IAPR-MEDPRAI

ARDIS: a Swedish historical handwritten digit dataset

Huseyin Kusetogullari[1] · Amir Yavariabdi[2] · Abbas Cheddad[1] · Håkan Grahn[1] · Johan Hall[3]

Abstract
This paper introduces a new image-based handwritten historical digit dataset named Arkiv Digital Sweden (ARDIS). The images in ARDIS dataset are extracted from 15,000 Swedish church records which were written by different priests with various handwriting styles in the nineteenth and twentieth centuries. The constructed dataset consists of three single-digit datasets and one-digit string dataset. The digit string dataset includes 10,000 samples in red–green–blue color space, whereas the other datasets contain 7600 single-digit images in different color spaces. An extensive analysis of machine learning methods on several digit datasets is carried out. Additionally, correlation between ARDIS and existing digit datasets Modified National Institute of Standards and Technology (MNIST) and US Postal Service (USPS) is investigated. Experimental results show that machine learning algorithms, including deep learning methods, provide low recognition accuracy as they face difficulties when trained on existing datasets and tested on ARDIS dataset. Accordingly, convolutional neural network trained on MNIST and USPS and tested on ARDIS provide the highest accuracies 58.80% and 35.44%, respectively. Consequently, the results reveal that machine learning methods trained on existing datasets can have difficulties to recognize digits effectively on our dataset which proves that ARDIS dataset has unique characteristics. This dataset is publicly available for the research community to further advance handwritten digit recognition algorithms.

Keywords Handwritten digit recognition · ARDIS dataset · Machine learning methods · Benchmark

✉ Abbas Cheddad
abbas.cheddad@bth.se

Huseyin Kusetogullari
huseyin.kusetogullari@bth.se;
huseyinkusetogullari@gmail.com

Amir Yavariabdi
amir.yavariabdi@gmail.com

Håkan Grahn
hakan.grahn@bth.se

Johan Hall
johan.hall@arkivdigital.se

[1] Department of Computer Science and Engineering, Blekinge Institute of Technology, 37141 Karlskrona, Sweden

[2] Department of Mechatronics Engineering, KTO Karatay University, Konya, Turkey

[3] Arkiv Digital, Växjö, Sweden

1 Introduction

Recently, digitization of handwritten documents has become significantly important to protect and store data more efficiently. The growth of digitized handwritten documents highlights new types of challenges and problems which lead to development of many automated and computerized analysis systems. Generally, the developed frameworks have been used to resolve various problems such as character recognition, identity prediction, digit segmentation and recognition, document binarization, automatic analysis of birth, marriage and death records, and many others [1–5]. Among them, this paper focuses on the handwritten digit recognition problem.

In the last three decades, there has been vast escalation in the development of handwritten digit recognition techniques to convert digits into machine readable form. This escalation stems from the fact that there are a wide range of applications including online handwriting recognition on smart phones, handwritten postal codes recognition to sort postal mails, processing bank check amounts, and storing documents and forms in digital formats based on handwritten numeric entries

(e.g., year or document numbers) for easier retrieval and information collection [6–8]. In this context, the existing methods are either based on scanned data or on trajectory data which are recorded during the writing process. Therefore, based on the types of input data, the handwritten digit recognition methods can be divided into two categories: online and off-line recognition. This paper focuses on off-line recognition approach.

Off-line recognition is performed after the digit has been written and it processes images which are captured using a scanner or a digital camera. It is a traditional but still challenging problem for many languages like English, Chinese, Japanese, Farsi, Arabic, and Indian. This difficulty stems from the fact that there are large variations in writing styles, stroke thicknesses, shapes, and orientations as well as existence of different kinds of noise which can cause discontinuity in numerical characters. To tackle these problems, in the last few decades, numerous frameworks based on machine learning methods have been proposed and developed mostly for modern handwritten documents. Moreover, for standard evaluation of handwritten digit recognition methods, a number of handwritten benchmark datasets based on modern handwriting have been created. This paper focuses on the off-line historical handwritten digit recognition as recognizing handwritten digits in historical document images is still an unsolved problem due to the high variation within both the document background and foreground.

Digits in handwritten historical documents are far more difficult to classify as several factors such as paper texture, aging, handwriting style, and the kind of ink and dip pen as well as digit thickness, orientation, and appearance may influence the performance of the classifier algorithms. In order to improve performance and reliability of digit classifiers on historical documents, a new digit dataset must be created since available handwritten digit datasets have some limitations. These limitations in current datasets are: (1) the digits are collected from recently written (modern) and non-degraded documents; (2) the digits are written in modern handwriting styles; and (3) the digits are mostly written by ballpoint and rollerball pens. Considering the aforementioned limitations, we construct a new handwritten digit dataset named ARDIS containing four different datasets (publicly available from: https://ardisdataset.github.io/ARDIS/).

In the ARDIS dataset, the digits are collected from Swedish historical documents that span the years from 1895 to 1970, which were written in printing, copperplate, cursive, and Gothic styles by different priests using various types of ink and dip pen. Figure 1 illustrates an example of the historical documents where the digits are collected. In ARDIS, the first dataset contains 10,000 digit string images with 75 classes based on date attribute. The other three datasets contain 7600 single-digit images in each, where the image color space, as well as background and foreground formats, is different from each other. To provide the research community with a rigorous and comprehensive scientific benchmark, these four different datasets are publicly available. Moreover, we give access to author-approved implementation of used machine learning algorithms for training and testing and ranking of existing algorithms. It is important to stress that, in this paper, the main focus is not on designing a new complex machine learning classifier framework, but rather understanding and analyzing of existing architectures on historical documents using available datasets and ARDIS. The experimental results show the poor performance of machine learning methods trained on publicly available digit datasets and tested on the ARDIS, which emphasizes the necessity and added value of constructing a new digit dataset for historical handwritten digit recognition.

2 Related works

Instead of undertaking a detailed discussion of the existing literature on handwritten digit recognition, we briefly summarize the frequently used machine learning approaches and datasets. An extensive survey of handwritten digit recognition methods can be found in [2, 5, 9, 10].

2.1 Handwritten digit recognition methods

One of the simplest machine learning approaches which have been used for handwritten digits recognition is k-nearest neighbor (kNN) classifier. In this manner, Babu et al. [11] propose a handwritten digit recognition method based on kNN classifier. In this paper, firstly, structural features such as number of holes, water reservoirs, maximum profile distances, and fill hole are extracted from the

Fig. 1 Partial view of a Swedish historical handwritten document recorded in 1899

images and used for the recognition of numerals. After that, a Euclidean minimum distance criterion is used to compute distance of each query instance to all training samples. Finally, kNN classifier is employed to classify the digits. The authors reported a 96.94% recognition rate on MNIST dataset. Many other kNN-based methods have also been proposed [12–14]. Even though kNN algorithm is simple to use, it has various disadvantages such as: (1) it has a significant computational cost; (2) it does not take the structure of the data space into account; and (3) it provides low recognition rate for multi-dimensional sets [15].

Another classifier approach that has been used in this context is random forest technique. For instance, Bernard et al. [16] test random forest classifier on MNIST dataset. In this work, the grayscale multi-resolution pyramid method [17] is used as a feature extraction technique. Using the verified data for selecting parameters of random forest classifier, they obtain a success accuracy of 93.27%. Generally, random forest classifier results in poor classification performance as it is constructed to mitigate the overall error rate. Moreover, to deal with the problem of handwritten digit recognition, several papers in the literature have also suggested adopting a probabilistic approach, such as naive Bayes classifiers [18], hidden Markov model [19], and Bayesian networks [20].

For decades, support vector machine (SVM) has been acknowledged as a powerful classification tool for data learning due to its high classification accuracy and good generalization capability. Maji et al. [21] propose a handwritten digit recognition method based on SVM classifier. In this method, pyramid histogram of oriented gradient (PHOG) is used to extract the features from the handwritten digit images. After that, the extracted features are classified using one-versus-all SVM classifier with linear, intersection, degree five polynomial, and radial basis function (RBF) kernels, respectively. In their experiments, for MNIST dataset, the best error rate is 0.79% achieved by the polynomial kernel SVM, and for USPS dataset, the success rate is 97.3% achieved by RBF kernel SVM. Moreover, many other SVM-based algorithms have been proposed and developed for handwritten digit recognition problem [22–28].

Artificial neural network (ANN) is another type of supervised machine learning method, which has also been widely used in handwritten digit recognition [29–35]. Generally, ANN differs from SVM in two important aspects: (1) To classify nonlinear data, SVM uses a kernel function to make the data linearly separable, but ANN employs multilayer connection and various activation functions to deal with nonlinear problems and (2) SVM minimizes the empirical and the structural risks learnt from the training samples; however, ANN minimizes only the empirical risk [36]. Zhan et al. [35] propose an ANN-based algorithm for handwritten digit string recognition. This method consists of two steps. Firstly, they use residual network to extract features from digit images. Secondly, a recurrent neural network is employed to model the data and for prediction. Note that in order to train recurrent neural networks, connectionist temporal classification is used as end to end training method. They obtain the recognition rates of 89.75% and 91.14% for ORAND-CAR-A and ORAND-CAR-B datasets [37], respectively. These lower accuracy rates show that these two datasets are more challenging than MNIST. Ciressan et al. [38] develop a digit recognition method using deep big multilayer perceptrons. To design the deep big ANN model, nine hidden layers involving 2500 neurons per layer are used to avoid overfitting. MNIST dataset is used as benchmark to evaluate the performance of the classifier which depict that the proposed ANN architecture provides high recognition rate. Holmstrom [39] uses ANN classifier based on PCA features. In this paper, the results show that ANN performs poorly on the PCA features.

Recently, many research works have shown the improvement in recognition performance using deep learning approaches. For instance, Ciresan et al. [40] propose a deep neural network model using convolutional neural network (CNN). The architecture of the method is as follows: (1) two convolutional layers with 20 and 40 filters and kernel size of 4×4 and 2×2; (2) each convolution layer is followed by one maxpooling layer over non-overlapping regions of size 3×3; (3) two fully connected layers containing 300 and 150 neurons; and (4) one output layer with 10 neurons. The classifier is applied to MNIST dataset and achieved 0.23% error rate. Wang et al. [41] propose a deep learning method to solve the very low-resolution digit recognition problem. The method is designed based on the CNN and it includes three convolutional layers and two fully connected layers. This method is applied to SVHN dataset and obtained the lowest error rates comparing to other machine learning methods. Sermanet et al. [42] develop a deep learning method for house numbers digit classification. Chellapilla et al. [43] design a CNN model with two convolutional layers and two fully connected layers for handwritten digit recognition problem. The model uses a graphical processing unit (GPU) implementation of convolutional neural networks for both training and testing of handwritten digits. In this paper, they showed the advantages of using GPUs over central processing units (CPUs). In [44], different models of CNNs have been discussed to achieve the highest accuracy rates for the handwritten digit recognition on NIST dataset. Many other deep learning methods have been designed and developed to obtain high recognition rate for different handwritten digit datasets [45–48].

2.2 Existing handwritten digit datasets

Different standard handwritten digit datasets have been created in which the handwritten digits are preprocessed

Table 1 Handwritten digit datasets in different languages

Dataset name	Language	Training images	Test images	Total images	Publicly available	Color type	Digit strings
MNIST [51]	Latin	60,000	10,000	70,000	Yes	Grayscale	No
CEDAR [10]	Latin	18,468	2711	21,179	No	–	No
Semeion [56]	Latin	1200	393	1593	Yes	Grayscale, binary	No
USPS [55]	Latin	7291	2007	9298	Yes	Grayscale	No
CASIA-HWDB [61]	Chinese	800	220	1020	–	Color	No
Chars74K [66, 67]	Latin	65,000	9000	74,000	Yes	Natural images	No
SRU [60]	Persian	6450	2150	8600	Yes	Grayscale	No
IRONOFF [59]	Latin	3700	386	4086	No	–	No
ADBase [62]	Arabic	60,000	10,000	70,000	Yes	Binary	No
LAMIS-MSHD [65]	French and Arabic	–	–	21,000	No	RGB	Yes
Synthetic [68]	Latin	7500	2500	10,000	Yes	Grayscale	No
CENPARMI [53]	Latin	4000	2000	6000	No	–	No
CENPARMI [53]	Middle East	18,390	10,035	28,425	No	–	No
ARDIS, Dataset I	Latin	–	–	10,000	Yes	RGB	Yes
ARDIS, Datasets II–IV	Latin	6600	1000	7600	Yes	Grayscale, RGB	No

manually or automatically [49, 50]. In the preprocessing phase, three different techniques are normally deployed, namely denoising, segmentation, and normalization. Consequently, the constructed dataset can be used for training and testing machine learning models. Without aiming to be exhaustive, the most widely used datasets (see Table 1) are listed and described below:

MNIST dataset This is one of the most well-known and most used standard datasets in digit recognition systems and it is publicly available [51]. MNIST dataset is created from the NIST dataset [51, 52]. It consists of 70,000 handwritten digit images in total, of which 60,000 are used for training and the rest are used for testing. Since there are 10 different digit classes, for each digit class, there are approximately 6000 different samples for training and 1000 for testing. In MNIST dataset, the digits are centralized and the images are size of 28×28 in grayscale. After that, each image is stored as a vector with 784 elements (28×28).

CENPARMI dataset CENPARMI [53] is another handwritten digit dataset consisting of 6000 sample images of which 4000 (400 samples per digit class) are used for training and 2000 are used for testing. The handwritten digit images of CENPARMI are obtained from live mail images of USPS, scanned at 166 dpi [53]. However, this dataset is not publicly available [54].

USPS dataset USPS [21, 55] includes 7291 training images and 2007 testing images in grayscale for the digits 0

to 9. The images are with the size of 16×16, and people have difficulty in recognizing the complex USPS digits with reported human error rate of 2.5% [21]. This dataset is publicly available.

Semeion dataset Semeion [56, 57] contains 1593 handwritten digits written by 80 participants. Each participant writes down all the digits from 0 to 9 on different papers, twice. These digit images are with the size of 16×16 in grayscale. The main problem of this dataset is that it has very less digit images for machine learning algorithms.

CEDAR dataset CEDAR [10] comprises of 21,179 images from SUNY5 at Buffalo (USA) and they are extracted from the images scanned at 300 dpi. The overall dataset is partitioned into two parts with 18,468 images for training and 2711 images for testing. This dataset is not publicly available [58].

IRONOFF online/off-line handwriting dataset In [59], IRONOFF dataset is introduced with isolated French characters, digits, and cursive words. This dataset is online and off-line collected from digitized documents written by French writers. Besides this, it contains 4086 isolated handwritten digits. For the off-line domain, the images are scanned with a resolution of 300 dpi with 8 bits per pixel. This dataset is not publicly available.

Besides the Latin handwritten digit datasets explained and described above, other handwritten digit datasets have been created in other languages. Some of them are described below:

SRU dataset SRU [60] is made up of 8600 handwritten digit images for training and testing processes in Persian language. This digit dataset is extracted from digitized documents written by 860 undergraduate students from universities in Tehran. All digit images are with the size of 40×40 pixels and obtained from the images scanned at 300 dpi resolution in grayscale. The training and test sets contain 6450 and 2150 samples, respectively.

CASIA-HWDB dataset CASIA-HWDB [61] dataset contains three different datasets. This dataset is created by 1020 Chinese participants. The isolated Chinese characters and alphanumeric samples are extracted from handwritten pages at scanned 300 dpi resolution in red–green–blue (RGB) color space. The alphanumeric and character images are segmented and labeled using annotation tools. In this dataset, the background of images is white and the digits are represented in grayscale.

ADBase dataset ADBase [62, 63] contains 70,000 Arabic handwritten binary digits written by 700 participants. Each participant writes 10 different digits on the given papers 10 times. The papers are scanned with 300 dpi resolution of which the digits are automatically extracted, categorized, and bounded. The training and test sets include 60,000 (6000 images per class) and 10,000 (1000 images per class) binary digit images, respectively. This dataset is publicly available [64].

LAMIS-MSHD dataset The LAMIS-MSHD (multi-script handwritten dataset) [65] is newly created and it comprises 600 Arabic and 600 French text samples, 1300 signatures and 21,000 digits. The dataset is extracted from 1300 forms written by 100 Algerian people with different age groups and educational backgrounds. The forms are scanned with a resolution of 300 dpi with 24 bits per pixel. This dataset is not publicly available [65].

Chars74K dataset Campos et al. [66] present a dataset with 64 classes. It contains 7705 handwritten characters, 3410 hand drawn characters, and 62,992 synthesised characters obtained from natural images, tablet, and computer, respectively. As a result in this dataset, there are more than 74,000 characters which are written in Latin, Hindu, and Arabic languages. The dataset is publicly available for researchers [66, 67].

Synthetic digit dataset Generally, the digits in the datasets explained and described above are generated by human efforts. Besides these datasets, there are also datasets that are generated artificially called synthetic. One of the synthetic datasets is publicly available in MATLAB toolbox [68]. This dataset includes 10,000 images of which 7500 images are training samples and 2500 images are test samples. Another synthetic dataset is presented by Hochuli et al. [44] which consists of numerical combinations of 2, 3, and 4 digits. The digit strings are built by concatenating isolated digits of NIST dataset by using the machine learning algorithm described by Ribas et al. [69].

2.3 Limitations of existing digit datasets

Section 2.2 comprehensively studies the available handwritten digit datasets which can be leveraged by the researchers in optical character recognition community. The study reveals that there are five main issues with the existing datasets which can be highlighted as follows: (1) lack of sharing datasets and availability; (2) lack of datasets that are constructed and labeled in same format; (3) lack of availability of digit datasets constructed from historical documents written in old handwriting styles with various types of dip pens; (4) lack of availability of handwritten digit string datasets (i.e., dates with transcriptions); and (5) lack of availability of datasets without background cleaning and size normalization. These issues simply limit the application of machine learning methods for handwritten digit recognition, especially in historical documents analysis where the variability in styles becomes more prominent. We believe these issues are the key elements to justify the extension of the existing handwritten digit datasets. Moreover, when a dataset is exposed to many different inter-writer and intra-writer variations, the recognition performance improves and becomes one step closer to human performance. Additionally, too few available digit datasets makes the digit recognition problem more challenging to evaluate the robustness of retrieval methods on large-scale galleries. Therefore, to support the development of research in both handwritten digit and handwritten numerical pattern recognition, it is necessary to construct new digit datasets that would address the shortcomings of the existing ones. In this manner, we construct four different datasets obtained from Swedish historical documents (Fig. 2).

Fig. 2 Illustration of handwritten digits collection from the top part of a Swedish handwritten document written in 1896 (color figure online)

3 ARDIS dataset

Arkiv Digital is the largest online private provider of Swedish church, census, military, and court records. The Arkiv Digital dataset contains approximately 80 million high-quality historical document images. The images in this unique dataset are captured by different types of Canon digital cameras in RGB format with the resolutions of 5184 × 3456 pixels in which the oldest recorded documents date back to the year 1186 and the newest ones are from 2016. The collected image dataset is undoubtedly a precious resource for genealogy, history, and computer science researchers.

In order to construct the ARDIS digit dataset, only church records are considered since they were written on a standardized template (e.g., tabulated form). These documents were written by different priests in Swedish churches from 1895 to 1970. As the documents were written by different writers and with different dip pens, the alphabets are scripted in various sizes, directions, widths, arrangements, and measurements. This might provide endless differences. The digits are extracted from about 15,000 church document images. Figure 3 demonstrates the distributions of the number of documents in each year which also indicates that there are 75 classes. Moreover, these documents are useful to keep track of information about the residents who were born, married, and/or dead in Sweden. Besides the information about residents, the documents also contain other types of information such as category of the book, year in which the document was written, and many other attributes. In the rest of this section, the procedure of collecting digits and characteristics of digit images are discussed.

3.1 Data collection

In this paper, we introduce four different handwritten digit datasets that are constructed from the Swedish historical documents. The datasets (publicly available from: https://ardisdataset.github.io/ARDIS/) are as follows:
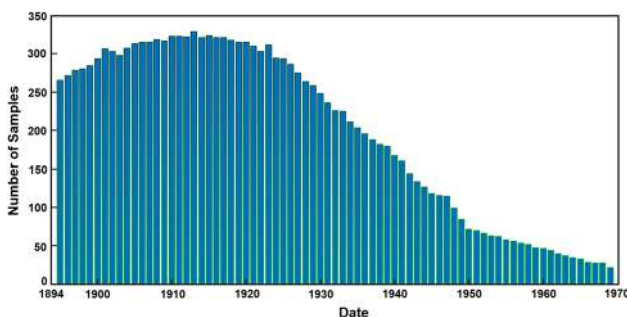

Fig. 3 Distribution of documents in each year in ARDIS dataset. The horizontal axis and the vertical axis indicate the years and sample numbers, respectively

Dataset I An automatic method is used to localize and detect year information from 10,000 out of the 15,000 documents which are subsequently manually labeled. Note that the years in the rest of the images are half-handwritten and half-typed, so that they are discarded. The handwritten year is cropped with the size of 175 × 95 pixels from each document image and stored in RGB color space to form this dataset as shown in the first row of Fig. 4. Each image in this dataset contains 4-digit year as illustrated on the top left and top right of the document image in Fig. 1. The label vector is one-dimensional array of the corresponding years on each document. This dataset can be used in various applications such as digit segmentation from digit string samples, image binarization and digit string recognition on degraded images (e.g., bleed-through, faint handwritten digits, and weak text stroke) [44].

Dataset II This dataset is collected from some of the 15,000 document images and includes only isolated digits from 0 to 9 in Latin alphabet. Each digit is manually segmented from the document images as shown in Fig. 2. To generate this dataset, only isolated digits are considered (blue boxes in Fig. 2), while connected and overlapping digits are discarded (red boxes in Fig. 2). To the best of our knowledge, this dataset is the first one to provide images in RGB color space and they are delivered in original size. Contrary to other existing digit datasets, the digit images are not size-normalized, but they are given in the original size as in real-world cases, where there is variation in size and writing style. Note that digit images in this dataset may contain extra part(s) from other digits and other artifacts (e.g., line dashes and noise) as shown in the second row of Fig. 4. This dataset of segmented digits consists of 10 classes (0–9), with 760 samples per class. This dataset is created to generate more reliable single-digit recognition
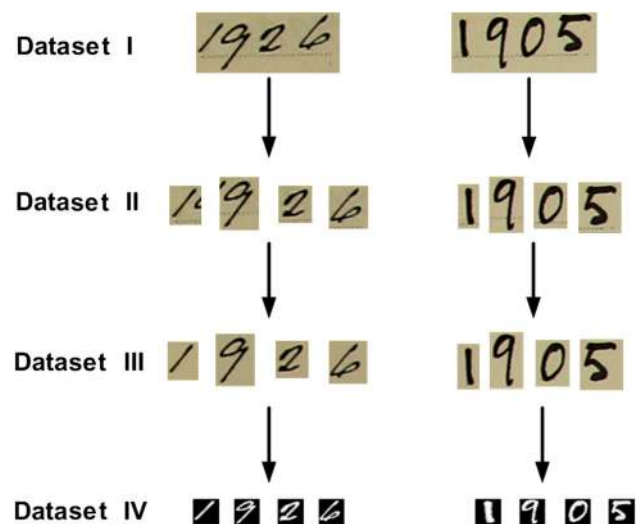

Fig. 4 Handwritten digit images from different datasets in ARDIS

and segmentation systems on images with complex backgrounds.

Dataset III The digits in this dataset are derived from the dataset II, where the images are denoised. The images in the previous dataset, as shown in second row of Fig. 4, contain artifacts such as noise, dash lines, and partial view of the other digits. In order to create dataset III, the artifacts on each image are manually cleaned as shown in the third row of Fig. 4. When setting up this dataset, a uniform distribution of the occurrences of each digit was ensured. In other words, this dataset consists of 7600 denoised handwritten digit images in RGB color space.

Dataset IV This dataset is derived from the dataset III, where the images are in grayscale and size-normalized as shown on the last row of Fig. 4. More specifically, this dataset contains images with the size of 28×28 where the background is black and digits are in grayscale. This dataset mimics the image dimensions in the MNIST dataset. Such standardization in data format allows researchers to easily combine it with MNIST to include more variations of handwriting styles. This may improve the performance of digit recognition methods. This dataset contains 7600 handwritten digit images of which 6600 samples are used for training and 1000 for testing.

3.2 Data characteristics

ARDIS dataset is featured in several aspects. First, this digit dataset is collected from Swedish church records written in the nineteenth and twentieth centuries. Therefore, the ARDIS dataset covers a wide range of the nineteenth- and twentieth-century handwritten styles such as Gothic, cursive, copperplate, and printing. Second, the digits are written by different priests using various types of dip pen, nib, and ink which result in different methods of sketching and yielding different appearances. For instance, only nib angle can control the thickness of strokes which generate uncountable variations in writing digits. Third, applying various pressures on a nip can cause of flowing different amount of ink which generates unlimited variations in digit writing. Other aspects such as size of digits, age of documents, and distortions also influence the characteristic of the digits. For instance, in the documents the same digits were written with many different sizes; thus, the shape of the digits can be diverse. The poor quality of the used papers and inks in the nineteenth and twentieth centuries results in rapid deterioration of documents and handwritings [70]. This simply generates many distortions in the appearance of digits and their backgrounds. All those characteristics in documents lead to a generation of unique digit dataset where the digits appear with many variations.

4 Benchmark evaluation

4.1 Architecture of compared methods

For quantitative evaluations, different classifier and learning methods such as kNN, random forest, one-versus-all SVM classifier with RBF kernel, recurrent neural network (RNN), and convolutional neural networks (CNNs) are used. The first compared method is kNN-based handwritten digit classifier. In the kNN, the distance between feature vector values of the test image and feature vector values of every training image is estimated using the Euclidian distance and digits are classified by the majority class of its k-nearest neighbors in the training dataset. In this algorithm, the raw pixel values are used as feature values. The appropriate choice of k has significant impact on the diagnostic performance of the kNN algorithm. In our experiments, the optimal value of k is empirically chosen as 1 for classification of handwritten digits.

The second compared method is random forest classifier. In the random forest approach, the raw pixels in the images are first normalized to $[0, 1]$ and then used as feature values. The random forest classifier includes two parameters which are: (1) the number L of trees in the forest and (2) the number K of random features preselected in the splitting process. In our experiments, we use $L = 100$ and $K = 12$ as optimal parameters. The comprehensive evaluation of these parameters is discussed in [16].

The third handwritten digit recognition method is based on RBF kernel SVM. To evaluate the performance of SVM learning and classifier methods, the raw pixels of images and the histogram of oriented gradients (HOGs) are used as feature vectors. Therefore, these two feature types generate two different experimental setup called as SVM and HOG–SVM in the rest of the paper. The HOG feature descriptor has two parameters that need to be set: (1) the size of the cell in pixels and (2) the number of orientation bins. Here, we set them as 4×4 and 8, respectively. Moreover, RBF kernel SVM classifier has also two different parameters, which are non-overlapping blocks γ and the dimensions of the eigenvector space C. In our experiments, we use $\gamma = 0.001$ and $C = 1$.

The fourth handwritten digit classifier is based on RNN with a three-layer neural network. In RNN classifier, the pixel values of the normalized image are used as feature values. Here, the number of training examples used in one iteration (batch size) is 128 and samples in each batch pass forward and backward through the RNN (epoch) 10 times. In addition, ReLU is used as an activation function in the hidden layers and Softmax is applied to estimate probabilities of each output class.

The fifth compared method is CNN-based handwritten digit classifier. This classifier includes two convolutional layers, two fully connected layers, and one output layer. The first convolutional layer uses 32 filters with the kernel size of 5×5, whereas the second convolutional layer employs 64 filters with the same kernel size. The convolutional layers are equipped with max-pooling filters with the pool size of 2. Each fully connected layer includes 128 nodes. Moreover, ReLU is used as an activation function in the convolutional and fully connected layers. In addition, Softmax is used to calculate probabilities of each output class in the last layer of the fully connected neural network. Note that the highest probability belongs to the target class. Total number of training examples present in a single batch is 200, and epoch size is 10. Furthermore, all the aforementioned methods are performed with Python 3.2 on Intel Core $i7$ processor (2.40-GHz) and 4-GB RAM.

4.2 Experimental setup

Dataset Split In this paper, for evaluation purposes, three different datasets such as MNIST, USPS, and ARDIS are used. MNIST dataset includes 60,000 training samples and 10,000 test samples. Each sample is in grayscale with the size of 28×28. In USPS dataset, the training and test sets contain 7291 and 2007, respectively. The images are in grayscale with the resolution of 28×28. In ARDIS dataset, we randomly split the data into training (approximately 86.85%) and test (about 13.15%) sets, resulting in 6600 training and 1000 test digit images. To fairly compare different classifiers and learning algorithms, the *dataset IV* of ARDIS is used. In this dataset, the images are in grayscale with the size of 28×28. In all the used datasets, the digits' pixels are in grayscale and the background is black. For instance, ten different digits from ARDIS, MNIST, and USPS digits are shown in Fig. 5.

Evaluation metrics In this paper, two different evaluation techniques are used to evaluate the performance of the classifiers on the digit datasets. The first one is classification accuracy which is defined as the percentage of the correctly labeled samples. It can be formulated as follows:

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{TN}} \quad (1)$$

where TP is true positive which is the number of digit values correctly identified and TN is true negative that is the number of digit samples incorrectly identified by the classifier. The second evaluation method is confusion matrix. In the confusion matrix, the diagonal elements represent the number of points for which the predicted label is equal to the true label, while the off-diagonal elements are those that are wrongly labeled by the classifier. Note that the higher the diagonal values of the confusion matrix, the better is the result of the classifier. In other words, this indicates many correct predictions.

4.3 Comparison of digit recognition methods on various datasets

In the first experiment, a preliminary evaluation was conducted on MNIST dataset. More specifically, the compared machine learning methods are trained and tested on MNIST dataset. The results are tabulated in Table 2. According to the results, all the methods provide promising results for MNIST handwritten digit recognition with over 93% accuracy rate. This is due to the fact MNIST training and test samples have very similar characteristics. The highest accuracy rate is obtained using CNN which is 99.18%, whereas the lowest percentage belongs to RBF kernel SVM on the raw pixels, which is 93.78%. Moreover, we also use RBF kernel SVM on the HOG features. The results illustrate good performance of RBF kernel SVM on the HOG features with the error rate of 2.18%. Random
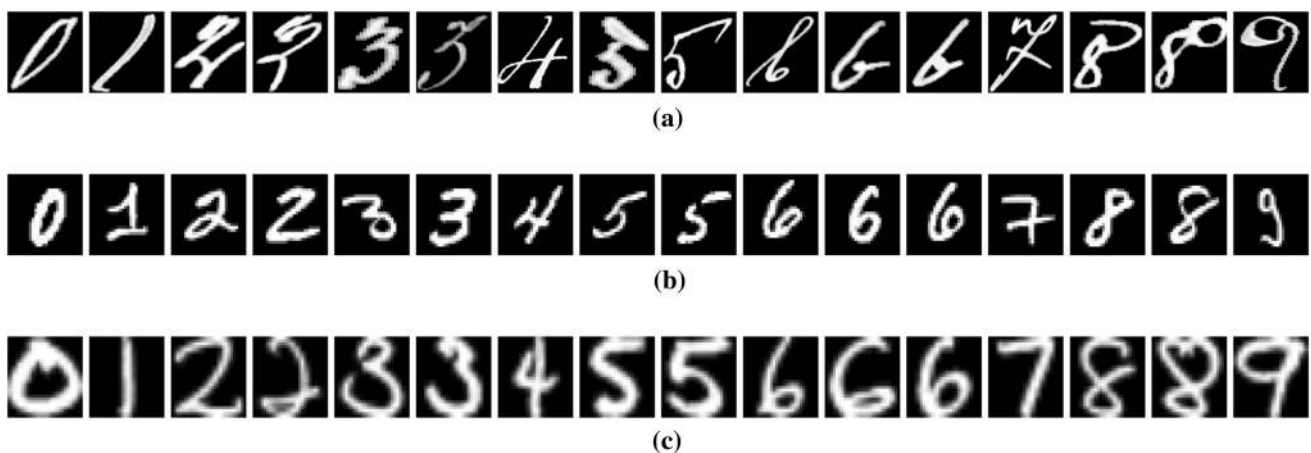

Fig. 5 Illustration of digit values from 0 to 9: **a** ARDIS, **b** MNIST, and **c** USPS

Table 2 Recognition accuracy of machine learning methods on MNIST dataset

Method	Recognition accuracy (%)
CNN	99.18
SVM	93.78
HOG–SVM	97.82
kNN	97.31
Random forest	94.82
RNN	96.95

forest and RNN provide the recognition accuracy of 94.82% and 96.95%, respectively. Furthermore, these results show that the machine learning models work well to achieve high-accuracy results for MNIST dataset, and hence, these models are used for the next experiments in the paper.

The second experiment focuses on evaluation of diversities and similarities of different digit dataset. To achieve this, two different cases are considered. The first case considers the evaluation of machine learning methods which are trained on MNIST dataset and tested on ARDIS. The second case studies the performance of the classification methods that are trained on USPS dataset and tested on ARDIS. The overall results are given in Table 3. The results show high recognition error rates on ARDIS which indicate that there are many diversities between the digits on the existing datasets (MNIST and USPS) and ARDIS. More specifically, these low recognition accuracy rates simply mean that the samples in ARDIS dataset are more challenging than MNIST and USPS, and hence, the models generated by them cannot classify the samples in ARDIS. In ARDIS digit classification, the main challenges are: (1) the digits are written in Gothic, printing, copperplate, and cursive handwriting styles using different types of dip pen; (2) the handwritten digits are not of the same size, thickness, and orientation; and (3) the pattern and appearance of the digits are varying widely as they are taken from the old

Table 3 Handwritten digit recognition accuracy using different machine learning methods for Case I: training set: MNIST, testing set: ARDIS and Case II: training set: USPS, testing set: ARDIS

Method	Case I: Accuracy (%)	Case II: Accuracy (%)
CNN	58.80	35.44
SVM	43.40	30.62
HOG–SVM	56.18	33.18
kNN	50.15	22.72
Random forest	20.12	17.15
RNN	45.74	28.96

handwritten documents and written by different priests. Due to these complexities, the models obtained using MNIST and USPS mostly fail to correctly discriminate the digits in ARDIS, especially for the numbers in copperplate and cursive styles. According to the results tabulated in Table 3, the highest recognition accuracy rate is obtained using CNN model with MNIST which is 58.80%. Moreover, the lowest recognition accuracy rate is obtained using random forest with USPS which is 17.15%. The results prove that the machine learning methods with the existing datasets cannot provide high recognition accuracy on ARDIS dataset. Furthermore, the quantitative evaluation demonstrates that the methods learned from the data represented by descriptive features (e.g., HOG and CNN features) significantly outperform as compared with the methods learned from the raw pixel and normalized pixel features.

Figure 6 shows the confusion matrices generated using CNN method which is trained on the publicly available datasets and tested on ARDIS. Figure 6a illustrates the results of CNN trained on MNIST and tested on ARDIS. The results show that numbers 2, 6, 7, and 9 reduce the recognition rates. For instance, CNN model incorrectly identifies the number 2 as the digits 5 and 8, the number 6 as the digits 0 and 5, the number 7 as the digit 2, and the number 9 as the digits 7 and 8. Figure 6b depicts the confusion matrix of CNN, trained on USPS and tested on ARDIS. It is clear that most of the numbers are wrongly predicted.

The third experiment aims at understanding and analyzing the effectiveness and robustness of the learning and recognition methods using ARDIS dataset. In this experiment, 6600 samples are used for training and 1000 samples for testing. Table 4 compares the recognition accuracy rates of six methods on ARDIS. The results verify that the methods provide very high recognition rates. The highest recognition result is achieved using CNN model with 98.60% accuracy rate. The second-highest performance belongs to RBF kernel SVM with HOG features with the error rate of 4.5%. RBF kernel SVM on the raw pixels provides the accuracy rate of 92.40%. RNN acts slightly worse than SVM on the raw pixels and gives 91.12% recognition rate. The worse recognition performances are obtained using random forest and kNN methods with error rates of 13.00% and 10.40%, respectively. Even though the digits in this dataset are complex and written in various handwriting styles, the overall results show that the learning methods provide more effective and robust models, even though ARDIS has less training samples (6600) than MNIST (60,000).
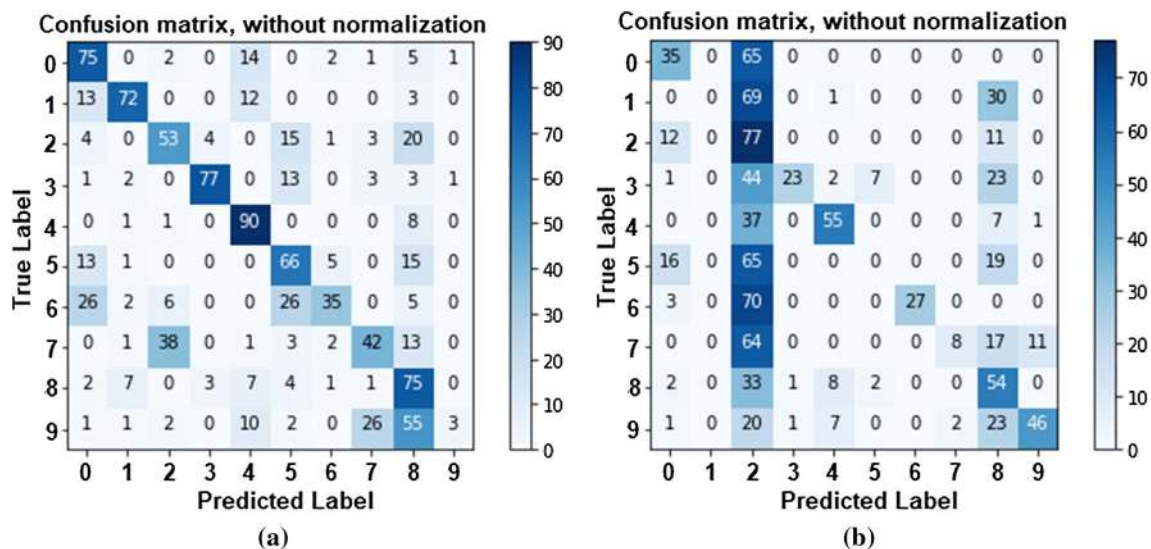
Fig. 6 Confusion matrix of the tested ARDIS samples with CNN classifier trained on: **a** MNIST and **b** USPS dataset

Table 4 Handwritten digit recognition using machine learning methods on ARDIS dataset

Method	Recognition accuracy %
CNN	98.60
SVM	92.40
HOG–SVM	95.50
kNN	89.60
Random forest	87.00
RNN	91.12

4.4 Performance of different CNN models on various digit datasets

In this section, the recognition performance of different CNN models on MNIST and ARDIS datasets is examined. In this experiment, two different scenarios are considered. In the first scenario, CNN classifier is trained on MNIST and the model is tested on ARDIS. In the second scenario, CNN is trained on ARDIS and tested on MNIST. For fair comparisons in both scenarios, 6600 training samples are used from each dataset, which is the size of training set in ARDIS dataset. The training samples are modeled using 1, 2, 3, and 4 convolutional layers, and each one is followed by two fully connected layers (each one has 128 nodes) and one output layer. Here, in all experiments, ReLU is used as an activation function in the convolutional and fully connected layers and Softmax function is used to obtain probabilities of each output class in the last layer of fully connected neural network. CNN with one convolutional layer uses 16 filters. CNN with two convolutional layer uses 16 and 32 filters. CNN with three convolutional layers

uses 16, 32, and 64 filters. CNN with four convolutional layers uses 16, 32, 64, and 64 filters. In the aforementioned CNN architectures, the kernel size is 3×3. Moreover, the epoch size, batch size, and learning rate are 10, 200, and 0.001, respectively. In the CNN models, the cross-entropy loss function is minimized using Adam optimizer and weights are initialized randomly. According to the accuracy rates in Fig. 7, the model generated by one and three convolutional layers using MNIST provides slightly better results than CNN trained on ARDIS. However, CNN with two and four convolutional layers trained on ARDIS and tested on MNIST gives better results than the models generated by MNIST. Aside from this, CNN with three and four convolutional layers provides accuracy rates of 59.50 and 54.81 for the first scenario and 57.26 and 57.21 for the second scenario, respectively. These results clearly illustrate that increasing convolutional layers in CNN does not always improve the classifier performance. Adding more convolutional layers to the network leads to higher training
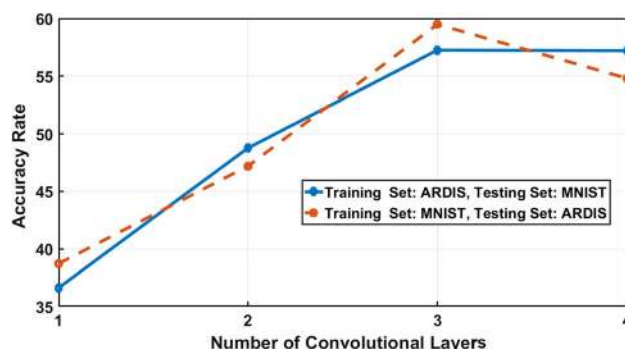

Fig. 7 Recognition accuracy results using different CNN models with different number of convolutional layers, performed on two datasets. The kernel size is set to 3×3

error due to the degradation and vanishing gradients which causes the optimization gets stuck in a local minimum [71, 72].

4.5 Merging datasets: the impact of different amount of training data

This section discusses the performance of the machine learning methods on various merged datasets. To generate the merged datasets, 15, 30, 60, and 100 percentages of the training samples from MNIST and ARDIS datasets are randomly selected and combined. Note that the classes are equally represented. This results in four different training datasets with different sizes. For instance, to obtain the first merged training samples, we randomly select 15% from each training dataset, which creates a merged dataset with 9900 training samples. Moreover, all the test samples in MNIST (10,000) and ARDIS (1000) datasets are used to compare the performance of the recognition methods. Furthermore, for 15%, 30%, and 60%, we run the algorithms 10 times and the averaged results are shown in Tables 5, 6, and 7.

Table 5 illustrates the recognition performance of the classifiers on 15% merged dataset (15% MNIST and 15% ARDIS). The results show that the compared methods on the merged dataset provide promising classification results. With this dataset, the recognition accuracy rates for CNN, HOG–SVM, SVM, RNN, kNN, and random forest are 97.62%, 95.73%, 94.48%, 94.12%, 93.59%, and 90.17%, respectively. Based on the results, the best performance belongs to CNN, whereas the worse recognition accuracy is obtained using random forest method. Besides this, the results indicate that combining ARDIS with MNIST, even with low percentages, leads to a learning model that can classify more diverse handwriting styles. Table 3 shows that CNN trained on MNIST and tested on ARDIS gave 58.80% accuracy rate; however, by adding only 15% of ARDIS dataset to MNIST, the recognition accuracy rate can be increased by 39.28%. In addition, the learning

Table 5 Handwritten digit recognition using machine learning methods on merged dataset: training set: 15% MNIST+ 15% ARDIS, testing set: MNIST+ARDIS

Method	Recognition accuracy (%)
CNN	97.62
SVM	94.48
HOG–SVM	95.73
kNN	93.59
Random forest	90.17
RNN	94.12

Table 6 Handwritten digit recognition using machine learning methods on merged dataset: training set: 30% MNIST+ 30% ARDIS, testing set: MNIST+ARDIS

Method	Recognition accuracy (%)
CNN	98.08
SVM	95.87
HOG–SVM	96.18
kNN	95.72
Random forest	92.21
RNN	96.05

Table 7 Handwritten digit recognition using machine learning methods on merged dataset: training set: 60% MNIST+ 60% ARDIS, testing set: MNIST+ARDIS

Method	Recognition accuracy %
CNN	98.47
SVM	96.23
HOG–SVM	97.38
kNN	96.01
Random forest	92.87
RNN	96.28

methods in Table 3 used 60,000 training samples which is computationally expensive, but the results in Table 5 are obtained using only 9900 training samples which decreases the computational cost.

Moreover, the results in Tables 6, 7, and 8 prove that increasing the number of training samples in the merged datasets raises the performance of all methods for handwritten digit recognition. Table 6 shows that the recognition accuracy rates for CNN, HOG–SVM, RNN, SVM, kNN, and random forest using 30% from each dataset are 98.08%, 96.18%, 96.05%, 95.87%, 95.72%, and 92.21%, respectively. This simply shows that increasing the number of training samples twice raises the accuracy of the

Table 8 Handwritten digit recognition using machine learning methods on merged dataset: training set: 100% MNIST+ 100% ARDIS, testing set: MNIST+ARDIS

Method	Recognition accuracy
CNN	99.34
SVM	96.48
HOG–SVM	98.08
kNN	96.63
Random forest	93.12
RNN	96.74

aforementioned classifiers by 0.46%, 1.07%, 1.93%, 1.39%, 2.13%, and 2.04%, respectively. Table 7 depicts that the accuracy rates for CNN, HOG–SVM, RNN, SVM, kNN, and random forest using 60% from each dataset are 98.47%, 97.38%, 96.28%, 96.23%, 96.01%, and 92.87%, respectively. These results indicate that increasing the number of training samples four times can improve the accuracy of the methods by 0.85%, 1.65%, 2.16%, 1.75%, 2.42%, and 2.70%, respectively. Table 8 illustrates that the accuracy rates for CNN, HOG–SVM, RNN, kNN, SVM, and random forest using 100% from each dataset are 99.34%, 98.08%, 96.74%, 96.63%, 96.48%, and 93.12%, respectively. This experiment demonstrate that combining all the training samples improves the accuracy of the machine learning methods by 1.72%, 2.35%, 2.62%, 3.04%, 2.00%, and 2.95%, respectively. From all the above experiments, we can conclude that the performance of kNN classifier highly depends on the number of training samples, whereas CNN method is the least sensitive method. RBF kernel SVM on the raw pixel features also shows that the number of training samples has low impact on its performance. This experimental setup also explains that combining the training set for handwriting digit recognition can be beneficial when the added data increase diversity of the original training data. For instance, the recognition rates in Table 3 are improved by adding ARDIS dataset to MNIST as ARDIS training data cover wide ranges of digits that are written with various writing styles, stroke thicknesses, orientations, sizes, and pen types. Furthermore, the same conclusion can be reached by comparing the results in Fig. 7 with the ones in Table 8.

5 Conclusion

In this paper, we introduced four different digit datasets in ARDIS which is the first publicly available historical digit dataset (https://ardisdataset.github.io/ARDIS/). They are constructed from the Swedish historical documents written between the year 1895 and 1970 and contain: (1) digit string images in RGB color space, (2) single-digit images with original appearance, (3) single-digit images with clean background without size normalization, and (4) single-digit images in the same format as MNIST. ARDIS dataset increases diversity by representing more variations in handwritten digits which can improve the performance of digit recognition systems. Moreover, in this paper, a number of machine learning methods trained on different digit datasets and tested on ARDIS dataset are evaluated and investigated. The results show that machine learning methods give poor recognition performance which indicates that the digits in ARDIS dataset have different features and characteristics as compared to the other existing

digit datasets. We encourage other researchers to use ARDIS dataset for testing their own affective handwritten digit recognition methods.

Compliance with ethical standards

Conflict of interest Johan Hall is an employee at the company Arkiv Digital AB (Sweden) which is mentioned in the article. The rest of authors declare that they have no conflict of interest.

References

1. Djeddi C, Al-Maadeed S, Gattal A, Siddiqi I, Ennaji A, Abed HE (2016) ICFHR2016 competition on multi-script writer demographics classification using "QUWI" database. In: IEEE international conference on frontiers in handwriting recognition, pp 602–606
2. Ghosh MMA, Maghari AY (2017) A comparative study on handwriting digit recognition using neural networks. In: IEEE International conference on promising electronic technologies, pp 77–81
3. Gattal A, Chibani Y, Hadjadji B (2017) Segmentation and recognition system for unknown-length handwritten digit strings. Pattern Anal Appl 20(2):307–323
4. Gattal A, Chibani Y, Hadjadji B, Nemmour H, Siddiqi I, Djeddi C (2015) Segmentation-verification based on fuzzy integral for connected handwritten digit recognition. In: International conference on image processing theory, tools and applications (IPTA), Orleans, pp 588–591
5. Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Jackel LD, LeCun Y, Muller UA, Sackinger E, Simard P, Vapnik V (1994) Comparison of classifier methods: a case study in handwritten digit recognition. In: IEEE International Conference on Pattern Recognition, pp 77–82
6. Niu XX, Suen CY (2012) A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognit 45:1318–1325
7. Salman SA, Ghani MU (2014) Handwritten digit recognition using DCT and HMMs. In: IEEE international conference on frontiers of information technology, pp 303–306
8. Keshta IM (2017) Handwritten digit recognition based on output-independent multi-layer perceptrons. Int J Adv Comput Sci Appl 8(6):26–31
9. Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 22(1):63–84
10. Liu CL, Nakashima K, Sako H, Fujisaw H (2003) Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognit 36(10):2271–2285

11. Babu UR, Venkateswarlu Y, Chintha AK (2014) Handwritten digit recognition using K-nearest neighbour classifier. In: IEEE world congress on computing and communication technologies, pp 60–65

12. Ilmi N, Budi WTA, Nur RK (2016) Handwriting digit recognition using local binary pattern variance and K-nearest neighbor classification. In: IEEE international conference on information and communication technology, pp 1–5

13. Ignat A, Aciobanitei B (2016) Handwritten digit recognition using rotations. In: IEEE International symposium on symbolic and numeric algorithms for scientific, pp 303–306

14. Impedovo S, Mangini FM (2012) A novel technique for handwritten digit classification using genetic clustering. In: IEEE international conference on frontiers in handwriting recognition, pp 236–240

15. Parvin H, Alizadeh H, Minaei-Bidgoli B (2008) MKNN: modified K-nearest neighbor. In: Proceedings of the world congress on engineering and computer science, pp 1–4

16. Bernard S, Heutte L, Adam S (2017) Using random forests for handwritten digit recognition. In: IEEE international conference on document analysis and recognition, pp 1–5

17. Prudent Y, Ennaji A (2005) A K-nearest classifier design. Electron Lett Comput Vis Image Anal 2(5):58–71

18. Taha MAM, Teuscher C (2017) Naive Bayesian inference of handwritten digits using a memristive associative memory. In: IEEE international symposium on nanoscale architectures, pp 139–140

19. Liu ZQ, Cai J, Buse R (2003) Markov random field model for recognizing handwritten digits, handwriting recognition. Stud Fuzziness Soft Comput 133:1–5

20. Pauplin O, Jiang J (2010) A dynamic bayesian network based structural learning towards automated handwritten digit recognition. In: International conference on hybrid artificial intelligence systems, pp 220–227

21. Maji S, Malik J (2009) Fast and accurate digit classification, EECS Department, University of California, Berkeley, Technical Report UCB/EECS-2009-159. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-159.html

22. Ebrahimzadeh R, Jampour M (2014) Efficient handwritten digit recognition based on histogram of oriented gradients and SVM. Int J Comput Appl 9(104):10–13

23. Gorgevik D, Cakmakov D (2005) Handwritten digit recognition by combining SVM classifiers. In: IEEE international conference on computer as a tool, pp 1393–1396

24. Sharma A (2012) Handwritten digit recognition using support vector machine, pp 1–7. arXiv preprint arXiv:1203.3847

25. Neves RFP, Lopes Filho ANG, Mello CAB, Zanchettin C (2011) A SVM based off-line handwritten digit recognizer. In: IEEE International conference on systems, man, and cybernetics, pp 510–515

26. Neves RFP, Zanchettin C, Lopes Filho ANG (2012) An efficient way of combining SVMs for handwritten digit recognition. In: International conference on artificial neural networks and machine learning, pp 229–237

27. Tuba E, Tuba M, Simian D (2016) Handwritten digit recognition by support vector machine optimized by bat algorithm. In: International conference in central Europe on computer graphics, visualization and computer vision, pp 1–8

28. Gattal A, Djeddi C, Chibani Y, Siddiqi I (2016) Isolated handwritten digit recognition using oBIFs and background features, 12th IAPR workshop on document analysis systems (DAS), Santorini, pp 305–310

29. Knerr S, Personnaz L, Dreyfus G (1992) Handwritten digit recognition by neural networks with single-layer training. IEEE Trans Neural Netw 6(3):962–968

30. Islam KT, Mujtaba G, Raj RG, Nweke HF (2017) Handwritten digits recognition with artificial neural network. In: IEEE international conference on engineering technology and technopreneurship, pp 1–4

31. AL-Mansoori S (2015) Intelligent handwritten digit recognition using artificial neural network. Int J Eng Res Appl 5(5):46–51

32. Alonso-Weber JM, Sesmero MP, Gutierrez G, Ledezma A, Sanchis A (2013) Handwritten digit recognition with pattern transformations and neural network averaging. In: International conference on artificial neural networks and machine learning, pp 335–342

33. Ciresan DC, Meier U, Gambardella LM, Schmidhuber J (2010) Deep big simple neural nets for handwritten digit recognition. J Neural Comput 22:1–14

34. McDonnell MD, Tissera MD, Vladusich T, van Schaik A, Tapson J (2015) Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the extreme learning machine algorithm. PLoS ONE 8(10):1–20

35. Zhan H, Wang Q, Lu Y (2017) Handwritten digit string recognition by combination of residual network and RNN-CTC. In: International conference on neural information processing, pp 583–591

36. Ren J (2012) ANN vs. SVM: which one performs better in classification of MCCs in mammogram imaging. J Knowl Based Syst 26:144–153

37. Diem M, Fiel S, Kleber F, Sablatnig R, Saavedra JM, Contreras D, Barrios JM, Oliveira LS (2014) ICFHR 2014 competition on handwritten digit string recognition in challenging datasets. In: IEEE international conference on frontiers in handwriting recognition, pp 779–784

38. Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (2012) Deep big multilayer perceptrons for digit recognition. Lecture notes in computer science. Springer, Berlin, pp 581–598

39. Holmstrom L (1997) Neural and statistical classifiers-taxonomy and two case studies. IEEE Trans Neural Netw 8:5–17

40. Ciresan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification, Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland, Technical report IDSIA

41. Wang Z, Chang S, Yang Y, Liu D, Huang TS (2016) Studying very low resolution recognizing using deep networks. In: Proceedings of computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, pp 1–9

42. Sermanet P, Chintala S, LeCun Y (2012) Convolutional neural networks applied to house numbers digit classification. In: IEEE international conference on pattern recognition, pp 3288–3291

43. Chellapilla K, Puri S, Simard P (2006) High performance convolutional neural networks for document processing. In: 10th international workshop on frontiers in handwriting recognition, La Baule (France). Université de Rennes 1, Suvisoft

44. Hochuli AG, Oliveira LS, Britto AS Jr, Sabourin R (2018) Handwritten digit segmentation: is it still necessary? Pattern Recognit 78:1–11

45. Jalalvand A, Demuynck K, De Neve W, Martens JP (2018) On the application of reservoir computing networks for noisy image recognition. Neurocomputing 277:237–248

46. Cirstea BI, Likforman-Sulem L (2016) Improving a deep convolutional neural network architecture for character recognition. Electron Imaging 2016:1–7

47. Ciresan DC, Meier U, Gambardella LM, Schmidhuber J (2011) Convolutional neural network committees for handwritten character classification. In: International conference on document analysis and recognition, Beijing, pp 1135–1139

48. Cheddad A, Kusetogullari H, Grahn H (2017) Object recognition using shape growth pattern. In: IEEE international symposium on image and signal processing and analysis, Ljubljana, pp 47–52

49. Fischer A, Indermuhle E, Bunke H, Viehhauser G, Stolz M (2010) Ground truth creation for handwriting recognition in historical documents. In: International workshop on document analysis systems, pp 3–10

50. Vajda S, Rangoni Y, Cecotti H (2015) Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: application to handwritten character recognition. Pattern Recognit Lett 58:23–28

51. LeCun Y (2003) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed 15 Aug 2018

52. LeCun Y, Jackel L, Bottou L, Brunot A, Cortes C (1995) Comparison of learning algorithms for handwritten digit recognition. In: International conference on artificial neural networks, France, pp 53–60

53. Liu C-L, Nakashima K, Sako H, Fujisawa H (2004) Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognit 37:265–279

54. CENPARMI (2003) The Cenparmi database of handwritten digits. http://www.concordia.ca/research/cenparmi/resources/herosvm.html. Accessed 15 Aug 2018

55. Hull JJ (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554

56. Srl T (1994) The Semeion database of handwritten digits. http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit. Accessed 15 Aug 2018

57. Wang C (2014) Handwriting digital recognition via modified logistic regression. J Multimed 9:8

58. CEDAR, Database of handwritten digits. https://cedar.buffalo.edu/Databases/CDROM1/. Accessed 15 Aug 2018

59. Gaudin CV, Lallican PM, Knerr S, Binter P (1999) The IRESTE On/Off (IRONOFF) dual handwriting database. In: International conference on document analysis and recognition, Bangalore, pp 455–458

60. Javidi Reza MM, Reza Ebrahimpour, Fatemeh Ebrahimpour (2011) Persian handwritten digits recognition: a divide and conquer approach based on mixture of MLP experts. Int J Phys Sci 6:30

61. Liu CL, Yin F, Wang DH, Wang QF (2013) Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recognit 46(1):155–162

62. Abdleazeem S, El-Sherif E (2008) Arabic handwritten digit recognition. Int J Doc Anal Recognit 11(3):127–141

63. El-Sherif EA, Abdelazeem S (2007) A two-stage system for Arabic handwritten digit recognition tested on a new large database. In: International conference on artificial intelligence and pattern recognition, Florida, USA, pp 9–12

64. Abdelazeem S, El-Sherif E Database of handwritten digits. http://datacenter.aucegypt.edu/shazeem/. Accessed 15 Aug 2018

65. Djeddi C, Gattal A, Souici-Meslati L, Siddiqi I, Chibani Y, El Abed H (2014) LAMIS-MSHD: a multi-script offline handwriting database. In:14th international conference on frontiers in handwriting recognition, Heraklion, pp 93–97

66. de Campos TE, Babu BR, Varma M (2009) Character recognition in natural images. In: International conference on computer vision theory and applications (VISAPP), Lisbon, Portugal, pp 1–8

67. Chars74k (2012) The chars74k database of character recognition in natural images. http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/. Accessed 15 Aug 2018

68. Matlab, The synthetic database of handwritten digits. https://se.mathworks.com/help/deeplearning/ref/trainnetwork.html. Accessed 15 Aug 2018

69. Ribas FC, Oliveira LS, Britto AS, Sabourin R (2013) Handwritten digit segmentation: a comparative study. Int J Doc Anal Recognit 16(2):567–578

70. Doncea SM, Ion RM (2014) FTIR (DRIFT) analysis of some printing inks from the 19th and 20th centuries. Revue Roumaine de Chimie 4(59):173–183

71. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. In: Computer vision and pattern recognition, pp 770–778

72. Srivastava RK, Greff K, Schmidhuber J (2015) Training very deep networks. In: International conference on neural information processing systems, pp 1–11