen on a whiteboard	657	115,320	1,539	n.d.	n.d.
Street View Text (SVT) [74]	Outdoor image text from businesses	Harvested from Google street view	–	725	–	n.d.	n.d.
Online and gesture-based datasets (see Section 2.2) [16]							
Digits, characters and words datasets							
OnHW-chars	**English characters**	**Sensor-enhanced pen**	**119**	**31,275**	**52**	**WD/WI**	–
UNIPEN [30]	Latin alphabet, characters, words and sentences	Pen-based computers	–	12,000	–	–	–
PenDigits [1]	Handwritten digits	Wacom PL-100V	44	10,992	10	WD/WI	[1]
UJIpenchars / UJIpenchars2 [47]	Isolated handwritten characters	Toshiba Portégé M400 tablet PC	11 60	1,364 11,640	62 97	WD/WI WD/WI	– –
LaViola [40]	Digits, characters and math symbols	Tablet TC 1100 with pen	11	11,602	48	WD/WI	[17, 37, 40]
IME-OnDB [11]	Letters and gesture	Pocket PC with pen	14	6,636	18	WI	[12, 18]
IAM-OnDB [45]	Word instances	Electronic whiteboard	221	86,272	11,059	WD/WI	[45]
Gestures datasets							
Match-Up & Conquer [58]	Multi-touch gestures	Multi-touch display 3MTM C3266PW	16	5,155	22	WD/WI	–
NicIcon [51]	Gestural commands	Wacom Intuos2 A4	34	26,163	14	WD/WI	[7, 17, 68, 76]
Sign-OnDB [2]	Single-stroke pen gestures	Tablet with pen	20	33,150	17	WD/WI	[17, 25, 43]
unistroke [77]	2D single-stroke gestures	HP iPAQ h4355 with pen	10	4,800	16	–	[5, 44, 50, 67]
MMG [6]	2D multi-stroke gestures	Finger or pen on tablet	20	9,600	16	WD/WI	[6, 69]
Multitouch gesture [59]	Multi-touch symbolic gestures	Multi-touch display 3MTM C3266PW	18	7,200	30		–
ILGDB [60]	Single-stroke pen gestures	Tablet with pen	38	4,656	588	WD	[17, 43]
UsiGesture [9]	Gestures	Tablet with pen	30	18,300	61	WD/WI	[8–10]
Objects, shapes and symbols datasets							
HBF49 [17]	Features	Written with online device	–	–	49	WD/WI	[17]
Object sketches [20]	Object sketches	Multi-strokes	1,350	20,000	250	WD/WI	[41, 42]
HHReco [32]	Geometric shapes	Wacom Graphire2 pen	19	7,791	13	WD/WI	[17, 52, 53]
CVCsymb [62]	Architectural and electrical symbols	Digital pen	25	5000	50	WD/WI	–
IMISketchSDB [34]	Offline architectural symbols	Architectural plans	50	1,871	13	WI	–
HOMUS [13]	Online music notations	Galaxy Note with SPen	100	15,200	38	WD/WI	–

about 86,272 word instances from an 11,059 dictionary written by 221 writers via an electronic interface from a whiteboard. Their Hidden Markov Model (HMM) based approach achieves 65.9 % accuracy. The IME-OnDB [11] dataset is a good benchmark for evaluating relative positioning of handwriting, as several subsets of gestures have the same shape only distinguishable through their spatial context. To take relative positioning into account [11] exploits a fuzzy approach. The PenDigits [1] dataset is a collection of digits written by 44 users on the *Wacom PL-100V*. 10,992 samples covering the 10 digit classes and allow WD and WI experiments. Through the combination of static and dynamic Multi-Layer Perceptron (MLP) classifiers the accuracy can be increased by different fusion techniques, i.e., voting, mixture, stacking, boosting and cascading. The accuracy of the combined classifier dropped from 99.3 % for the WD testing set to 98.3 % for the WI testing set (see similarity to our results in Section 5).

The following datasets build a database for human gestures and are more related to human computer interaction. The NicIcon [51] dataset contains 26,163 of offline and online written iconic multi-strokes gestures (emergency situations, e.g., accident, fire), and includes pen-up movements and pressure meassures. Through a highly varying order and number of strokes, the dataset is quite noisy. WD and WI experiments exist: fusion of HMM-based and Zernike methods [7], a CKMeans with auto-completion algorithm [68], and MLP, SVM and Dynamic Time Warping (DTW) classifiers using global and stroke-level features [76].

For the Sign-OnDB [2] dataset 33,759 samples of single-stroke gestures are collected from 20 persons written on a tablet. Some of the 17 classes can only be distinguished based on dynamic information [17]. The ILGDB [60] dataset is a collection of single-stroke gestures recorded with a tablet. Each of the 28 writers provided 21 different gestures of their choice, and hence, in this dataset exists a large number of different samples unequally distributed over the classes. Consequently, only WD experiments exist [43]. The Match-Up&Conquer [58] multi-touch dataset is designed to address how users articulate gestures. Similar is the Multitouch gesture [59] dataset that covers 7,200 samples from 18 participants. 30 different gesture classes, e.g., circle, triangle, heart and cat, are unique in the number of strokes of the shape, number of fingers touching the surface, and bimanual or single-handed inputs. The unistroke [77] dataset consists of 16 different 2D single-stroke gestures, e.g., triangle, question mark and start. This dataset is evaluated by the $1 [77] and $N [5] recognizer, protractor [44], and DTW [50, 67]. The $N-Protractor [6] is derived from the $1 unistroke [77] recognizer that uses a closed-form template-matching method instead of an iterative search method in the $N [5] recognizer. They provided the Mixed Multistroke Gesture (MMG) [6] dataset representing 16 classes of 2D multistroke gesture symbols. UsiGesture [9] is a software support platform that accomodates multiple algorithms for pen-based gesture recognition. The goal is a dataset made of characters, symbols and commands, that allows to evaluate a gesture recognition algorithm depending on contextual variables, e.g., environment, platform and user.

Objects, shapes and symbols are addressed in the following datasets. The CVCSymb [62] dataset is a combination of online and offline architectural and electrical symbols. 5,000 samples have been drawn by 50 writers separated in two groups of 25 writers each. This results in total in 50 WD and WI classes. The HHReco [32] dataset consists of 7,410 samples in total of 13 different geometric shapes, i.e., circles, cylinder, archs, and polygons, written by 19 people on a *Wacom Graphire2* tablet. Ouyang et al. [53] use an image deformation model to achieve 98.2 % accuracy focusing on the visual appearance of the symbols.

The Object sketches [20] dataset is a collection of 20,000 unique sketches, e.g., teapot and car, evenly distributed over 250 object classes. They built upon a bag-of-features representation to extract local features and construct a visual vocabulary using kMeans clustering to train a SVM classifier, and achieved 56 % accuracy. Related is SHREC'13 [41] and SHREC'14 [42] that refer to sketch-based 3D shape retrieval containing 7,200/12,680 sketches and 1,258/ 8,987 3D models. The IMISketchSDB [34] dataset is an offline collection of 13 different architectural symbols, e.g., furniture, covering 1,871 samples from 50 plans. HOMUS [13] is the only dataset that addresses musical symbols, which consists of 15,200 offline and online samples from 100 users covering 32 symbol

classes. kNN, DTW and HMM are used for the online classification task, and kNN, NN, SVM and HMM are used for the offline classification task.

HBF49 [17] is a unique set of 49 features focusing on the universal representation space adapted to a large variety of symbols that can be used as a reference for evaluation of symbol recognition systems. They reported experiments with 1NN and SVM classifiers for eight different datasets [2, 32, 34, 40, 51, 60, 62, 70] for WD and WI cases.

A similar recording platform to the STABILO DigiPen is a phone with gyroscopes and accelerometers used in [19]. The GyroPen method reconstructs the trajectory of the phone's corner touching a writing surface for pen-like interactions. An online recognition system is used using an extended feature set to recognize the words with trajectory coordinates as input. Neelasagar et al. [35] also use the accelerometer and gyroscope signals from a smartphone for 3D handwritten character and gesture recognition. The acceleration signals are pre-processed with segmentation, filtering and normalization, while the gyroscope signals are lowpass filtered and normalized to get the orientation correction of the device.

Inertial pens that are closest to ours are the ones used in [36, 39], but the recorded dataset is not published. In these publications, accelerometer, gyroscope and magnetometer are integrated in a pen along with a microcontroller and a wireless transmission module that records movement data for writing alphabets and making gestures [36]. Unfortunately, the data acquisition unit is a large device that influences the style of the writer. Statistical features gave the best results in combination with a probabilistic NN and a kNN classifier. A similar device was constructed by [73] The recorded acceleration signals are calibrated, lowpass filtered, segmented and normalized, before aligning the signals with the 10 digit classes by a DTW method. WD (90.6 % accuracy) and WI (84.8 % accuracy) experiments are reported that are in the same range as our experiments on our OnHW-chars dataset (see Section 5).

Koellner et al. [39] use the STABILO DigiPen, which is the same device we used ourselves for the recording of the OnHW dataset, but their dataset is not published. The dataset consists of 20,000 English lowercase letters written from 15 users, and hence, created a dataset with more samples per writer per class than our dataset. WD and WI results for kNN, LDA, NB and LSTM classifier are reported.

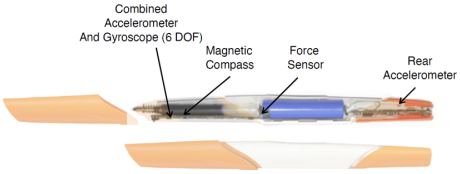
The recording platform of most of the online datasets are pen-based computers [11, 30, 62], tablets [1, 2, 6, 9, 32, 40, 47, 51, 58–60], phones [13, 77], or a whiteboard [45]. The classes of only some of these datasets [1, 11, 30, 40, 47] are similar to our OnHW-chars dataset, i.e., gesture-based [2, 6, 9, 51, 58–60, 77] and object-/symbol-based [13, 17, 20, 32, 34, 62] classification are related to other applications. As many other datasets, WD and WI is possible on OnHW-chars, but a large number of writers is necessary to evaluate for that in detail. While IAM-OnDB [45] (221) and object sketches [20] (1,350) have higher, all other datasets have lower number of writers than OnHW-chars. Similar to [2, 45, 51] OnHW-chars is in the upper scope of number of samples (31,275).

3 PROPOSED DATASET

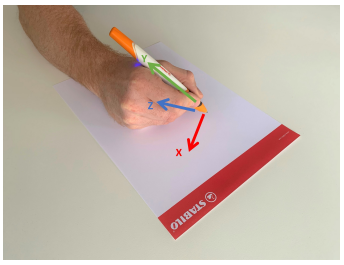
This section introduces our novel online handwriting (OnHW) dataset. We present our IMU-sensor enhanced ballpoint pen, i.e., the STABILO DigiPen, in Section 3.1, and describe the data acquisition constraints and calibration aspects in Section 3.2. In Section 3.3, we present our novel OnHW-chars dataset in detail.

3.1 Sensor-Enhanced Ballpoint Pen

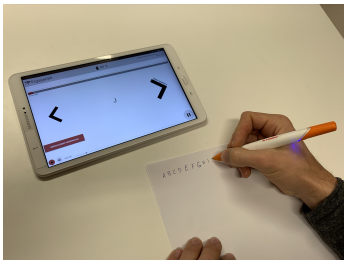
The STABILO DigiPen is a sensor-enhanced ballpoint pen with internal data processing capabilities, see Fig.2a. A Bluetooth module enables live streaming at 200 Hz to a connected device. The pen's overall length is 167 mm, its diameter is 15 mm, and it weighs 25 g. With its ergonomic soft-touch grip zone it is easy to use and feels comfortable and natural. Each DigiPen is equipped with a front accelerometer (STM LSM6DSL), a gyroscope



(a) Sensor-based STABILO DigiPen with integrated accelerometers, gyroscope, magnetometer, force sensor, and battery.



(b) Axis of the pen tip for the 6DoF accelerometer and gyroscope sensors.



(c) Data acquisition setup of the STABILO DigiPen connected to a tablet device for ground truth recording.

Fig. 2. The STABILO DigiPen.

(STM LSM6DSL), a rear accelerometer (Freescale MMA8451Q), a magnetometer (ALPS HSCDTD008A), and a force sensor (ALPS HSFPAR003A) [66].

The data recordings store 14 measurements provided by the sensors: two acceleration, one gyroscope and one magnetometer signals (each in X, Y, and Z direction, see Fig. 2b), the force with which the pen tip touches the surface, and the timestep at which the tablet receives the data from the pen.

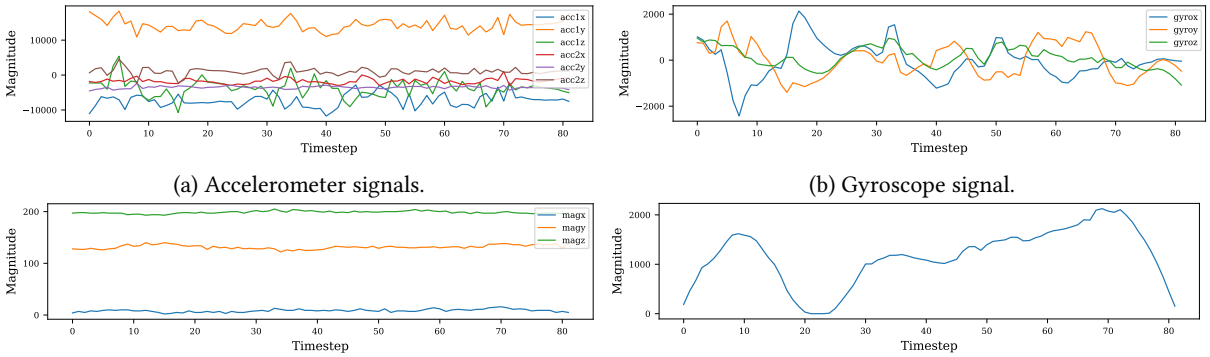
3.2 Data Acquisition and Calibration

We use a recording app provided by STABILO International GmbH to obtain the sensor data, which is connected to the DigiPen and tells the user which character to write (see Fig. 2c). Through this setup we also record the ground truth labels. We applied the following constraints for our data recording to achieve a homogeneous and equally distributed dataset. The writer has to sit on a chair in front of a table, and has to write on a normal, white paper (80 mg/m^2) padded by five additional sheets. There was no guideline concerning the size of the handwriting and the way of holding the pen, just the logo needs to face upwards. Users are allowed to write in a printed and cursive style.

Prior to recording we need to calibrate the pen with a short two-step procedure to determine the gyroscope and magnetometer biases and the magnetometer scaling. While placing the pen on the table for a couple of seconds, it is possible to find the gyroscope biases bg_x, bg_y and bg_z for each axis as the gyroscope values are supposed to be zero. Then, the pen should be rotated in all directions (covering a sphere). With the cloud of magnetometer points, we can calculate the sensor's bias bm_x, bm_y and bm_z (the sphere's center) and the sensor's scaling factor sm_x, sm_y and sm_z (the sphere's radii). More information can be found in [54, 61]. With these values, the raw values can be scaled and the bias removed. For each sensor the SI value without bias SI_{bias} can be computed with

$$SI_{bias} = \frac{raw_{value} - bias}{\frac{max}{max_{SI}} sm_*}, \tag{1}$$

where raw_{value} is the measured valued from the datasheet sensor, $bias$ is the bias from the calibration procedure (bg_x, bg_y and bg_z for the gyroscope, bm_x, bm_y and bm_z for the magnetometer, and 0 for the accelerometers and the force sensor), max is the maximal range value that is 32,768 of the front accelerometer and the gyroscope, 8,192 for the back accelerometer and the magnetometer, and 4,096 for the force sensor, max_{SI} is the maximal SI value that is $2\,g$ for both accelerometers, $1,000\,^\circ\,s^{-1}$ for the gyroscope, $2.4\,mT$ for the magnetometer, $5.32\,N$ for the force sensor, and sm_* is the measured scaling factor for the corresponding axis, otherwise it is 1 [66]. With



(a) Accelerometer signals.

(b) Gyroscope signal.

(c) Magnetometer signal.

(d) Force.

Fig. 3. Exemplary sample of the uppercase character 'B' for the front and back accelerometer (a), gyroscope (b), magnetometer (c), and force sensor (d).

Table 2. Overview of the number of samples of the OnHW-chars dataset including lowercase and uppercase characters from 119 writers for a writer-dependent (WD) and writer-independent (WI) evaluation.

	Total		Samples WD/WI	
Dataset	**Writer**	**Samples**	**Training**	**Testing**
Lowercase characters	119	15,650	11,542	4,108
Uppercase characters	119	15,625	11,517	4,108
Total	–	31,275	23,059	8,216

the calibration procedure from [61] the bias of the accelerometer cannot be determined, and hence, we set it to 0. We use the raw data in Section 4.

Fig. 3 shows exemplary raw signals of a written character 'B' (note that there is a total number of 82 timesteps). As the letter is constructed of two strokes, the pen is lifted one time and the measured force is 0 N between timestep 21 and 24, see Fig. 3d. We describe a proper pre-processing of such signals in Section 4.1.1.

3.3 The OnHW-chars Dataset

In the future, our OnHW dataset consists of several sub-datasets. In this paper, we first provide the OnHW-chars dataset that consists of lowercase and uppercase characters. The recording of further datasets is an ongoing project and will be continuously increased for a more profound and detailed evaluation. Words, sentences, symbols and numbers from the same writers will be published in a later stage. Our DigiPen records data measurements at 100 Hz. For each writer, three `.csv` files are provided: One file that contains the calibration data, one file that contains the character labels with the start and end timesteps, and one file that contains the 13 measurements for each timestep. The OnHW dataset is publicly available for download here: https://stabilodigital.com/onhw-dataset/.[1]

For the novel OnHW-chars dataset 119 right-handed persons wrote the English alphabet for six times both in lowercase and uppercase letters. All writer are grown-up and above the age of 18, but the exact age was not reported due to anonymity. The ratio between women and men is 45 % women and 55 % men. This allows to solve classification problems with 52 classes. In total, this resulted in 312 samples per person (with some small

[1] Alternative download link: https://iis.fraunhofer.de/onhw-dataset/

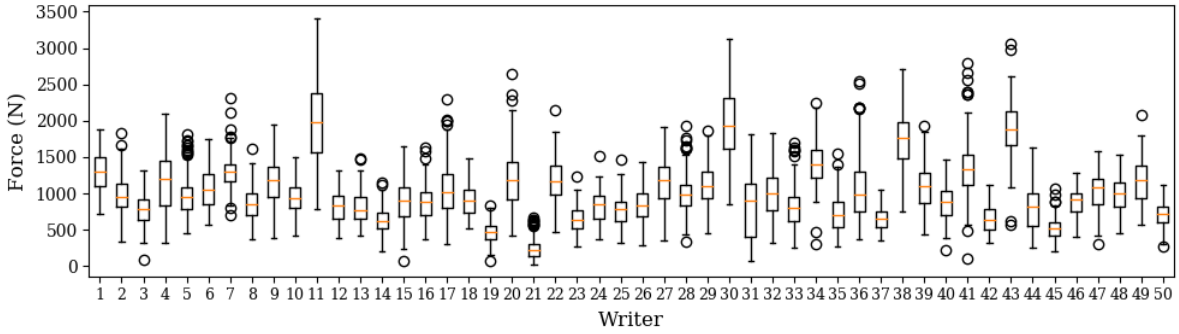


Fig. 4. Analysis of the writing properties of 50 participants. We provide the distribution of the force averaged over each sample.

Table 3. Analysis of the character properties. Presented are the average number of timesteps (TS) and strokes (S) and their deviations (D_{TS}, D_S) for every character.

Char.	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
TS	45.4	45.4	26.8	50.7	39.5	49.2	53.9	41.6	40.1	53.7	53.9	27.1	58.3	40.0	33.7	50.8	53.7	32.4
D_{TS}	32.6	28.7	21.5	29.7	27.8	18.1	23.3	17.7	21.6	24.7	27.6	21.3	33.5	20.8	17.5	37.3	25.5	16.6
S	1.07	1.04	1.01	1.12	1.03	1.58	1.03	1.01	1.86	1.91	1.52	1.01	1.06	1.01	1.01	1.15	1.21	1.01
D_S	0.63	0.24	0.12	0.43	0.43	0.66	0.25	0.11	0.71	1.07	0.74	0.18	0.51	0.13	0.19	0.48	0.51	0.12
Char.	s	t	u	v	w	x	y	z	A	B	C	D	E	F	G	H	I	J
TS	36.0	46.7	37.7	33.2	50.8	44.2	48.2	52.8	60.9	71.2	30.4	56.2	74.2	67.8	56.3	65.6	27.7	45.6
D_{TS}	20.2	19.2	19.9	19.2	28.1	21.5	39.9	26.7	30.9	25.2	21.9	26.9	26.1	36.6	25.8	22.7	27.5	25.9
S	1.01	1.81	1.04	1.01	1.00	1.79	1.35	1.62	1.70	1.61	1.09	1.73	2.73	2.53	1.18	2.45	1.10	1.06
D_S	0.18	0.68	0.47	0.13	0.07	0.73	0.94	0.65	1.13	0.66	1.55	0.77	1.27	1.15	0.48	1.19	0.50	0.46
Char.	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
TS	59.7	33.4	60.7	55.1	37.1	52.4	65.3	63.6	39.5	47.5	40.1	35.4	58.1	47.8	52.0	60.0		
D_{TS}	26.5	20.1	22.5	29.9	22.7	22.4	24.3	28.4	31.3	19.7	24.2	18.6	35.1	23.0	38.2	31.4		
S	1.83	1.04	1.21	1.27	1.01	1.43	1.86	1.39	1.01	1.85	1.01	1.01	1.08	1.81	1.58	1.64		
D_S	0.93	0.38	0.56	0.61	0.11	0.62	0.71	0.61	0.13	0.70	0.11	0.12	1.21	0.64	0.72	0.69		

deviations). An overview of the sample numbers are given in Table 2. There are 15,650 lowercase characters and 15,625 uppercase characters. The complete `OnHW-chars` dataset consists of 31,275 samples in total. Consequently, the `OnHW-chars` dataset is a large dataset of 119 writers to evaluate for a large diversity of properties, i.e., different writing style (e.g., printed or cursive characters), holding of the pen, pressure of the writer on the pen, and influence of noise (e.g., bias and scaling) on the classification accuracy.

As `OnHW-char` is an online dataset it is possible to evaluate for time-series based recognition methods, i.e., incorporate the drift of the sensors. Constructing a WI recognizer is a much more challenging task as constructing a WD recognizer. However, many applications only allow for WI recognizers, as the application does not allow for a re-training to a new writer before using the pen in many cases. For the WD evaluation we split the dataset in 90 recordings for training (23,059 samples in total), and 29 recordings for testing (8,216 samples in total). This corresponds to a split of 73.73 % for training and 26.27 % for testing for the WI case (see Section 5). To better evaluate for the WD and WI tasks, Fig. 4 shows writing properties for 50 different writers, e.g., the writers 11, 30 and 43 put high pressure on the pen, while the writer 19, 21 and 45 put very low pressure on the pen.

Table 3 presents an analysis of the character properties. If the number of timesteps and strokes are highly different between the characters, the features of such samples might be better separable for ML-based classifiers. Obviously, the trajectory for uppercase characters is longer, and consequently, the average timesteps (TS) the writer requires for lowercase characters are 44.1, while the average timesteps for uppercase characters are 52.1. For example, the characters 'B' (71.2), 'E' (74.2), 'F' (67.8) and 'H' (65.6) require more time to write than, e.g., 'c' (26.8), 'l' (27.1) and 'I' (27.7), as they are constituted by more strokes. Lowercase characters are constituted of 1.24 strokes on average, while uppercase characters are constituted of 1.50 strokes, e.g., the characters 'c', 'h', 'o', 'r', 's', 'v' and 'w' are always written in one stroke. The standard deviations D_{TS} and D_S indicate a high difference in writing style of a character, e.g., the stroke deviation is 0.44 for lowercase and 0.69 for uppercase characters on average.

Classifying characters from right-handed and left-handed writers from one single signal-based dataset is a quite challenging task as the pen rotation is significantly different. Hence, we decided to exclude left-handed recordings for now.

4 PROPOSED BASELINE CLASSIFIERS

There is an exhaustive literature that deals with the classification of characters, gestures, symbols and objects gatherd from a 2D tablet-based recording platform. Popular methods include Dynamic Time Warping (DTW) [13, 50, 51, 67, 73, 76, 77], k-Nearest Neighbors (kNN) [13, 17, 20, 36, 39, 60], Support Vector Machines (SVMs) [17, 20, 32, 37, 51, 60, 76], Hidden Markov Models (HMMs) [7, 13, 28, 45, 46] and Neural Networks (NN) [1, 1, 28, 36, 39, 46, 76]. Indeed, research that addresses a signal-based handwritten text analysis gathered from a digital pen comparable to our platform is very rare.

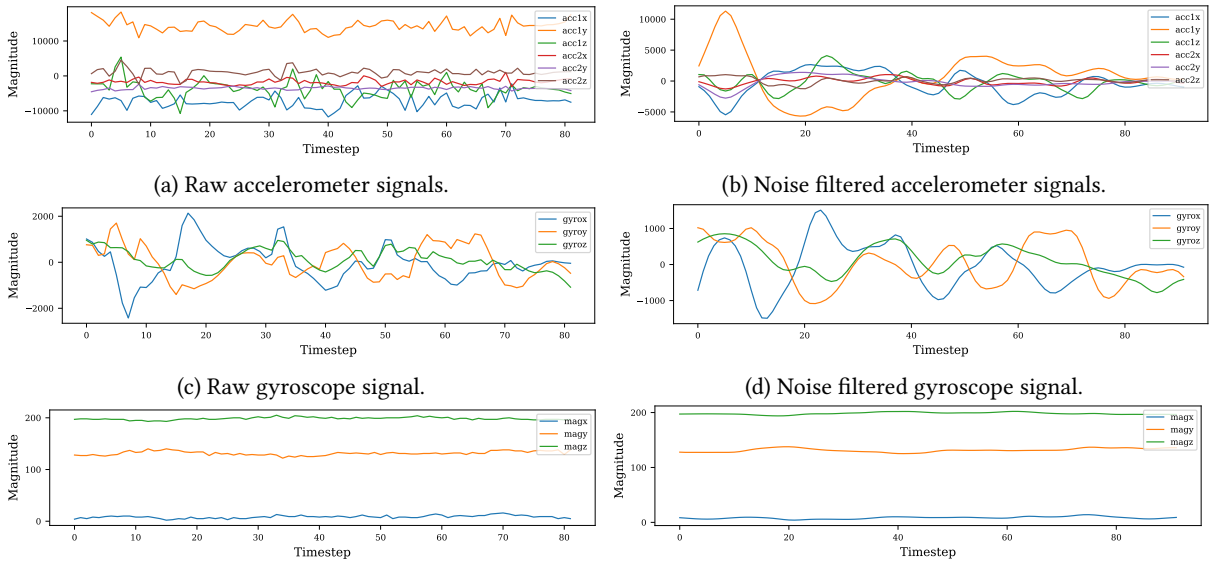
Online character recognition is based on the analysis of a given sequence of strokes applied over time. Such analysis usually pre-processes the input signals (by noise filtering), and then extracts features that allow for a recognition of written characters. In this section we present these steps and apply character classification over the proposed dataset. Given the unavailability of any previous results of classifying the complete alphabet letters, we run the following experiments over the separated uppercase and lowercase letters, hence classifying 26 different classes. In addition, we present the results of applying the classifiers over the complete 52 character classes. We present results for classical ML models in Section 4.1, and for DL models that use the raw input data to classify the written characters in Section 4.2.

4.1 Character Classification using Classical Machine Learning Models

We implemented pre-processing steps for applying different ML algorithms to evaluate how accurately different models classify the alphabet characters. As a pre-processing step we applied noise filtering to reduce the noise within the data (see Section 4.1.1). Using the filtered data, we extracted different features (see Section 4.1.2), and used an autoencoder for automatic feature extraction (see Section 4.1.3) as a representation of the information in the data to be used in the classification algorithms (see Section 4.1.4).

4.1.1 Pre-processing. Sensor noise represents the random variation in its output when functioning under static conditions. Hence, our raw sensor data output usually contains distorted signals (due to the paper surface inconsistency and trembling during writing). Pre-processing helps to remove insignificant or redundant information, which helps to extract high quality features from the signal streams. Commonly used pre-processing techniques include resampling, normalization, segmentation, and filtering [1, 35, 39, 45, 48, 72, 73]. More sophisticated approaches such as Butterworth filters and Savitzky-Golay pre-processing have been used in [23, 24, 39].

We apply a high pass filter with a cutoff frequency of $1\,Hz$ to remove the gravitational acceleration from the accelerometer recordings. Gravity is a constant force and the high pass filter allows us to keep the fast changing forces applied when recording while filtering the slow changing gravitational force. To disregard the noise within



(a) Raw accelerometer signals.

(b) Noise filtered accelerometer signals.

(c) Raw gyroscope signal.

(d) Noise filtered gyroscope signal.

(e) Raw magnetometer signal.

(f) Noise filtered magnetometer signal.

Fig. 5. Comparison of raw sensor signals (a,c,e) and noise filtered signals (b,d,f).

the raw data, we use a moving filter with a window of size 11, which acts as a low pass filter that allows the removal of high frequency noise from the input data. Since random noise usually includes random jumps in the data signals, the filter allows signal smoothing. Fig. 5 shows a recorded letter sample signal before (left) and after (right) pre-processing.

4.1.2 Manual Feature Extraction. Feature extraction is the concept of deriving a new set of inputs from the original raw dataset that represents valuable information of the data in a format that best fits an ML algorithm. Well established statistical features include the mean, standard deviation, variance, mean absolute deviation, location of zero crossings, signal range, and minimal and maximal values. Fast Fourier Transform (FFT), Autocorrelation Function (ACF) and Wavelets (WFLT) are used in [39]. For trajectory-based classification techniques static features (box aspect ratio, length, curvature, area of convex hull, closure, perpendicularity, ratio of the principal axes, etc.) and dynamic features (initial angle, position of first and last points, etc.) are important [51, 60]. Furthermore, pressure data available in online data is also important, i.e., average pressure and pen down count [51]. For a very good overview and discussion on different features we refer the reader to [17].

In our system and dataset we use the two accelerometers and the gyroscope to extract multiple time and frequency domain features that would allow a higher recognition rate. We extract the features per channel and concatenate the resulting feature vectors forming the final feature vector that is used for the character classification. The extracted features are mainly statistical and geometrical features of the raw signals. For the time domain features we used the maximum, the minimum, the mean value, the standard deviation, and the correlation coefficients of each of the axes. We also include the skewness (i.e., that describes the lack of symmetry of the data distribution), interquartile range (i.e., a measure of statistical dispersion), median absolute deviation (i.e., a measure of deviation from the median of the data), and area under curve. For the frequency domain features, we apply Fast Fourier Transform (FFT) to compute the Discrete Fourier Transform (DFT), then extract the previously stated features, in addition to including the weighted mean of the frequency distribution, the DFT

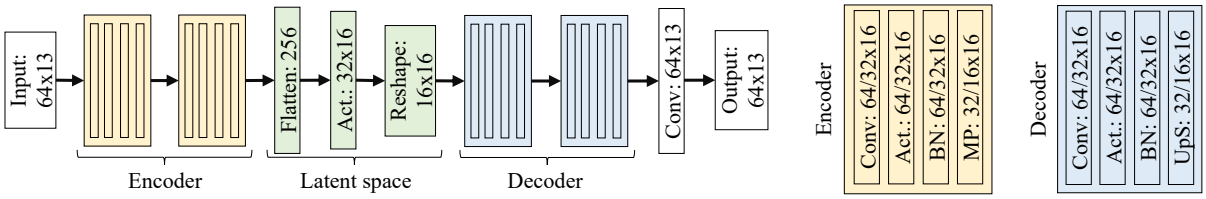


Fig. 6. Network structure of the Autoencoder. The input is encoded and the latent space representation decoded. Layers: Conv = Convlution; Act. = Activation; BN = BatchNormalization; MP = MaxPooling1D; UpS = UpSampling1D.

coefficients, the local maxima of DFT coefficients, and their corresponding frequencies. The final feature vector is composed of 327 features of the concatenated channel feature vectors.

4.1.3 Automatic Feature Extraction. As there is no best practice on standard features that are usually considered best for online character classification, we also investigate the use of an autoencoder to automate the extraction of a feature vector as the extraction of manual, hand-crafted features has its drawbacks. The number of features to extract, the relation between the extracted features, and which specific features are useful for specific cases, still have no precise solution, when considering the research done in this domain. Thus, an automated feature extraction process allows for better feature vector extraction from the data and get better classification accuracy.

An autoencoder is a neural network that efficiently applies the task of representation learning. It transforms data into a compressed knowledge and information representation, producing a feature vector that represents the information contained in a sample of the data. Additionally, as an autoencoder learns to compress the dimensionality of the data into a specific sized feature space, it learns how to ignore the noise in the data, thus allowing the use of the raw data with the minimal need for pre-processing steps.

CNNs are well known having the capability to extract features, and are popular specifically when working with image datasets as implementations of 2D CNNs. We use CNNs as an architecture for an autoencoder and apply 1D CNN implementations for the time-series data that is recorded from the sensors, allowing the extraction of a feature vector of defined dimensions automatically that represents a sample information in the defined vector dimensions. This information includes the 13 channels of the data, representing the four triaxial sensors, and the force sensor. To fit the data into a CNN, it is necessary for all the samples of the dataset to have the same number of timesteps, with a sample being defined as a letter recording. Given the different time of recording per letter, each sample is resampled into a defined number of timesteps equal to 64. This is chosen to allow the extraction of a feature space vector of size 256. This feature space dimension is assumed to be sufficient to allow for better classification. Fig. 6 shows the architecture of the autoencoder and the defined dimensions per layer of the network.

4.1.4 ML-based Character Classification. Following the extraction of the specific features from the data, we use the complete feature vector as an input into several ML approaches to classify the written characters based on the features of the sensor data. We apply several classifiers using Python libraries. Online character classification is mainly based on techniques like kNN, HMM, SVM, LDA and NB.

As our baselines, we implemented Decision Tree, Random Forest, Logistic Regression, Linear SVM, and kNN. We perform grid search for the optimal hyperparameters that We line out in the following. **Decision Trees** (DTs) use tree-like structures of decisions and the possible consequences, in which each internal node represents a test on an attribute. The branch represents the evaluation of the test. The leaf node represents the class. We use DTs with default parameters, i.e., maximal depth and maximal leaf nodes are set to *None*. A **Random Forest** (RF) is

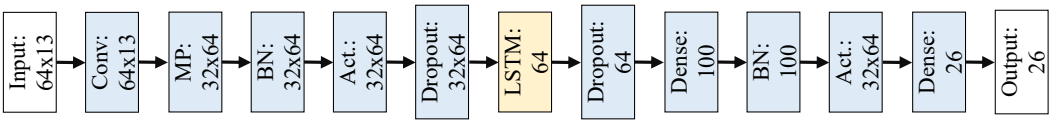


Fig. 7. Network structure of the CNN+LSTM. Layers: Conv = Convlution; Act. = Activation; BN = BatchNormalization; MP = MaxPooling1D.

an ensemble learning method that constructs a multitude of DTs while training. We apply a RF classifier with 100 trees, no defined max depth, a minimal sample split of 2, and minimal samples leaf of 1. For the **Logistic Regression** (LR) classifier we use a L2 norm for penalization, set the parameter $C = 1$, and set the tolerance for stopping criteria to 0.0001. We run the *lbfgs* solver maximal 100 iterations. **kNN** classifiers are non-parametric methods where the sample class is predicted by a plurality vote of its neighbors. If $k = 1$, the sample is assigned to the class of the single nearest neighbor. kNN is used in [17, 20, 36, 39] ($k > 1$) and in [60] ($k = 1$). We apply the kNN classifier considering five nearest neighbors ($k = 5$), we set the leaf size to 30 and weights to *uniform*. **Support Vector Machines** (SVM) classifiers are non-probabilistic classifier that is a representation of the samples as points in space, such that the samples of the separate categories are divided by a clear gap. For an SVM, also used in [17, 20, 32, 51, 60], a kernel with a gamma parameter and a slack variable has to be set. In our configuration, we use the slack variable $C = 1$, the tolerance 0.0001 and the L2 norm as penalty function, and trained maximal to a 1,000 iterations.

The stated classifiers are different methods that consider different attributes and approaches for applying classification over the available data features. As a result, these methods would produce different classification results and accuracies based on how the extracted features are functional for each classifier.

4.2 DL-based Character Classification

Classical classification approaches require the process of feature extraction from time-series data to train ML models. The feature extraction difficulty lies in the limitations of the expertise in that specific field. Autoencoders are established to be automated feature extraction methods, but in a two-stage training process, they are, however, not informed about the final classification task, and have hence no access to the complete information. Therefore, we present end-to-end DL methods that provide state-of-the-art results with no feature extraction. Similarly to fitting the data into the autoencoder, that data was resampled to have a fixed length of timesteps providing a form to fit the data into the different types of networks.

Liwicki et al. [46] used RNNs, i.e., a bidirectional Long-Short-Term Memory (BiLSTM), with the Connectionist Temporal Classification (CTC) [27] objective function for online whiteboard handwriting recognition (74.0 % accuracy), and showed an improvement over HMM-based systems (65.4 % accuracy) on the `IAM-OnDB` [45] dataset. LSTMs [31] are RNN architectures designed to bridge long time delays between relevant input and target events. BiLSTMs [64] are able to incorporate context on both sides of every position in the input sequence, e.g., in word recognition where the information left and right of a given letter is useful. The RNN approach of [28] achieved 79.7 % accuracy on the online word recognition task. Dynamic and static neural networks are used in [1].

We implement the CNN, LSTM and BiLSTM networks with a similar design using different architectures. The design includes two layers of the architecture with a 40 % dropout rate, a fully connected layer with 100 units, followed by the output layer including the number of classes, 26 classes for either lowercase or uppercase letter classification, and 52 classes for the complete alphabet classification (see Fig. 7). The LSTM and BiLSTM hidden layers include 64 units each. For the CNN hidden layers, we use a configuration of 64 feature maps, and a kernel size of 4 with max pooling of size 2. We use a rectified linear unit activation in the hidden layers with the Softmax activation function in the output classification layer. The cross entropy loss function is applied with Adam optimization using a 0.001 learning rate.

Table 4. Evaluation results. Accuracies are given in % for different classifier.

Method			Lowercase		Uppercase		Combined	
			WD	WI	WD	WI	WD	WI
ML-based	Features	Random Forest	54.77	43.04	56.29	45.96	42.39	30.44
		Decision Tree	29.36	22.89	29.87	24.32	19.19	14.96
		Logistic Regression	55.36	49.60	58.54	53.26	43.95	39.11
		Linear SVM	61.55	51.07	63.70	54.00	48.77	38.71
		kNN	49.17	29.87	51.32	30.94	38.30	19.61
ML-based	Autoencoder	Random Forest	58.02	45.55	63.19	43.73	43.60	43.62
		Decision Tree	30.49	21.73	33.23	19.68	20.32	20.33
		Logistic Regression	56.16	44.93	62.59	43.73	41.66	41.66
		Linear SVM	62.09	51.80	70.61	51.74	46.54	46.56
		kNN	42.43	34.09	57.49	36.68	33.08	33.08
DL-based		CNN	**84.62**	**76.85**	**89.89**	**83.01**	**70.50**	64.01
		LSTM	79.83	73.03	88.68	81.91	67.83	60.29
		CNN+LSTM	82.64	74.25	88.55	82.96	69.42	**64.13**
		BiLSTM	82.43	75.72	89.15	81.09	69.37	63.38

5 RESULTS AND DISCUSSION

In this section we report results for both cases presented in Section 3.3, i.e., writer-dependent and writer-independent recognition, with the training and test dataset splits as shown in Table 2. Considering the WI case, the datasets are split based on writers, keeping the writers in the test dataset completely different from the ones in the training dataset, while in the WD case, a single writer could be included in both datasets. For the WI task, we present averaged results for a 5-fold cross validation. Table 4 shows the performance of the baseline classifiers described in Section 4. We see that (in most cases) classical ML models perform slightly better when they get presented feature vectors from the autoencoder model. While this is not at a significant level it still shows that hand-crafted engineering of features is unnecessary. Among all the ML models, the linear SVM performs best. However, yet the recognition rates of the SVM only reaches an accuracy of 71 % over the WD recognition. The other classical ML models, i.e., the DT and RF models, yield much lower results over the test dataset due to early overfitting of the models during the training process (they reached a 100 % recognition rate over the training dataset).

The best classification accuracy is obtained with the CNN model for almost all of the different cases. The CNN model reaches almost 85 % accuracy for the lowercase WD task, and 77 % accuracy for the lowercase WI task. The recognition rate increases to almost 90 % when classifying uppercase characters in the WD case, and to 83 % correct classification in the WI recognition. The results of the CNN+LSTM model and the BiLSTM model are similar, with slight differences between the different cases. The LSTM provides the lowest accuracies when dealing the WI recognition.

The results show that state-of-the-art DL methods produce more accurate classification results than classical ML methods, even when considering an automated feature extraction method. We can also see that, in most cases, the best recognition rate is obtained at the uppercase letter classification. The accuracy drops for all cases when extending the classification into the complete 52 classes, as there are several characters that differ only in size and not in number of strokes, e.g., 'C'/'c', 'U'/'u', 'W'/'w', 'X'/'x', and 'Z'/'z', see the secondary diagonal of the confusion matrix in Fig. 8 for the CNN model for the WI combined case. The recognition rate is higher for the WD case in direct comparison to the WI case, showing that it is a more challenging task to accomplish as stated in Section 3.3.

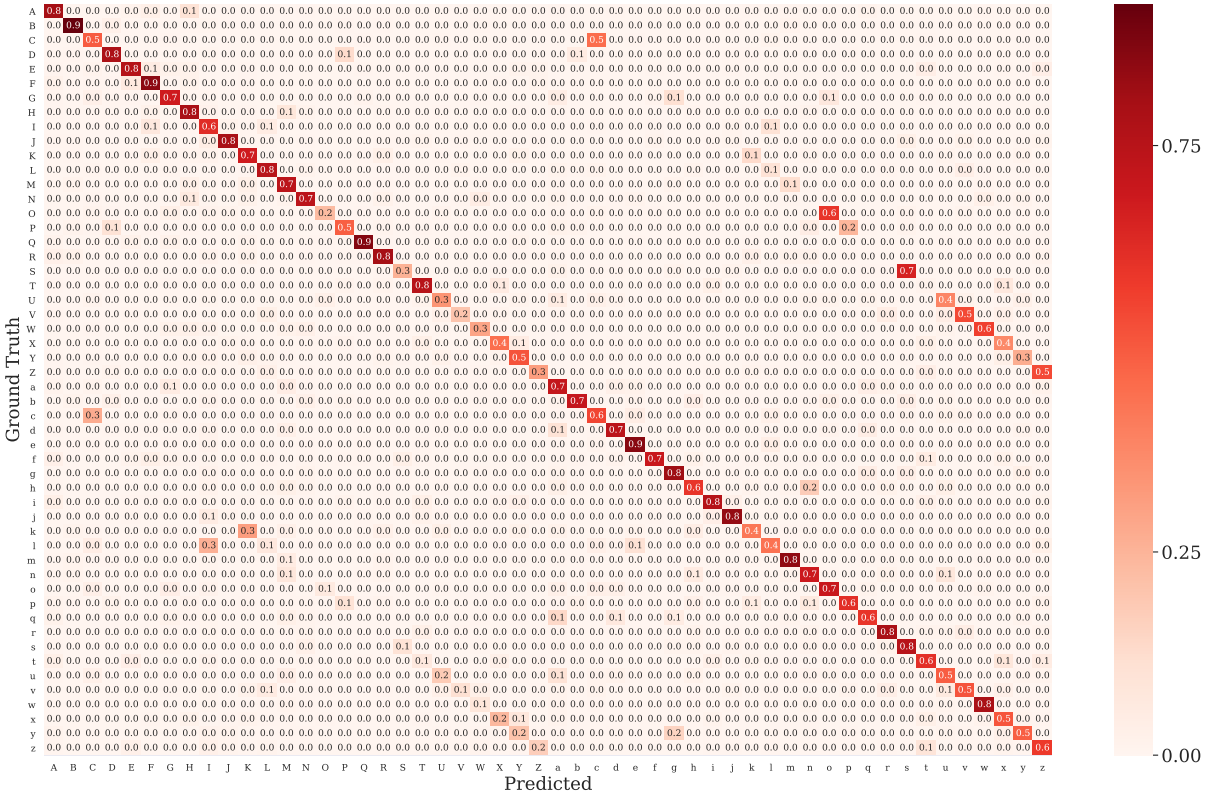


Fig. 8. Confusion matrix for predicted and ground truth WI combined letters. Presented are the CNN results.

The results presented can be improved with further investigation of the best hyperparameters in the models, and only serve as a baseline. When considering ML methods, better modeling of the data can be obtained using time-series analysis features that are not considered in our experiments, e.g., Wavelets and Shapelets. The dimensions of the feature vector, along with the autoencoder parameters can be improved for this classification task. Other DL models can be tested over the dataset with deeper hyperparameter optimization study to improve the recognition rate, specifically for the combined letter classification.

We evaluated the impact of each sensor on the results by training the models and leaving the data of one sensor out. The data of the magnetometer does not improve the classification accuracy. We publish the dataset including the magnetometer data for a possible investigation of further research.

The accuracies obtained with these experiments indicate that the presented dataset fulfills the requirements that are necessary for applying a writing recognition system stated in Section 2. The number of writers that contributed to the dataset allows a high recognition rate, and specifically grants the possibility of applying a recognition system that is able to recognize the handwriting of previously unseen users without having prior interaction or data. This makes the system a completely WI recognizer.

Since the number of contributions per writer to the dataset are approximately the same, the presented dataset shapes the 52 classes of the alphabet letters in balanced mode. This allows an implemented system to better distinguish between the available classes with less confusion between the letters to be recognized, specifically

when dealing with similarly written characters. Although, the accuracy drops slightly for such characters, e.g., 'U' (75 % accuracy) and 'V' (69 % accuracy). The attribute similarity shown in Table 3 underlines that the average number of strokes are 1.01 for both letters, and the stroke deviation is small. Also the characters 'X' (66 % accuracy) and 'T' (82 % accuracy) are confused, as the recording attributes (X: $TS = 47.8$, $D_{TS} = 23.0$, $S = 1.81$, and $D_S = 0.64$, T: $TS = 47.5$, $D_{TS} = 19.7$, $S = 1.85$, and $D_S = 0.70$) are similar. Furthermore, placing no restrictions on the writers during the data recording sessions, such as writing speeds, directions and sizes, made the data as natural as possible. This allows the implemented systems to generalize the recognition to several different writing styles.

Additionally, using solely a sensor-enhanced pen to collect the data grants the possibility for the extension of the dataset, since no other devices are required in the process. Using the `OnHW-chars` dataset allows the implementation of a WI handwriting recognizer that only requires the use of a sensor-enhanced DigiPen.

6 CONCLUSION

In this paper, we addressed the handwriting recognition task and the available public datasets that are popular in the scientific community. We summarized available offline and online handwriting datasets, and made an in-depth comparison to our novel `OnHW` dataset that includes data for writing alphabet characters on regular paper. The dataset was collected using the STABILO Digipen. It consists of 31,275 letter samples, distributed into 15,650 lowercase and 15,625 uppercase letters collected from 119 writers who contributed approximately equally to the dataset. The dataset provides a time-series representation of sensor signals that recorded the pen movement during writing, which include linear accelerations, angular velocities and magnetic field recordings that help in identifying the angle at which the pen was held, along with the force applied by the pen on the paper to identify when writing and hovering occurs. To the extent of our knowledge, there are several attempts for applying online handwriting using sensor-enhanced pens, however no data used within these projects that covers character level recognition was made publicly available. This presented dataset forms a novel benchmark for future research to further improve online handwriting recognition, specifically character classification while writing on normal paper.

In addition, we implemented a series of experiments for the online character classification task, applied over different subsets of the dataset, based on multiple ML and DL algorithms, which are widely used in the time-series classification domain. We draw benefits of data pre-processing, feature extraction, and letter classification. The experimental results showed that CNNs achieve the best results when classifying characters over different subsets, achieving accuracies of 90 % for the WD and 83 % for the WI classification task on average. These presented models serve as benchmark models that can be used in the scientific community when applying character classification using sensor data provided from a pen.

ACKNOWLEDGMENTS

This work was supported by the Federal Ministry of Education and Research (BMBF) of Germany by the research program Human-Computer-Interaction through the project "Schreibtrainer" (grant number 16SV8228), and by the Bayerisches Staatsministerium für Wirtschaft, Landesentwicklung und Energie which is part of the EINNS project (Entwicklung Intelligenter Neuronaler Netze zur Schrifterkennung) (grant number IUK-1902-0004).

REFERENCES

[1] Fevzi Alimoglu and Ethem Alpaydin. 1997. "Combining Multiple Representations and Classifiers for Pen-based Handwritten Digit Recognition". In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*, Vol. 2. Ulm, Germany, 637–640.

[2] Abdullah Almaksour, Eric Anquetil, Solen Quiniou, and Mohamed Cheriet. 2010. "Personalizable Pen-Based Interface Using Lifelong Learning". In *Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR)*. Kolkata, India, 188–193.

[3] María A. Pèrez Alonso. 2015. "Metacognition and Sensorimotor Components Underlying the Process of Handwriting and Keyboarding and Their Impact on Learning. An Analysis from the Perspective of Embodied Psychology". In *Proc. Social and Behavioral Sciences*.

[4] Christoph Amma, Dirk Gehrig, and Tanja Schultz. 2010. "Airwriting Recognition using Wearable Motion Sensors". In *Intl. Conf. on Augmented Human (AH)*. Megève, France.

[5] Lisa Anthony and Jacob O. Wobbrock. 2010. "A Lightweight Multistroke Recognizer for User Interface Prototypes". In *Proc. of Graphics Interface (GI)*. 245–252.

[6] Lisa Anthony and Jacob O. Wobbrock. 2012. "$N-Protractor: A Fast and Accurate Multistroke Recognizer". In *Proc. of Graphics Interface (GI)*. 117–120.

[7] Relja Arandjelović and Tevfik Metin Sezgin. 2011. "Sketch Recognition by Fusion of Temporal and Image-based Features". In *Journal of Pattern Recognition*. 1225–1234.

[8] François Beuvens and Jean Vanderdonckt. 2012. "Designing Graphical User Interfaces Integrating Gestures". In *Intl. Conf. on Design of Communication*. Seattle, CA, 313–322.

[9] François Beuvens and Jean Vanderdonckt. 2012. "UsiGesture: an Environment for Integrating Pen-based Interaction in User Interface Development". In *Intl. Conf. on Research Challenges in Information Science (RCIS)*. Valencia, Spain, 1–12.

[10] François Beuvens and Jean Vanderdonckt. 2014. "UsiGesture: Test and Evaluation of an Environment for Integrating Gestures in User Interfaces". In *Intl. Journal of Human-Computer Interaction*, Vol. 7(2). 139–160.

[11] François Bouteruche, Sébastien Macé, and Eric Anquetil. 2006. "Fuzzy Relative Positioning for On-Line Handwritten Stroke Analysis". In *Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*. La Baule, France.

[12] Martin Bresler, Daniel Prusa, and Vàclav Hlavàc. 2015. "Detection of Arrows in On-Line Sketched Diagrams Using Relative Stroke Positioning". In *Winter Conf. on Applications of Computer Vision (WACV)*. Waikoloa, HI, 610–617.

[13] Jorge Calvo-Zaragoza and Jose Oncina. 2014. "Recognition of Pen-Based Music Notation: the HOMUS dataset". In *Intl. Conf. on Pattern Recognition (ICPR)*. 3038–3043.

[14] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andrè van Schaik. 2017. "EMNIST: Extending MNIST to Handwritten Letters". In *Intl. Joint Conference on Neural Networks (IJCNN)*. Anchorage, AK, 2921–2926.

[15] Dalbir and Sanjiv Kumar Singh. 2015. "Review of Online & Offline Character Recognition". In *Intl. Journal of Engineering and Computer Science (IJECS)*, Vol. 4(5). 11729–11732.

[16] Adrien Delaye. [n.d.]. "Pen and Touch Datasets". https://sites.google.com/site/adriendelaye/home/pen-and-touch-datasets

[17] Adrien Delaye and Eric Anquetil. 2013. "HBF49 Feature Set: A First Unified Baseline for Online Symbol Recognition". In *Intl. Conf. on Pattern Recognition (ICPR)*, Vol. 46(1). 117–130.

[18] Adrien Delaye, Sébastien Macé, and Eric Anquetil. 2009. "Modeling Relative Positioning of Handwritten Patterns". In *Intl. Conf. on Graphonomics Society (IGS)*. Dijon, France.

[19] Thomas Deselaers, Daniel Keysers, Jan Hosang, and Henry A. Rowley. 2015. "GyroPen: Gyroscopes for Pen-Input with Mobile Phones". In *Trans. on Human-Machine Systems*, Vol. 45(2). 263–271.

[20] Mathias Eitz, James Hays, and Marc Alexa. 2012. "How Do Humans Sketch Objects". In *Trans. on Graphics*.

[21] Maged M. M. Fahmy. 2010. "Online Signature Verification and Handwriting Classification". In *Journal on Ain Shams Engineering (ASEJ)*, Vol. 1(1). 59–70.

[22] Nikos Fakotakis, Ergina Kavallieratou, and Kokkinakis George. 2002. "Handwritten Character Recognition based on Structural Characteristics". In *Intl. Conf. on Pattern Recognition (ICPR)*, Vol. 3.

[23] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H. Siddiqui, Michael Philippsen, and Christopher Mutschler. 2020. RNN-Aided Human Velocity Estimation from a Single IMU. In *Journal of Sensors*, Vol. 20(13).

[24] Tobias Feigl, Christopher Mutschler, and Michael Philippsen. 2018. Supervised Learning for Yaw Orientation Estimation. In *Intl. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*. Nantes, France, 206–212.

[25] Vittorio Fuccella, Poika Isokoski, and Benoît Martin. 2013. "Gestures and Widgets: Performance in Text Editing on Multi-touch Capable Mobile Devices". In *Intl. Conf. on Human Factors in Computing Systems (CHI)*. 2785–2794.

[26] Sabrina Gerth, Annegret Klassert, Thomas Dolk, Michael Fliesser, Martin H. Fischer, Guido Nottbuschand, and Julia Festman. 2016. "Is Handwriting Performance Affected by the Writing Surface? Comparing Preschoolers', Second Graders', and Adults' Writing Performance on a Tablet vs. Paper". In *Journal on Frontiers in Psychology*, Vol. 7.

[27] Alex Graves, Santiago Fernández, Faustino John Gomez, and Jürgen Schmidhuber. 2006. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". In *Intl. Conf. on Machine Learning (ICML)*. 369–376.

[28] Alex Graves, Marcus Liwicki, Santiago Fernàndez, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. "A Novel Connectionist System for Unconstrained Handwriting Recognition". In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 31(5). 855–868.

[29] Patrick Grother. 1995. "NIST Special Database 19 Handprinted Forms and Characters Database". In *National Institue of Standards and Technology*.

[30] Isabelle Guyon, Lambert Schomaker, Rójean Plamondon, and Stanley A. Janet. 1994. "UNIPEN Project of On-line Data Exchange and Recognizer Benchmarks". In *Intl. Conf. on Pattern Recognition (ICPR)*, Vol. 2.

[31] Sepp Hochreiter and Jürgen Schmidhuber. 1997. "Long Short-Term Memory". In *Neural Computation (NC)*, Vol. 9(8).
[32] Heloise Hse and A. Richard Newton. 2004. "Sketched Symbol Recognition using Zernike Moments". In *Intl. Conf. on Pattern Recognition (ICPR)*, Vol. 1. Cambridge, UK, 367–370.
[33] Jonathan J. Hull. 1994. "Database for Handwritten Text Recognition Research". In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 16(5). 550–554.
[34] Eric Anquetil (UMR IRISA). 2020. "IMISketchSDB". https://www-intuidoc.irisa.fr/base-de-donnees-imisketchsdb/
[35] Neelasagar K. and K. Suresh. 2015. "Real Time 3D-Handwritten Character and Gesture Recognition for Smartphone". In *Intl. Journal of Computer Applications (IJCA)*, Vol. 123(13). 1–8.
[36] Shaikh Jahidabegum K. 2015. "Character Recognition System for Text Entry Using Inertial Pen". In *Intl. Journal of Science, Engineering and Technology Research (IJSETR)*, Vol. 4.
[37] Birendra Keshari and Stephen M. Watt. 2008. "Online Mathematical Symbol Recognition using SVMs with Features from Functional Approximation".
[38] Minwoo Kim, Jaechan Cho, Seongjoo Lee, and Yunho Jung. 2019. "IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces". In *Journal of Sensors*, Vol. 19(18). 3827.
[39] Christopher Koellner, Marc Kurz, and Erik Sonnleitner. 2019. "What Did You Mean? An Evaluation of Online Character Recognition Approaches". In *Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Barcelona, Spain, 1–6.
[40] Joseph J. LaViola and Robert C. Zeleznik. 2007. "A Practical Approach for Writer-Dependent Symbol Recognition Using a Writer-Independent Symbol Recognizer". In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 29(11).
[41] Bo Li, Yijuan Lu, Afzal A. Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose M. Saavedra, and Shoki Tashiro. 2013. "SHREC'13 Track: Large Scale Sketch-based 3D Shape Retrieval". In *Eurographics Workshop on 3D Object Retrieval (3DOR)*. Girona, Spain.
[42] Bo Li, Yijuan Lu, Chen-Feng Li, Afzal A. Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Hongbo Fu, Takahiko Furuya, H. Johan, J. Liu, Ryutarou Ohbuchi, A. Tatsuma, and Changqing Zou. 2014. "SHREC'14 Track: Extended Large Scale Sketch-Based 3D Shape Retrieval". In *Eurographics Workshop on 3D Object Retrieval (3DOR)*. Strasbourg, France.
[43] PeiYu Li, Ney Renau-Ferrer, Eric Anquetil, and Eric Jamet. 2012. "Semi-Customizable Gestural Commands Approach and its Evaluation". In *Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR)*. Bari, Italy, 473–478.
[44] Yang Li. 2010. "Protractor: A Fast and Accurate Gesture Recognizer". In *Intl. Conf. on Human Factors in Computing Systems (CHI)*. 2169–2172.
[45] Marcus Liwicki and H. Bunke. 2005. "IAM-OnDB - An On-line English Sentence Database Acquired from Handwritten Text on a Whiteboard". In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*. Seoul, South Korea.
[46] Marcus Liwicki, Alex Graves, Horst Bunke, and Jürgen Schmidhuber. 2007. "A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks". In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*. 367–371.
[47] David Llorens, Federico Prat, Andrés Marzal, Juan Miguel Vilar, Maria Jóse Castro-Bleda, Juan Carlos Amengual, Sergio Barrachina Mir, Antonio Castellanos, Salvador Espa na Boquera, Jon Ander Gómez, Jorge Gorbey-Moya, Albert Gordo, Vicente Palazón-González, Guillermo Peris Ripollés, Rafael Ramos-Garijo, and Francisco Zamora-Martinez. 2008. "The UJIpenchars Database: a Pen-Based Database of Isolated Handwritten Characters". In *Intl. Conf. on Language Resources and Evaluation (LREC)*. Marrakech, Morocco.
[48] Leena Mahajan and G. A. Kulkarni. 2014. "Digital Pen for Handwritten Digit and Gesture Recognition Using Trajectory Recognition Algorithm Based on Triaxial Accelerometer - A Review". In *Intl. Journal of Innovative Research in Science, Engineering and Technology*, Vol. 3(4). 356–363.
[49] U.-V. Marti and H. Bunke. 2002. "The IAM-database: an English Sentence Database for Offline Handwriting Recognition". In *Intl. Journal on Document Analysis and Recognition (ICDAR)*. 39–46.
[50] C. S. Myers and L. R. Rabiner. 1981. "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition". In *The Bell System Technical Journal*, Vol. 60(7). 1389–1409.
[51] Ralph Niels, Don Willems, and Louis Vuurpij. 2009. "The NicIcon Database of Handwritten Icons for Crisis Management".
[52] Michael Oltmans. 2007. "Envisioning Sketch Recognition: A Local Feature Based Approach to Recognizing Informal Sketches". In *PhD, Massachusetts Institute of Technology (MIT)*.
[53] Tom Y. Ouyang and Randall Davis. 2009. "A Visual Approach to Sketched Symbol Recognition". In *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*. 1463—-1468.
[54] Tse-Yu Pan, Chih-Hsuan Kuo, Hou-Tim Liu, and Min-Chun Hu. 2019. "Handwriting Trajectory Reconstruction Using Low-Cost IMU". In *Trans. on Emerging Topics in Computational Intelligence (TETCI)*, Vol. 3(3). 261–270.
[55] Vietminh Paz-Villagrán, Jérémy Danna, and jean-luc Velay. 2013. "Lifts and Stops in Proficient and Dysgraphic Handwriting". In *Journal Human Movement Science*, Vol. 33(1).
[56] Réjean Plamondon and Sargur N. Srihari. 2000. "On-line and Off-line Handwriting Recognition: A Comprehensive Survey". In *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 22(1). 63–84.
[57] Anisha Priya, Surbhi Mishra, Saloni Raj, Sudarshan Mandal, and Sujoy Datta. 2016. "Online and Offline Character Recognition: A Survey". In *Intl. Conf. on Communication and Signal Processing (ICCSP)*. Melmaruvathur, India, 967–970.

[58] Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014. "Match-up & Conquer: A Two-step Technique for Recognizing Unconstrained Bimanual and Multi-finger Touch Input". In *Intl. Working Conf. on Advanced Visual Interfaces (AVI)*. 201–208.

[59] Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014. "Understanding Users' Perceived Difficulty of Multi-Touch Gesture Articulation". In *Intl. Conf. on Multimodal Interaction (ICMI)*. 232–239.

[60] Ney Renau-Ferrer, Peiyu Li, Adrien Delaye, and Eric Anquetil. 2012. "The ILGDB Database of Realistic Pen-based Gestural Commands". In *Intl. Conf. on Pattern Recognition (ICPR)*. Tsukuba, Japan, 3741–3744.

[61] Valérie Renaudin, Muhammad Haris Afzal, and Gérard Lachapelle. 2001. "Complete Triaxis Magnetometer Calibration in the Magnetic Domain". In *Journal of Sensors*.

[62] J. M. Romeu, B. Lamiroy, G. Sanchez, and J. Llados. 2006. "Automatic Adjacency Grammar Generation from User Drawn Sketches". In *Intl. Conf. on Pattern Recognition (ICPR)*. Hong Kong, China, 1026–1029.

[63] Manju Sahu and Alpha Kujur. 2017. "Differentiation and Comparison of Left Handed and Right Handed Writers on the Basis of Strokes and Slope of Letter". In *Intl. Journal of Current Research and Review (IJCRR)*, Vol. 9(11). 6–9.

[64] Mike Schuster and K. K. Paliwal. 1997. "Bidirectional Recurrent Neural Networks". In *Trans. on Signal Processing*, Vol. 45(11). 2673–2681.

[65] Timothy J. Smoker, Carrie E. Murphy, and Alison K. Rockwell. 2009. "Comparing Memory for Handwriting versus Typing". In *Human Factors and Ergonomics Society (HFES)*.

[66] STABILO DigiVision. 2020. "The STABILO Digipen: A Sensor-equipped Ballpoint Pen with Wireless Connectivity. The UbiComp 2020 Challenge". https://stabilodigital.com/stabilo-digivision/

[67] C. C. Tappert. 1982. "Cursive Script Recognition by Elastic Matching". In *Journal of Research and Development (JRnD)*, Vol. 26(6). 765–771.

[68] Caglar Tirkaz, Berrin Yanikoglu, and T. Metin Sezgin. 2012. "Sketched Symbol Recognition with Auto-completion". In *Journal of Pattern Recognition*. 3926–3937.

[69] Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. "Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes". In *Intl. Conf. on Multimodal Interaction (ICMI)*. 273–280.

[70] Christian Viard-Gaudin, Pierre Michel Lallican, Stefan Knerr, and Philippe Binter. 1999. "The IRESTE On/Off (IRONOFF) Dual Handwriting Database". In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*. Bangalore, India, 455–458.

[71] Alessandro Vinciarelli and Michael Peter Perrone. 2003. "Combining Online and Offline Handwriting Recognition". In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*. Edinburgh, UK, 844–848.

[72] Jeen-Shing Wang and Fang-Chen Chuang. 2012. "An Accelerometer-Based Digital Pen with a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition". In *Trans. on Industrial Electronics (IES)*, Vol. 59(7). 2998–3007.

[73] Jeen-Shing Wang, Yu-Liang Hsu, and Cheng-Ling Chu. 2013. "Online Handwriting Recognition Using an Accelerometer-Based Pen Device". In *Intl. Conf. on Computational Science and Engineering (CSE)*. Sydney, Australia, 229–232.

[74] Kai Wang and Serge Belongie. [n.d.]. "Word Spotting in the Wild". In *Europ. Conf. on Computer Vision (ECCV)*. Heraklion, Crete.

[75] R. Allen Wilkinson, Jon Geist, Stanley Janet, Patrick J. Grother, Christopher J. C. Burges, Robert Creecy, Bob Hammond, Jonathan J. Hull, Norman J. Larsen, Thomas P. Vogl, and Charles L. Wilson. [n.d.]. "The First Census Optical Character Recognition System Conference".

[76] Don Willems, Ralph Niels, Marcel van Gerven, and Louis Vuurpijl. 2009. "Iconic and Multi-stroke Gesture Recognition". In *Journal of Pattern Recognition*, Vol. 42(12). 3303–3312.

[77] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. "Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes". In *ACM Symp. on User Interface Software and Technology (UIST)*. 159–168.

[78] Adeel Yousaf, Muhammad Jaleed Khan, M. Imran, and Khurram Khurshid. 2017. "Benchmark Dataset for Offline Handwritten Character Recognition". In *Intl. Conf. on Emerging Technologies (ICET)*. 1–5.

[79] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. "Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors". In *Intl. Conf. on Mobile Computing, Applications and Services (MobiCASE)*. Austin, TX, 197–205.