Big Data Research 23 (2021) 100182

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset[☆]

Huseyin Kusetogullari[a,*], Amir Yavariabdi[b], Johan Hall[c], Niklas Lavesson[d]

[a] *Department of Computer Science, Blekinge Institute of Technology, 37141 Karlskrona, and School of Informatics, Skövde University, 54128 Skövde, Sweden*
[b] *Department of Mechatronics Engineering, KTO Karatay University, Konya, Turkey*
[c] *Arkiv Digital, Växjö, Sweden*
[d] *Department of Computer Science, School of Engineering, Jönköping University, SE-553 18 Jönköping, Sweden*

ARTICLE INFO

Article history:
Received 15 March 2020
Received in revised form 22 August 2020
Accepted 29 November 2020
Available online 28 December 2020

Keywords:
Historical handwritten documents
Handwritten digit detection
Ensemble deep learning
Digit string recognition
DIDA handwritten digit dataset

ABSTRACT

This paper introduces a novel deep learning architecture, named DIGITNET, and a large-scale digit dataset, named DIDA, to detect and recognize handwritten digits in historical document images written in the nineteen century. To generate the DIDA dataset, digit images are collected from 100, 000 Swedish handwritten historical document images, which were written by different priests with different handwriting styles. This dataset contains three sub-datasets including single digit, large-scale bounding box annotated multi-digit, and digit string with 250, 000, 25, 000, and 200, 000 samples in Red-Green-Blue (RGB) color spaces, respectively. Moreover, DIDA is used to train the DIGITNET network, which consists of two deep learning architectures, called DIGITNET-dect and DIGITNET-rec, respectively, to isolate digits and recognize digit strings in historical handwritten documents. In DIGITNET-dect architecture, to extract features from digits, three residual units where each residual unit has three convolution neural network structures are used and then a detection strategy based on You Look Only Once (YOLO) algorithm is employed to detect handwritten digits at two different scales. In DIGITNET-rec, the detected isolated digits are passed through 3 different designed Convolutional Neural Network (CNN) architectures and then the classification results of three different CNNs are combined using a voting scheme to recognize digit strings. The proposed model is also trained with various existing handwritten digit datasets and then validated over historical handwritten digit strings. The experimental results show that the proposed architecture trained with DIDA (publicly available from: https://didadataset.github.io/DIDA/) outperforms the state-of-the-art methods.

1. Introduction

In the last two decades, there has been tremendous escalation in digitalization of handwritten documents to preserve the valuable historical information [1]. Even though the use of digital documents is convenient and efficient and they can improve the ability to preserve valuable old documents and ease the access, it is of great importance to develop handwritten recognition methods to make historical manuscript images available for indexing and searching, which lead to efficient information retrieval. More specifically, to achieve this objective, a handwritten recognition framework must automatically extract textual (e.g. characters, words, and sentences) and/or numerical (single- and multi-digits) contents from handwritten document images. However, this is a very challenging problem due to large intra- and inter-intensity variations as well as inter-class similarities and intra-class disparities in images.

In the context of handwritten document image analysis, one of the most important and well-known problems is to detect and recognize digit strings in the handwritten document images [2]. Digit detection and recognition have been used in many applications such as document indexing based on dates (e.g. document date, birthdate, marriage date and death date) and automated reading of postal codes and amounts in bank cheques, tax forms, and census forms [3]. In order to avoid time-consuming and inefficient search processes, it is a vital task to develop an automatic handwritten digit string detection and recognition system. Therefore, this pa-

[☆] This project is funded by the research project scalable resource efficient systems for big data analytics by the Knowledge Foundation (Grant: 20140032) in Sweden.

[*] Corresponding author.
E-mail address: huseyin.kusetogullari@bth.se (H. Kusetogullari).

https://doi.org/10.1016/j.bdr.2020.100182

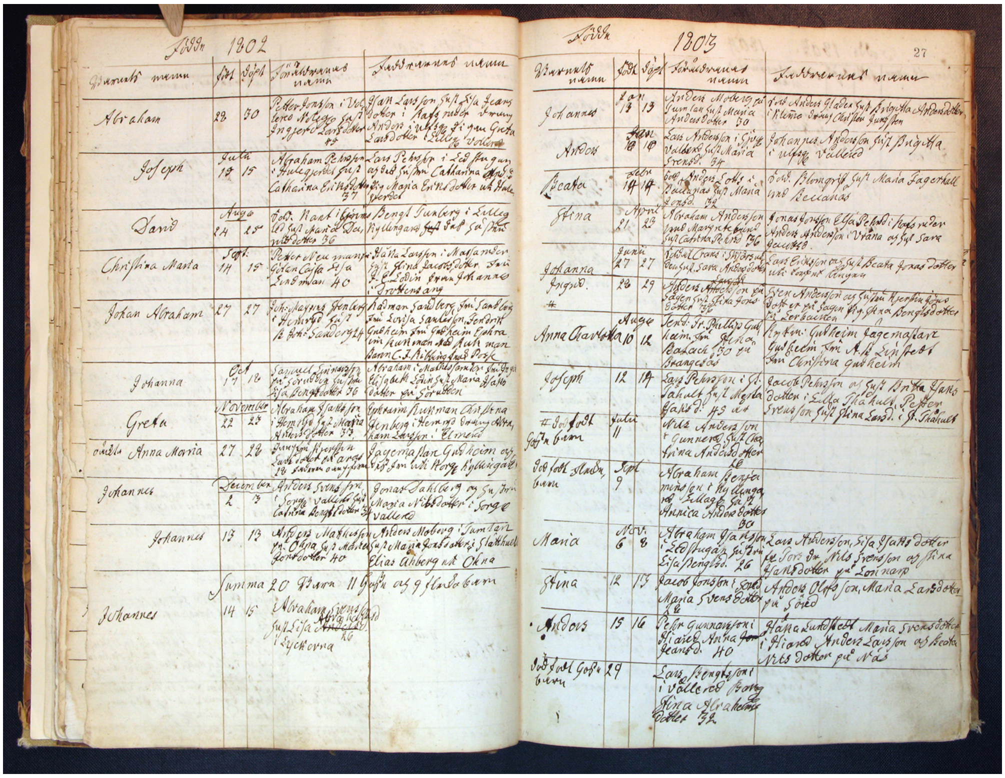
1802
1803
27

Fig. 1. Illustration of a Swedish historical handwritten document written in 1802 and 1803.

per proposes a new deep learning based framework to resolve the corresponding problem.

Generally, automatic handwritten numeral string recognition frameworks can be classified into two different categories: 1) segmentation-based and 2) detection-free recognition approaches [4]. In the former one, many methods based on image processing techniques have been developed to automatically separate digits and then recognize the isolated digits [5,6]. Typically, in this approach, methods consist of four different steps: 1) pre-processing (e.g. image de-noising), 2) digit detection, 3) feature extraction, and 4) classification. Amongst these four steps, the most critical one is the detection step since the performance of the recognition approach highly depends on the accuracy of the detection techniques. Detection of individual digits on digit strings is a challenging problem since there are many issues such as touching digits, arbitrary orientations, various digit scales, significantly variant aspect ratios of digits, existence of digits with partial appearance in digit images, digit overlapping, ink flying in time, bleed-through, faint digits and many others. All these complexities bring difficulties in accurate positioning and recognition of individual digits. The latter approach has been applied to recognize the numeral strings without pre-processing and detection steps [7,8]. Note that, our proposed framework focuses on detection-based digit string recognition in Swedish handwritten historical document images (see Fig. 1).

1.1. Related works

To recognize digits in document images, many segmentation-based digit string recognition methods have been proposed and developed. For instance, Kim et al. [9] proposed a segmentation-based handwritten digit string recognition approach of connected two digits. The handwritten digit string image is first binarized and then separation points between two touching digits have been obtained using counter analysis, candidate break points analysis, and ligature analysis techniques. The method has been applied to 3, 500 touching pairs of handwritten digits of the NIST SD19 database [10] and the results show that the method provides high detection accuracy to separate two touching digits into individual digits. Lacerda et al. [11] presents an algorithm for segmentation of connected handwritten digits based on a feature extraction technique. In this method, two different features are obtained using skeletonization and self-organizing maps. The first feature extraction is used to extract topological and geometric features in digit images whereas the second one is employed to extract features at touching regions. The results show that the method can deal with several types of touching digits. However, these two methods only consider segmentation of touching digits and they do not consider other important issues in the context of handwritten document image analysis. A segmentation and recognition system has been presented for unknown-length handwritten digit strings [12]. In order to segment the digits, configuration touching points between the digits have been considered and three different segmentation techniques such as contour analysis, sliding window Radon transform and histogram of the vertical projection are used. These three different segmentation techniques are combined to obtain the final segmentation results. Finally, the Support Vector Machine (SVM) classifier is used to recognize the segmented digits. Merabti et al. [13] proposed a handwritten numeral digit segmentation and recognition system for unconstrained handwritten connected digits. The method finds two structural features which are used to find possible cutting points of connected digits. The performance of the segmentation approach is evaluated using a digit recognition method which is the Fuzzy-Artificial Immune System (Fuzzy-AIS). The method is applied to the handwritten digit database NIST SD19 [10]. Generally, the existing segmentation-based digit recognition methods only focus on detection and/or recognition of touching digits, broken digits and unknown-length handwritten digit strings in modern document images. However, historical handwritten document images consist of many other challenges such as large variations in paper texture, document aging, handwriting style, digit thickness, orientation, and digit appearance

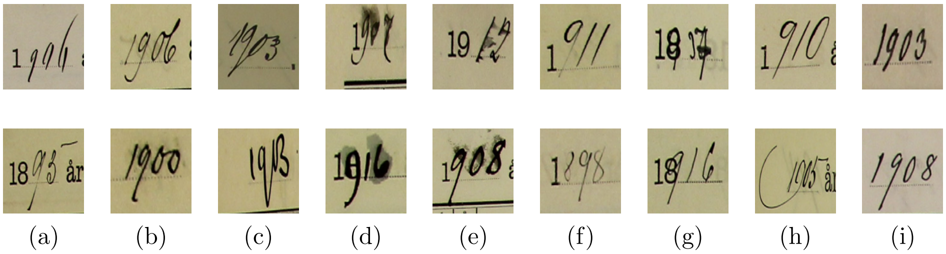

Fig. 2. Illustration of different challenges for digit segmentation and recognition: (a) broken digits, (b) single-touch digit, (c) multi-touch digits, (d) degradation, (e) ink flowing, (f) faint, (g) overlapping, (h) size variability, and (i) digits without artefacts.

along with ink bleed through, stains and faded inks, which makes the state-of-the-art methods not applicable in historical document images. Moreover, handwritten digit instances in document images often lie very close to each other, making them very difficult to separate via existing methods. Fig. 2 illustrates some of these challenging issues. Moreover, state-of-the-art methods have been using image processing techniques to detect the digits and this strategy makes the automatic handwritten digit string recognition methods not efficient and not reliable. Therefore, it is necessary to apply a learning based approach to detect the digits.

Recently, segmentation-free digit string recognition models have also been proposed and developed to recognize numeral strings. In this approach, it is necessary to use large amount of training data in order to obtain high accuracy rate since there are 100, 1.000 and 10.000 possible n digit combinations from 0 to 9, when n is two, three and four, respectively. However, it is very difficult to find such a large dataset but this dataset can be created synthetically to resolve this issue. The synthetic data is generated using single digit database (e.g. NIST) where digit images are size-normalized. However, in the real numeral string recognition problem, handwritten digits are with different sizes and they may have corrupted with various artefacts which increase the complexity of the problem. Based on segmentation-free digit recognition approach, Hochuli et al. [14] designed four different Convolutional Neural Networks (CNN) to analyze the performance of the recognition performance without employing digit segmentation. Matan et al. [15] designed a Convolutional Neural Network for large input fields containing unsegmented characters. The results show that the recognition accuracy of the method is 66% for 3000 images of ZIP Codes. Another segmentation-free method is given by Choi et al. [7] where an ANN strategy with 100 hidden layers is used to model double-digit strings. In this work, to train the ANN model, a synthetic training dataset is generated by combining pairs of double-digit numbers from the NIST dataset. The results show that the recognition accuracy on 1374 pairs of digits is 95.3%. Ciresan et al. [8] proposes a CNN approach to recognise double-digit strings. To train a 100-class CNN model, a synthetic double-digit string dataset base on the NIST dataset, which includes 200, 000 isolated digits and the touching pairs images, is used. Many other handwritten digit segmentation methods have been applied [16–20].

1.2. Contribution

This paper presents a new automatic handwritten digit string recognition framework, called DIGITNET. The proposed framework is, to the best of our knowledge, the first work in which the CNN is designed to separate numeral strings to single digits in document images and recognize them. The proposed method has two steps which are: 1) DIGITNET-dect, in which a new learning architecture based on residual networks [21] and the YOLOv3 detection algorithm [22] is designed and 2) DIGITNET-rec, in which an ensemble CNN digit recognition framework based on three CNN models and majority voting is used. To train the proposed DIGITNET architectures, a new historical handwritten digit dataset (DIDA), which consists of 250, 000 single digits, 25, 000 images with bounding box annotations, and 200, 000 digit strings, is introduced and used.

The main contributions and strengths of the paper are summarized as follows:

1) The paper proposes a new automatic historical handwritten digit detection and recognition framework named DIGITNET.
2) We design the DIGITNET framework based on deep learning network architectures which includes DIGITNET-dect and DIGITNET-rec. The DIGITNET-dect is used to detect the digits in historical handwritten document images and the DIGITNET-rec is used to recognize the detected digits.
3) We introduce the largest publicly available handwritten digit dataset (DIDA), which provides a large number of highly diverse, accurate and detailed annotations of digits and digit strings in historical document images.
4) This is a time-saving and cost effective work in the big data applications for the industrial-based companies which are archiving handwritten document images. For instance, the Arkiv Digital company in Sweden has collected over 90 million historical Swedish handwritten document images and this number increases every single day. Thus, the automated digit string recognition system can be practical and useful for such companies to search for dates and numbers in handwritten document images.

2. Big image database and dataset generation

In the last few decades, various standard handwritten digit datasets have been generated which are usually used to resolve segmentation, detection, and recognition problems. The most important and well-known existing handwritten digit datasets are MNIST [23], NIST SD19 [10], and USPS [24] which are also publicly available to the optical character recognition research community. Note that, extensive research about the existing handwritten digit datasets can be found in [25]. These datasets are constructed from non-degraded handwritten documents written by rollerball and ballpoint pens with clean background and modern handwritten styles. Moreover, these datasets are size normalized. These characteristics of existing datasets simply limit the application of existing methods for handwritten digit detection and recognition in historical documents analysis where the variability and complexity becomes more prominent. Therefore, to support the research in handwritten digit detection and recognition, it is vital to create a new digit dataset based on historical documents to solve the problem of the existing ones.

2.1. Big handwritten document image database

Arkiv Digital (AD) is a Swedish company which is recording the handwritten Swedish church, census, military, and court documents as images using digital cameras [26]. The images have been captured in RGB color space with the resolutions of 5184 × 3456

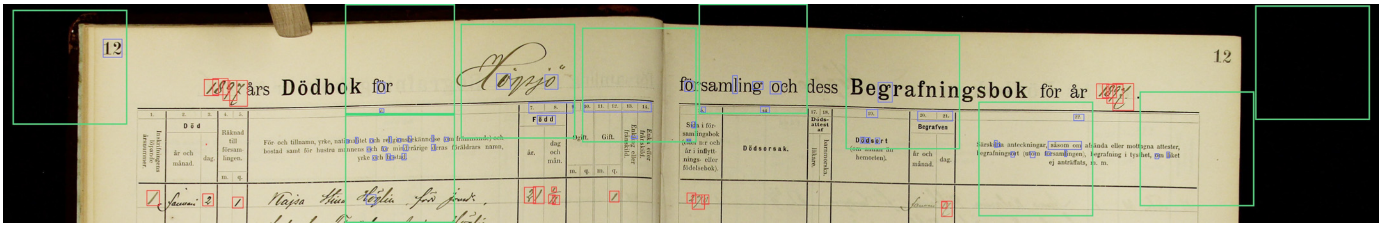

Fig. 3. Illustration of part of Swedish historical handwritten document written in 1897. Red boxes show cropped single digits in ARDIS dataset. Green boxes show 10 different selected regions used to fuse the ARDIS digits. Blue boxes depict the undesired characters and digits deleted in the cropped images. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

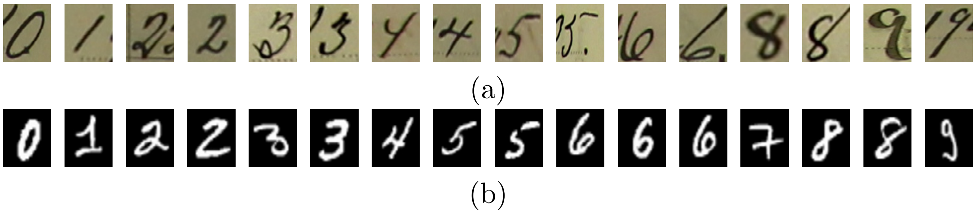

Fig. 4. Illustration of digit values from 0 – 9: (a) DIDA, (b) size-normalized and cleaned handwritten digit dataset (MNIST, USPS and ARDIS).

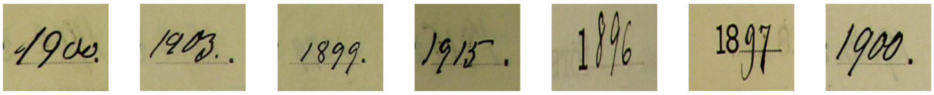

Fig. 5. Illustration of fixed-length digit string images (year digit strings).

pixels. The recorded documents are dated from 1186 to 2016. The AD large-scale handwritten document image database consists of more than 90 million Swedish historical document images. In this case, the database has 90, 000, 000 × 5184 × 3456 × 3 pixels which is very hard to process with a human effort. The large-scale image document database is certainly a very important and valuable resource for different researchers in the various research areas such as genealogy, history, and computer science [26].

2.2. DIDA: historical handwritten digit dataset

The DIDA dataset has been collected from the Swedish historical handwritten document images between the year 1800 and 1940 and it is the largest historical handwritten digit dataset which is introduced to the Optical character recognition (OCR) community to develop handwritten digit recognition methods. This dataset is the extended version of the ARDIS digit dataset [26]. The ARDIS dataset [26] has been verified and shown as a unique and different digit dataset which consists of 7600 single digits. However, the main issue of this dataset is that there are only 7, 600 single digits and it is not enough to obtain an efficient and effective deep learning detection model. Therefore, this dataset must be extended and labelled to train a deep learning-based detection algorithm. To generate DIDA, 250, 000 single digits (see red boxes in Fig. 3) and 200, 000 multi-digits are cropped from different document images. Note that, the DIDA dataset is publicly available to the research community to further advance handwritten digit detection, segmentation and recognition algorithms (publicly available from: https://didadataset.github.io/DIDA/). The DIDA dataset consists of 3 different subdatasets: 1) single digit samples, 2) digit string samples, and 3) digit images with bounding box annotations.

Dataset I: This dataset contains only single digit samples. To construct this dataset the handwritten digits from 0 to 9 are manually cropped from 100, 000 Swedish document images (red boxes in Fig. 3) and labelled. This subdataset involves isolated, connected, broken, and overlapped digits. Moreover, the digit images are recorded without size normalization and denoising. Therefore, the digits are recorded with many artifacts such as noise, dash lines, partial view of other digits, bleed-through, faint and others. Existing of such artifacts in the dataset is vital as in the real-world problem the digits can be corrupted with many artifacts and discarding them can decrease accuracy of detection and recognition algorithms. Furthermore, the DIDA dataset has 200, 000 handwritten digit samples with 10 different classes from 0 to 9, and each class contains 20, 000 single digit images. To the best of our knowledge, this dataset is the largest one to present handwritten single digit samples in RGB color space with the original sizes and appearances. This simply in contrast with the existing publicly available handwritten digit datasets, where the digit images are size-normalized, denoised and cleaned so they can not be used in real world cases (see Fig. 4 (a)). This dataset is generated in order to resolve handwritten digit detection and recognition problems in the document images.

Dataset II: This dataset consists of multi-digit strings which is divided into two parts: 1) fixed-length and 2) variable-length digit strings. The former one includes the four digit or year strings which are appeared on the top-left and top-right on the Swedish handwritten document images with two different forms: 1) entirely handwritten, and 2) typewritten with handwritten. Fig. 5 depicts several year digit string samples written in both forms. In order to create 4-digit year string dataset, a supervised-based learning approach has been applied to automatically detect the years on the top-left and top-right of 100, 000 Swedish handwritten document images. Then, the detected year digit strings are cropped with the size of 240 × 210 pixels and stored in the original structure. The fixed-length digit string dataset contains 130, 000 samples where they are manually labelled. The label vector is one-dimensional array of the corresponding years on each document. The latter one consists of 70, 000 variable-length handwritten digit strings with minimum length of 2 digits and maximum length of 10 digits, shown in Fig. 6. These digit strings are manually collected from any location on Swedish document images except the year digit strings locations. To the best our knowledge, this is the

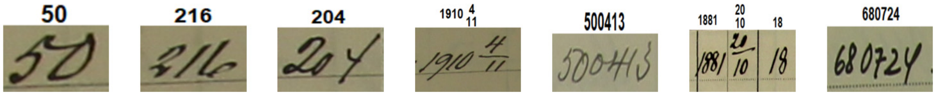

Fig. 6. Illustration of variable-length digit string images.

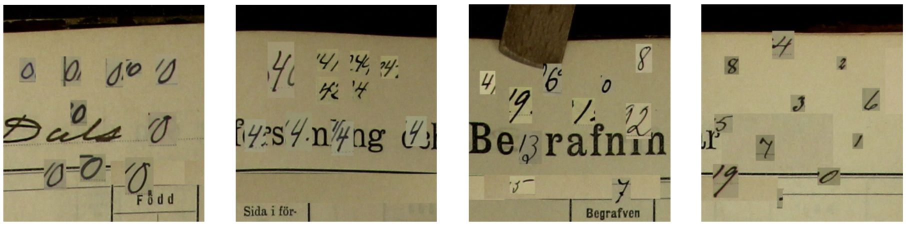

Fig. 7. Illustration of annotated single handwritten digit samples in the 416 × 416 images involving positive (digits) and negative (background) samples.

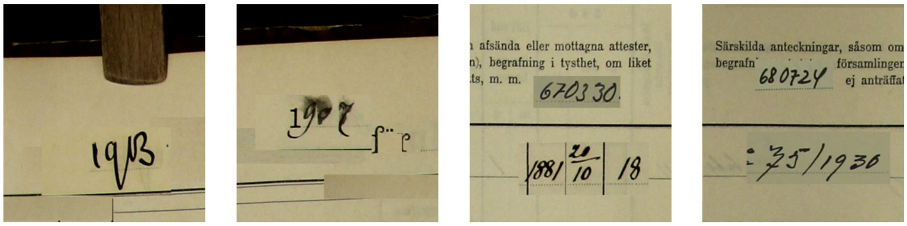

Fig. 8. Illustration of superimposed digit strings as test images for deep learning object detection algorithms.

first real handwritten multi-digit dataset that contains numerous issues such as single-touching, multi-touching, faint, degradation, broken, isolated, overlapping, variation of digit size to detect, segment and recognize the digit strings. Moreover, they are stored in the original form and manually labelled. Overall, Dataset II contains 200, 000 fixed-length, and variable-length digit strings which can be used by researchers in various applications such as digit segmentation from digit string samples, image binarization, and digit string recognition.

Dataset III: To the best of our knowledge, this is the first digit dataset created for the use of object detection algorithms. To generate this dataset, firstly, 100 different historical document images are randomly selected and then in each selected image, 10 regions of interests are cropped to 416 × 416 pixels (see green boxes in Fig. 3). Note that, these images initially should not include any digit so they are deleted in the cropped images as shown with the blue boxes in Fig. 3. Next, each 10 digits from dataset I are manually superimposed to one of the cropped images which is shown in Fig. 7 and then bounding box coordinates and labels are recorded. This simply generates 25, 000 augmented digit images for all digit classes. In this paper, the heights and widths of the augmented and digit images in dataset I are denoted as $H_{aug} = 416$, H_{digit}, $W_{aug} = 416$, and W_{digit}. Moreover, to generate the ground-truth label, the category numbers as well as digits' normalized coordinate, width and height are saved in a text file with the following format: [*category number*] [*digit center in X (X)*] [*digit center in Y (Y)*] [*digit width (W)*] [*digit height (H)*]. These coordinates are formulated as follows:

$$\begin{aligned}
dw &= \frac{1}{W_{aug}} \\
dh &= \frac{1}{H_{aug}} \\
X &= \frac{x_{digit} + (x_{digit} + W_{digit})}{2} \times dw \\
Y &= \frac{y_{digit} + (y_{digit} + H_{digit})}{2} \times dh \\
W &= W_{digit} \times dw \\
H &= H_{digit} \times dh
\end{aligned} \tag{1}$$

where x_{digit} and y_{digit} are the top left coordinates of the superimposed digit in the augmented image. It is important to mention that each text file consists of 10 labels. The main reasons that the digits are not labelled in the original historical church documents are threefold. The first reason is due to the large resolution of the original documents which is 5184 × 3456 pixels. This large resolution document images can massively increase computational cost of object detection algorithms and cause memory shortage problem. The second reason is to avoid down-sampling which changes the appearance and characterizes of the digits. The third reason is that in many documents some digits cannot be labelled due to bad appearance of digits. Therefore, these digits will be used as negative samples in detection algorithms which can lead to low detection accuracy. Furthermore, multi-digit images in dataset II are also fused on the cropped images which are used as testing data (see Fig. 8)). Note that, bounding box coordinates and label of each digit in multi-digit images are recorded.

The DIDA dataset has several advantages over the existing handwritten digit datasets (e.g. MNIST, USPS and ARDIS) including 1) it is the largest historical digit dataset with 250, 000 samples, 2) it contains 200, 000 multi-digit dataset cropped from the Swedish historical document images, and 3) it can directly used object detection algorithms as the digit bounding box annotations are recorded.

3. DIGITNET: digit detection and digit string recognition

The purpose of this work is to detect individual digits in the digit strings and recognizing them in order to have automatic indexing. In this manner, the handwritten Swedish historical document images in ARKIV digital company database can be easily indexed and automatically sorted via date or any other digit strings. Moreover, people can easily search information based on birth-date, death-date or document recorded date in the documents. Digit segmentation and recognition in historical document images

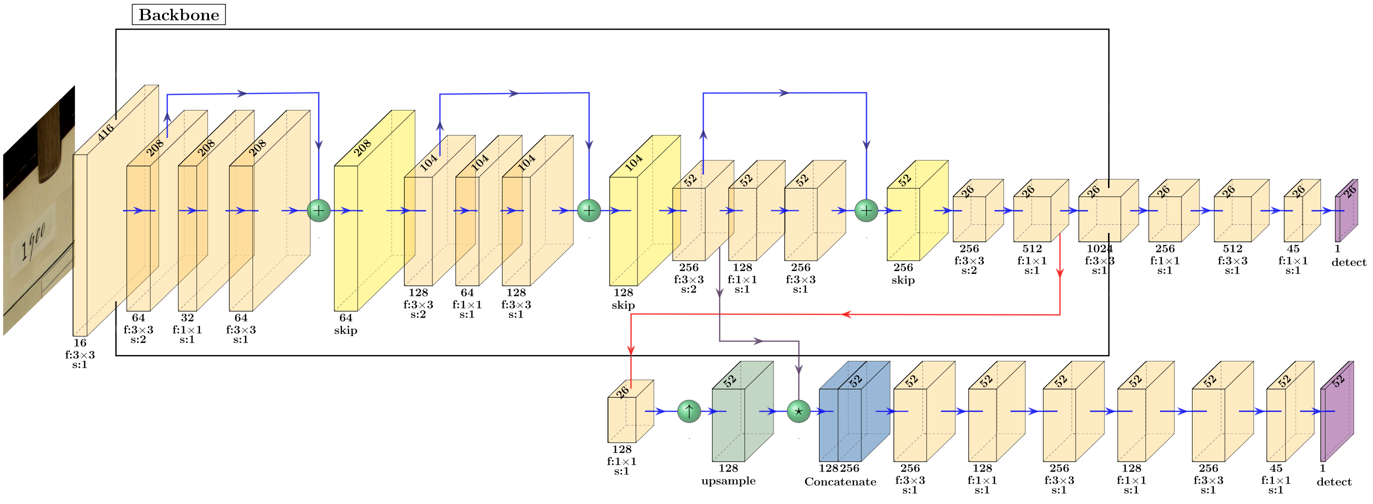

Fig. 9. Proposed handwritten digit detection network architecture. In layers, f and s denote filter size and stride, respectively.

are very challenging tasks because of highly visual complexities such as document aging, bleed through, faint, and large variations in handwriting styles, paper textures, and digit appearances. All these complexities bring difficulties in accurate positioning and recognition of individual digits. To overcome these difficulties, a new two-stage framework, called DIGITNET, is proposed to automatically detect (DIGITNET-dect) and recognize (DIGITNET-rec) handwritten digits in Swedish handwritten historical document images. In DIGITNET-dect, a 2-scale regression-based deep detection method with a new deep residual network model is used, whereas, in DIGITNET-rec, an ensemble CNN architecture is utilized.

3.1. DIGITNET-dect: digit detection network architecture

Currently, regression-based deep CNN methods (e.g. [22,27–33]) have been widely used in object detection problem. Amongst all methods, YOLOv3 object detection framework [22] is one of the most effective object detection algorithms in terms of speed and accuracy, as it encompasses many of the best techniques in computer vision literature [34]. To date, YOLOv3 [22] and its lighter version (YOLOv3-tiny) have been used for detecting and localizing objects in a wide range of application areas including medical imaging [35], remote sensing [36], natural scene images [37], and many others [38–40]. However, to the best our knowledge, this work is the first success attempt to use YOLOv3 detection strategy [22] to detect handwritten digits in historical document images.

YOLOv3 framework [22], which is a one step process, is proposed to predict the bounding boxes and class probabilities simultaneously from the input image. This framework splits the input image into $G \times G$ cells which are used in a 2D fully convolutional neural network architecture to predict 3 bounding boxes for each cell. Besides of this, if the cell contains an object, a conditional class probability for C classes is estimated. In this manner, the network predicts center coordinates (x, y), width and height (w, h) and a confidence score for each bounding box. The confidence score is defined as the product of the probability that the bounding box contains the object of interest (objectness score) and intersection of predicted box and ground truth box (IoU). In contrast to other YOLO versions, YOLOv3 [22] predicts the objectness score for each bounding box using a multilabel classification approach based on logistic regression rather than softmax to better model the data. The model in this framework returns a tensor of shape $G \times G \times (3 \times (4 + 1 + C))$. In the final step of YOLOv3, the fine tune bounding box(es) is generated using thresholding class-specific confidence score and non-maximum suppression. Note that the class-specific confidence score, which reflects how possible the object belonging to the class exists in individual box confidence, is estimated via the product of the individual box confidence and conditional class probability.

YOLOv3 [22] predicts bounding boxes at 3 different scales using a network architecture which combines residual [21] and feature pyramid networks [41]. This network architecture contains 75 convolutional layers and 31 other layers including shortcut, concatenation, upsampling, and detection. In this network model, detections are taken place at layers 82, 94, and 106 where the cell sizes are: 13×13, 26×26, and 52×52, respectively. It is worth to note that the first detection layer detects the largest objects, the second one detects the medium-size objects, and the third one detects the smallest objects in natural scene input image. Finally, for training YOLOv3, sum of the squared error loss is used for bounding box and the binary cross-entropy loss is utilized for the objectiveness score and class probabilities.

Generally, YOLOv3 [22] is designed to detect and classify objects in natural scene images, where the objects are dominated by size. However, when dataset (e.g. DIDA) consists of objects which occupy a small area in the image, large intra-class variations, and only medium and small objects grouped close together the framework starts to provide poor detection performance. In Swedish historical handwritten documents, digits were written by different people in different handwritten styles using different types of ink and dip pen. These simply generate a dataset with many complexities as it consists of medium and small digits, large intra-class variations, touching digits, and many others. Therefore, since YOLOv3 network model cannot overcome with the aforementioned issues, it is necessary to design a new network architecture to precisely detect handwritten digits in historical document images. The proposed DIGITNET-dect, which is shown in Fig. 9, totally includes 16 convolutional layers and 7 other layers such as 1 up-sampling, 3 short-cut, 1 concatenation, and 2 detection. The details of the proposed digit detection network architecture are explained below.

Backbone The backbone of the proposed framework contains three residual units. Each residual unit has three convolution neural network structures consisting of two 3×3 and one 1×1 convolution filers. At the end of each residual block, an element-wise addition is performed between the input and output vectors. Batch Normalization is used after each convolutional layer followed

Model	Input	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Fully Connected	Fully Connected	Output
Model-1	56x56 RGB	Convolution 3x3x@1@16	Max-pooling and RELU 2x2@2@16	Convolution 5x5x@1@32	Max-pooling and RELU 2x2@2@32	Convolution 3x3x@1@64	-	128	128	10
Model-2	56x56 RGB	Convolution 5x5x@1@64	Max-pooling and RELU 2x2@2@64	Convolution 3x3x@1@64	Convolution 3x3x@3@64	Max-pooling and RELU 2x2@2@64	Convolution 3x3x@1@64	128	256	10
Model-3	56x56 RGB	Convolution 7x7x@1@64	Max-pooling and RELU 2x2@2@64	Convolution 3x3x@1@128	Max-pooling and RELU 2x2@2@128	Convolution 3x3x@1@64	-	128	512	10

Fig. 10. The CNN models for handwritten digit recognition.

by the Leaky ReLU activation function. Moreover, similar to YOLOv3 structure no pooling layer is used. Indeed, instead of pooling layer a convolutional layer with stride 2 is employed to down-sample the feature map. The down-sampling step is performed in four convolutional layers where three of them are located in the first layers of the residual units and one in separate convolutional layer.

Head subnet The head subnet of the proposed neural network architecture adopts a feature pyramid network [41] to segment digits at two different scales. This is in contrast with the original YOLOv3 network architecture [22] as it uses 3-scale detection strategy. The main reason that feature pyramid network is used is due to the fact that this strategy augments a convolutional network with a top-down pathway and lateral connections. In this manner, the network efficiently constructs a multi-scale feature pyramid from an input image. In the proposed architecture, the two levels of the pyramid can be used for segmenting digits with various sizes. More specifically, the lower resolution feature maps have larger strides that leads to a very coarse representation of the input image, which is assigned for large-to-medium-size digit detection. While the higher resolution feature maps have more fine-grained features and are used for small-size digit detection. The proposed network architecture builds feature pyramid network on the top of the backbone architecture and constructs a pyramid with down-sampling strides 16 and 8. It uses concatenation to perform the merging step in lateral connections instead of element-wise addition used in the original feature pyramid network [41]. In other words, the concatenation layer is used to take a feature map from earlier in the network and merge it with up-sampled features to get more meaningful semantic information. In the proposed architecture, instead of using a fully-connected layer, a 1×1 convolutional layer with the tensor shapes of $26 \times 26 \times (3 \times (4+1+10))$ for the lower resolution feature maps and $52 \times 52 \times (3 \times (4+1+10))$ for the higher resolution feature maps are used. As a result, at each scale, the number of filters in the last convolutional layer is set to 45.

Anchors Anchors are widely used and adopted in the state-of-the-art detection methods such as Faster R-CNN [42], single shot multibox detector [30], YOLOv2 [31] and YOLOv3 [22]. The anchor boxes are one of the most important and critical parameters in achieving high detection rate and they should be selected based on the scale and size of objects in the training data. Generally, selecting large number of anchor boxes will allow for greater overlap between prior boxes and bounding boxes. However, as the quantity of anchor boxes rises, the depth of last convolution layer before detection layers increases linearly, which results in large network size and high computational time. Therefore for a reasonable trade-off between accuracy and speed, YOLOv3 uses 9 anchor boxes (3 for each scale). However, since there are two scales in the proposed network architecture, the default number of anchor boxes is set to 6. The proposed method uses k-means clustering, where $k = 6$, on training annotations to automatically find optimum priors boxes. This is a vital task as estimating optimal anchor boxes, which are defined as box dimensions, results in high overlap between ground truth digits and prior boxes. Note that the estimated 6 anchors are sorted based on area and then distributed evenly across scales (3 for low resolution digit feature maps and 3 for higher resolution digit feature maps). In this manner, feature maps learn to be responsive to particular scales of the digits. Moreover, the proposed method only assigns one prior box for each ground truth box. Consequently, only the prior box which has the highest intersection over union with the ground truth label will be responsible for predicting a digit. This simply ensures that an anchor box predicts ground truth for a digit centered at its own cell center, and not a cell far away.

Limitation in DIGITNET-dect In the DIGITNET-dect, each obtained box predicts the classes of digits in the training set. To achieve this, there are two possible approaches which are softmax classifier and logistic classifier. In this paper, DIGITNET-dect uses independent logistic classifier, which is a multi-label approach, to better model the data. The softmax classifier is not used because it imposes the assumption that each box has exactly one class which is often not the case in our dataset as there are many overlapping digits. Even though, in our problem, multi-label classifier provides very good model for digit detection, it gives low recognition accuracy rate, especially when the correlation between feature maps and class variables becomes complex. Therefore, we propose to detect digits using DIGITNET-dect and providing each obtained bounding boxes to a DIGITNET-rec architecture model to make fine classification, which is discussed in the next subsection.

3.2. DIGITNET-rec: handwritten digit recognition network architecture

In the DIGITNET-rec, a digit classifier based on the Convolutional Neural Network (CNN) is applied to recognize the detected digits. In order to achieve high recognition accuracy, three different CNN-based handwritten digit classifiers are constructed which consist of number of layers such as convolutional, batch normalization, max-pooling, fully-connected layers and softmax layer. Moreover, the training was performed with the Stochastic Gradient Descent technique using the back-propagation with mini-batches of 256 instances and the DIDA Dataset I. Finally to recognize digit strings, the classification results of three CNNs are combined using max-voting scheme.

Each constructed classifier consists of different numbers of convolutional layers, kernel sizes, filters and strides and Fig. 10 summarizes the parameters used in all three classifiers. For instance, the model 1 as shown in Fig. 10 has 3 convolutional layers, 2 max-pooling layers and 2 fully-connected layers, and 10 output layers. In the first convolutional layer, the kernel size, stride and number of filters are 3×3, 1 and 16 ($3 \times 3@1@16$) respectively, and max pooling layer (2 × 2@2@16) is applied in the second layer. In the third layer, the convolutional layer (5 × 5@1@64) is applied which includes 64 filters with the kernel size of 5 × 5 and stride is 1. In the fourth layer, max pooling (2 × 2@2@16) is applied. In the next layer, 64 filters with the kernel size of 5 × 5 and stride is 1 are used 5 × 5@1@64. Two fully-connected layers are used, which each one has 128 nodes. Note that, ReLU is used as an activation function in the convolutional and fully-connected layers. In the last layer, Softmax is applied as a last layer to estimate the probabilities of output classes. The highest probability of the class shows the desired output. Total number of training examples present in a single batch is 256 and epoch size is 10. The other two classifiers have different numbers of convolutional and fully connected layers as shown in Fig. 10.

4. Experimental results

4.1. Handwritten digit detection methods

The proposed digit detection method is evaluated against four state-of-the-art digit detection methods and two one-step deep learning object detection methods. To detect digits in historical documents using digit detection algorithms, the methods based on Self-Organizing Maps (SOM) [11], Connected Component (COC) [12], Features of Connected Component (FEC) [13], Skeleton (SKE) [17] are implemented. In addition, the performance of the DIGITNET-dect architecture is assessed against YOLOv3-tiny and YOLOv3 [22]. The digit detection algorithms [11–13,17] are free of using any parameters. In the proposed digit detection method and compared object detection algorithms, anchor boxes are used to predict bounding boxes. Note that YOLOv3 algorithm uses nine anchors whereas the propose and YOLOv3-tiny methods employ six anchors. In YOLOv3, nine anchors are selected using k-means clustering and they are set to $(3, 7)$, $(10, 13)$, $(16, 30)$, $(33, 23)$, $(50, 71)$, $(92, 65)$, $(100, 90)$, $(110, 120)$, and $(130, 188)$. In YOLOv3-tiny the last 6 aforementioned anchors are used, whereas in the DIGITNET-dect the first 6 ones are employed. In all methods, the parameters such as learning rate, batch size, subdivision, and epoch are 0.0001 and 64, 16, and $50,000$, respectively. The models are trained and tested on a computer with a single NVIDIA GTX 1050TI GPU, an Intel i9-8950HK CPU, and 16 GB RAM.

4.2. Digit detection performance using various datasets

In the first experiment, digit detection performance is analysed using three different datasets which are extended MNIST, extended USPS and created the new dataset DIDA. Each dataset is used to train the proposed deep learning model, separately. We randomly split each dataset into training (75%), testing (15%) and validation (10%) sets. Moreover, $100,000$ handwritten digit strings which are obtained from the Swedish historical handwritten document images are used for testing.

The publicly available MNIST and USPS datasets consist of $70,000$ and $9,298$ samples, respectively. Because of the size variations of the handwritten digits on the Swedish historical document images, the images in the MNIST and USPS datasets are upsampled with various resolution sizes. Hence, the extended MNIST and USPS datasets include $240,000$ and $205,894$ digit images, respectively. In the MNIST and USPS datasets, the digits' pixels are in grayscale and the background is black. In order to fairly compare the datasets, the digits' pixels are fused into the cropped document images (416 × 416 × 3) and labelled with the same manner that the DIDA dataset III is generated.

The first experiment focuses on analysing and understanding the handwritten digit detection performance of the DIGITNET-dect which is trained on any of the aforementioned datasets. Moreover, diversities and similarities of different digit datasets are analysed and discussed for single digit detection in digit strings. The proposed detection model is separately trained on the extended MNIST dataset, extended USPS and DIDA datasets. Thus, three different trained models have been obtained using these digit datasets. Each trained model is tested on $100,000$ digit string images selected from the DIDA dataset II. The results are tabulated in Table 1. In this paper, the Correct Detection (CD) rate is defined as $CD = \frac{TP}{(TP+FP)} \times 100\%$, where TP is True Positive and FP is False Positive. To determine whether a predicted bounding box by a model is TP or FP the overlap between digit detection results and ground-truth bounding box is considered. When the IoU score is greater than 0.5, it is considered a TP, else it is considered a FP.

Table 1
Digit detection performance using different digit datasets trained on the proposed DIGITNET-dect model.

Dataset	Correct detection rate
Extended-MNIST	52.83%
Extended-USPS	39.48%
DIDA	75.96%

According to the results, the proposed model trained on the DIDA provides the highest detection rate 75.96%. The DIGITNET-dect model trained on the extended USPS gives the poorest detection performance 39.48% and the detection performance of the model trained on the extended MNIST is 52.83%. The results indicate that there are diversities between the digits on the existing datasets (MNIST and USPS) and the DIDA. More specifically, these low detection rates simply mean that the samples in the DIDA dataset contain different digit patterns than the other digit datasets, and hence, the proposed DIGITNET-dect trained by other datasets cannot detect the digits very well on the historical handwritten numeral string images.

4.3. Comparison of digit detection methods on Swedish handwritten document images

First, we examine the performance of the proposed handwritten digit detection (DIGITNET-dect) method to detect the handwritten digits on the numeral string images. To analyze the results, the proposed DIGITNET-dect method is compared with the SOM-based digit detection method [11], COC-based digit detection method [12], FEC-based digit detection method [13], SKE-based digit detection method [17], YOLOv3-tiny and YOLOv3 [22]. Note that, the proposed detection model, YOLOv3 [22] and YOLOv3-tiny are trained on the DIDA dataset III. Moreover, test images are selected from the DIDA dataset II where $4,200$ numeral string images have been used. The implemented algorithms have been applied to detect digits on six different types of numeral string issues which are **Type I**: Single-Touch (467 test images), **Type II**: Multi-Touch (185 test images), **Type III**: Broken (256 test images), **Type IV**: Degradation (456 test images), **Type V**: Faint or invisibility (286 test images) and **Type VI**: No artefacts (2550 test images). Table 2 shows the comparison results of correct detection rates of the implemented algorithms for different numeral string issues. According to results, the proposed DIGITNET-dect method provides the best correct detection rates in all types of issues and the overall correct detection rate is estimated at 76.84% and YOLOv3-tiny gives the second best performance with 69.56%. On the other hand, YOLOv3 gives the lowest detection rate with 45.21% and performs poorest for three types of issues which are the Types I, II, and VI. The main reason that YOLOv3 provides the least accurate results is due to the fact that the network depth is very high for digit

Table 2
Correct detection rates of the algorithms for different numeral string issues.

Method	Issues						
	Type I	Type II	Type III	Type IV	Type V	Type VI	Total
Images	467	185	256	456	286	2550	Detec. rate
SKE [17]	356	104	86	166	73	2346	53.33%
COC [12]	368	106	124	178	104	2402	59.02%
FEC [13]	374	116	139	195	128	2386	63.03%
SOM [11]	345	98	88	206	97	2304	55.11%
YOLOv3 [22]	268	52	98	175	78	2086	45.21%
tiny-YOLO	366	94	168	297	182	2392	69.56%
DIGITNET-dect	384	122	184	328	208	2456	76.84%

detection problem so that accuracy gets saturated due to high training error. Moreover, the proposed detection framework provides higher performance than the YOLOv3-tiny as the DIGITNET-dect downsamples the input images by factors of 16 and 8 but, the YOLOv3-tiny downsamples the input images by factors of 32 and 16 which leads to relatively course features for digit differentiation. In other words, YOLOv3-tiny due to this downsampling strategy poses a problem for small-scale digits. Furthermore, other handwritten digit detection methods which are FEC [13], COC [12], SOM [11], and SKE [17] provide the third, fourth, fifth and sixth best digit detection performance, respectively. Based on the results in different cases, these methods are successful mostly for single-touch and no artefact cases because they are designed to resolve one or two type digit detection issues on the numeral string images. These results also clarify that the methods have difficulties to detect the digits when the challenges become more complex. Clearly, the DIGITNET-dect gives the highest detection accuracy compared to the other methods.

Fig. 11 depicts some numeral strings that are correctly detected using our proposed approach and it also shows that our algorithm can detect digits in various cases such as single- or multiple-touching numeral strings, faint and broken. On the other hand, state-of-the-art methods cannot overcome all detection issues. For instance, the method proposed by Gattal et al. [12] can partially detect the digit 8 as the digit deal with invisibility issue. Therefore, this algorithm cannot separate the digit fully. Moreover, the SKE-based digit detection method [17] and YOLOv3-tiny cannot detect the digit 8. Additionally, the results on other three images which include connected numeral strings show that the compared methods cannot separate the digits with high accuracy and provide high miss detection rates, but the proposed approach can detect the connected digits with higher detection rate. Furthermore, more numeral string images are tested to detect the digits using the proposed method and the results are illustrated in Fig. 12. Based on the results, the proposed detection approach DIGITNET-dect performs well to separate digits from the numeral string images with various handwritten detection issues.

In the second experiment, all the handwritten digit detection methods have been tested on 200, 000 digit string images and the results are tabulated in Table 3. Note that, the proposed DIGITNET-dect model, YOLOv3 [22] and YOLOv3-tiny are trained on DIDA dataset III. The results show that the highest handwritten digit detection rate is achieved using the proposed digit detection architecture with 70.15% and the lowest rate is obtained using the YOLOv3 [22] with 48.56%. The second-highest performance belongs to the YOLOv3-tiny method with 64.48% detection accuracy rate. The third- and fourth-best detection performance belong to Gattal et al. [12] and Chen et al. [17] with detection accuracy rate of 60.23% and 58.18%, respectively. The main issues with the other methods are that 1) they mostly focus on one type or two-types of digit detection problem and 2) they use non-effective approaches such as combination of image procession techniques. Moreover, efficient deep learning methods such as YOLOv3 and YOLOv3-tiny have also been trained on the DIDA dataset III and the results depict that YOLOv3-tiny performs better than the YOLOv3. One reason can be because of the fact that YOLOv3 architecture has 75 convolutional layers, which causes the network to overfit to the training set. Consequently, the overall results show that the proposed DIGITNET-dect model outperforms the compared state-of-the-art methods.

4.4. Comparison of digit recognition methods

This experiment aims at understanding and analyzing the effectiveness and robustness of the proposed DIGITNET-rec model which is compared with different machine learning classifiers. In this experiment, DIDA dataset I has been used for training, which consists of 250, 000 single digit samples and the detected digits which are obtained using three best digit detection methods have been used for testing. Table 4 illustrates the recognition accuracy of the trained classifiers. The results show that the classifiers achieve high recognition rates for the digits detected using the DIGITNET-dect. According to the results, the highest and lowest recognition results are achieved using the DIGITNET-rec and kNN classifier with 97.12% and 80.10%, respectively. Moreover, HOG-SVM classifier obtains the second-best recognition rate with 96.05% since Support Vector Machine (SVM) applies with the Histogram Oriented Gradient (HOG) feature extraction technique. Other methods which are Random Forest, Recurrent Neural Networks (RNN) and SVM apply on the raw pixels. Amongst them, RNN performs slightly better than the other methods and it finds 95.88%. These results verify that DIGITNET-dect detects the locations of the boundaries of the digits on numeral strings more accurately than the YOLOv3-tiny and the FEC which increases the performance of the classifiers. Moreover, the recognition performance of the classifiers with the YOLOv3-tiny digit detection algorithm is worse than the FEC digit detection algorithm even though YOLOv3-tiny provides better detection results than the FEC as discussed in the previous subsection. The main reason is that the FEC digit detection algorithm detects the boundary of the digits on the numeral strings which raises the performance of the classifiers.

Table 5 shows the impact of using ensemble model as compared to the three individual deep learning classifier models in DIGITNET-rec. The results indicate that using DIGITNET-rec with ensemble model provides the best recognition accuracy.

5. Conclusion

In this paper, a large historical handwritten digit dataset named DIDA is introduced (publicly available from: https://didadataset.github.io/DIDA/). The dataset has been collected from the historical Swedish handwritten document images written between the year 1800 and 1940 and contains: 1) single digit images with original appearance, 2) multi-digit images in RGB color space and 3) image

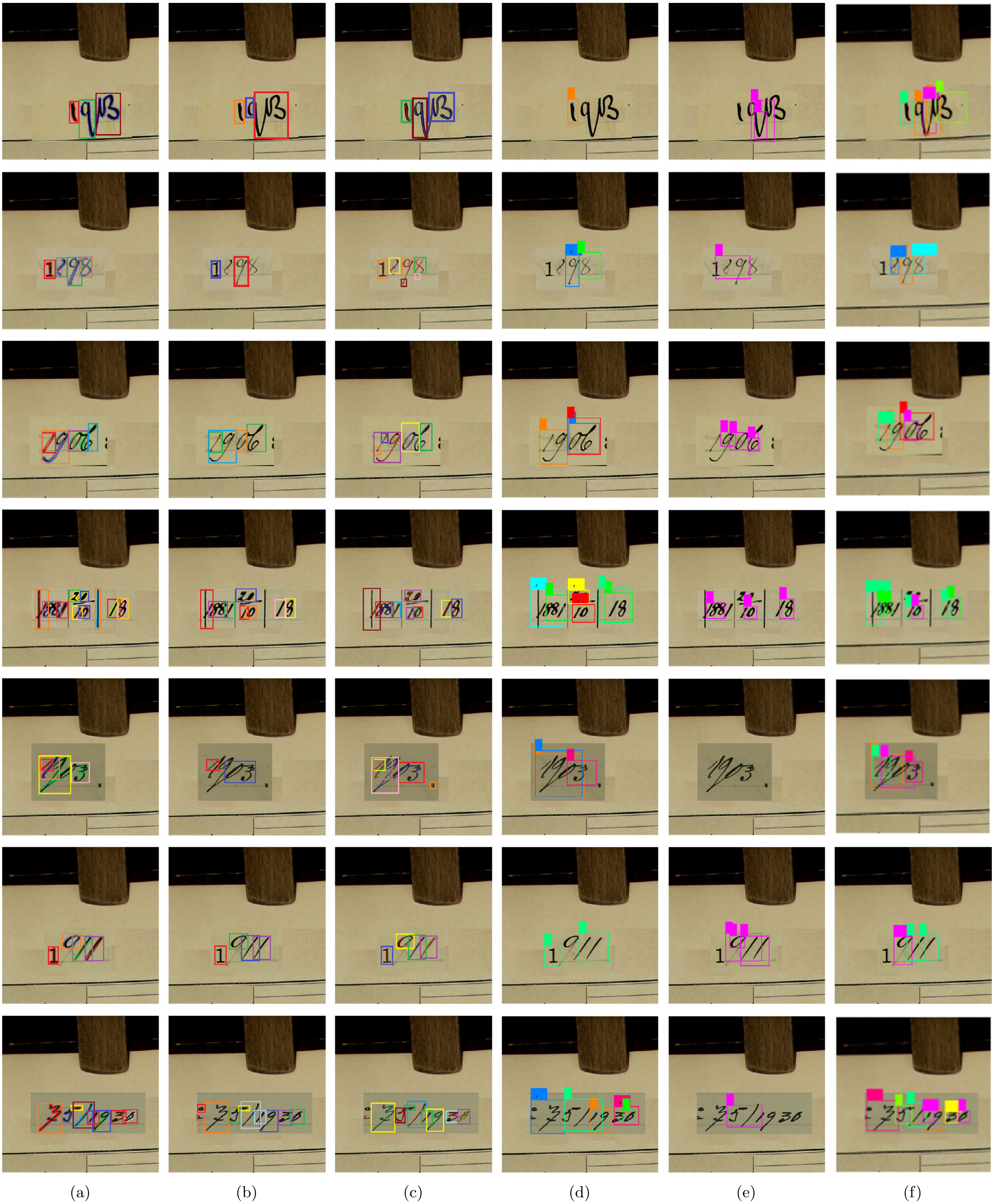

Fig. 11. Qualitative comparison of six methods on several multi-digit images from the DIDA dataset II. (a) Connected component-based digit detection method [12]; (b) Self-Organizing Maps based digit detection method [11]; (c) Skeleton-based detection method [17]; (d) YOLOv3-tiny; (e) YOLOv3 [22]; and (f) DIGITNET-dect.

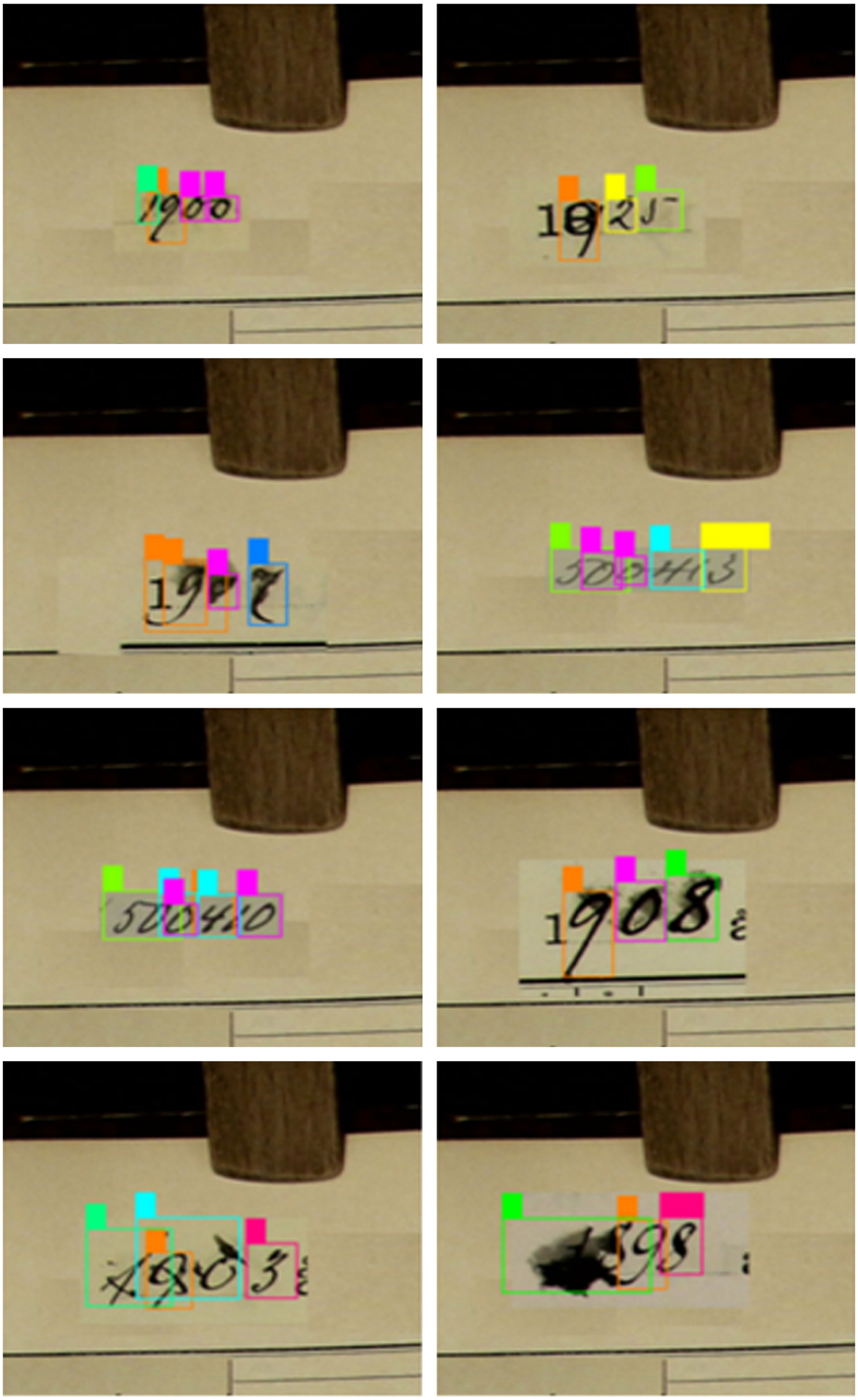

Fig. 12. Detection results using the proposed DIGITNET-dect model for different digit string images.

dataset for deep learning object detection algorithms. Besides of the dataset, a new deep learning framework for historical handwritten digit detection and recognition called DIGITNET is presented which consists of two steps which are DIGITNET-dect and DIGITNET-rec. The DIGITNET-dect is trained on the DIDA dataset III, MNIST, and USPS handwritten digit datasets and tested on handwritten digit string images in DIDA dataset II. The results show that the model trained on the DIDA dataset provides the highest detection rate. Moreover, the experimental results verify effectiveness and efficiency of the DIGITNET-dect, which outperforms the state-of-the-art digit detection methods as well as deep learning methods. In the DIGITNET-rec, three different convolutional neural networks (CNN) architectures are constructed for recognition of the detected digits and then majority voting is used to recognize detected digits. Consequently, the proposed DIGITNET model achieves the highest handwritten digit detection and digit string recognition rates in the historical Swedish handwritten document images.

Table 3
Handwritten digit detection performance using different detection methods on 200, 000 handwritten digit string images.

Method	Correct detection rate
SKE [17]	58.18%
COC [12]	60.23%
FEC [13]	52.26%
SOM [11]	55.24%
YOLOv3 [22]	48.56%
YOLOv3-tiny	64.48%
DIGITNET-dect	70.15%

Table 4
Recognition accuracy of the machine learning methods with the handwritten digit detection methods which are YOLOv3-tiny, FEC [12] and DIGITNET-dect.

Method	Recognition accuracy		
	YOLOv3-tiny	FEC [13]	DIGITNET-dect
DIGITNET-rec	88.59 %	96.13 %	97.12%
SVM	80.68 %	92.01 %	94.56%
HOG-SVM	83.42 %	94.14 %	96.05%
kNN	80.10 %	88.55 %	94.04%
Random Forest	82.36 %	90.33 %	95.34%
RNN	81.76 %	92.56 %	95.88%

Table 5
The impact of using ensemble model in DIGITNET-rec as compared to three individual deep learning classifier models.

Method	Recognition accuracy
DIGITNET-rec with Ensemble	97.12%
DIGITNET-rec with Model-1	95.47%
DIGITNET-rec with Model-2	94.18%
DIGITNET-rec with Model-3	94.76%

This work focuses on handwritten digit string recognition in Swedish historical document images using a deep learning framework, however to automatically analyze whole document images, it is important to extend the current work with: 1) creating a new dataset which would include Swedish characters and words and 2) developing a new deep learning framework to be able to segment and recognize characters and words.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Conflict of interest

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

Acknowledgement

This study was conducted within the research project scalable resource efficient systems for big data analytics funded by the Knowledge Foundation (Grant: 20140032) in Sweden.

References

[1] E. Granell, E. Chammas, L.L. Sulem, C.D.M. Hinarejos, C. Mokbel, B.I. Cirstea, Transcription of Spanish historical handwritten documents with deep neural networks, J. Imaging 4 (15) (2018) 1–22.
[2] M. Revow, C.K.I. Williams, G.E. Hinton, Using generative models for handwritten digit recognition, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 592–606.
[3] O. Elitez, Handwritten digit string segmentation and recognition using deep learning, Master's thesis, Middle East Technical University, Turkey, 2015.
[4] F.C. Ribas, L.S. Oliveira, A.S. Britto Jr, R. Sabourin, Handwritten digit segmentation: a comparative study, Int. J. Doc. Anal. Recognit. 16 (2013) 127–137.
[5] Z. Shi, DateFinder: detecting date regions on handwritten document images based on positional expectancy, Master's thesis, University of Groningen, The Netherlands, 2016.
[6] R. Saabni, Recognizing handwritten single digits and digit strings using deep architecture of neural networks, in: International Conference on Artificial Intelligence and Pattern Recognition, IEEE, 2016, pp. 1–6.
[7] S. Choi, I. Oh, A segmentation-free recognition of two touching numerals using neural networks, in: International Conference on Document Analysis and Recognition, IEEE, 1999, pp. 253–256.
[8] D. Ciresan, Avoiding segmentation in multi-digit numeral string recognition by combining single and two-digit classifiers trained without negative examples, in: International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE, 2008, pp. 225–230.
[9] K.K. Kim, J.H. Kim, C.Y. Suen, Segmentation-based recognition of handwritten touching pairs of digits using structural features, Pattern Recognit. Lett. 23 (1) (2002) 13–24.
[10] P.J. Grother, NIST special database 19, 2017, https://www.nist.gov/srd/nist-special-database-19.
[11] E.B. Lacerda, C.A.B. Mello, Segmentation of connected handwritten digits using self-organizing maps, Expert Syst. Appl. 40 (15) (2013) 5867–5877.
[12] A. Gattal, Y. Chibani, B. Hadjadji, Segmentation and recognition system for unknown-length handwritten digit strings, Pattern Anal. Appl. 20 (2017) 307–323.
[13] H. Merabti, B. Farou, H. Seridi, A segmentation-recognition approach with a fuzzy-artificial immune system for unconstrained handwritten connected digits, Informatica 42 (1) (2018) 95–106.
[14] A.G. Hochuli, L.S. Oliveira, A.S. Britto Jr, R. Sabourin, Handwritten digit segmentation: is it still necessary?, Pattern Recognit. 78 (2018) 1–11.
[15] O. Matan, J.C. Burges, Y. LeCun, J.S. Denker, Multi-digit recognition using a space displacement neural network, in: Advances in Neural Information Processing Systems, IEEE, 1992, pp. 488–495.
[16] Z. Shi, V. Govindaraju, Segmentation and recognition of connected handwritten numeral strings, Pattern Recognit. 30 (9) (1997) 1501–1504.
[17] Y.K. Chen, J.F. Wang, Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1304–1317.
[18] T. Saba, A. Rehman, M. Elarbi-Boudihir, Methods and strategies on off-line cursive touched characters segmentation: a directional review, Artif. Intell. Rev. 42 (2014) 1047–1066.
[19] D. Yu, H. Yan, Separation of touching handwritten multi-numeral strings based on morphological structural features, Pattern Recognit. 34 (3) (2001) 587–598.
[20] B.E. Kessab, C. Daoui, B. Bouikhalene, R. Salouan, A comparative study between the k-nearest neighbours and the multi-layer perceptron for cursive handwritten Arabic numerals recognition, Int. J. Comput. Appl. 107 (21) (2014) 25–30.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.
[22] J. Redmon, A. Farhadi, YOLOv3: an incremental improvement, arXiv:1804.02767, 2018.
[23] Y. LeCun, C. Cortes, C.J. Burges, MNIST handwritten digit database, http://yann.lecun.com/exdb/mnist, 2010.
[24] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.
[25] H. Sajedi, Handwriting recognition of digits, signs, and numerical strings in Persian, Comput. Electr. Eng. 49 (2016) 52–65.
[26] H. Kusetogullari, A. Yavariabdi, A. Cheddad, H. Grahn, J. Hall, ARDIS: a Swedish historical handwritten digit dataset, Neural Comput. Appl. (2019) 1–14.
[27] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2014.
[28] D. Yoo, S. Park, J.-Y. Lee, A.S. Paek, I.S. Kweon, AttentionNet: aggregating weak directions for accurate object detection, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2015.
[29] M. Najibi, M. Rastegari, L.S. Davis, G-CNN: an iterative grid based object detector, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2016.
[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Conference on European Conference on Computer Vision, 2016.

[31] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, arXiv:1612.08242, 2016.

[32] C.Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: deconvolutional single shot detector, arXiv:1701.06659, 2017.

[33] Z. Shen, Z. Liu, J. Li, Y.G. Jiang, Y. Chen, X. Xue, DSOD: learning deeply supervised object detectors from scratch, in: International Conference on Computer Vision, IEEE, 2017.

[34] Z. Zhao, P. Zheng, S. Xu, X. Wu, Object detection with deep learning: a review, IEEE Trans. Neural Netw. Learn. Syst. (2017) 1–21.

[35] S.S. Ramachandran, J. George, S. Skaria, V.V. Varun, Using YOLO based deep learning network for real time detection and localization of lung nodules from low dose CT scans, in: Medical Imaging: Computer-Aided Diagnosis, SPIE, 2018, pp. 225–230.

[36] S. Hossain, D. Lee, Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices, Sensors 19 (15) (2019) 1–24.

[37] H. Saribas, B. Uzun, B. Benligiray, O. Eker, H. Cevikalp, A hybrid method for tracking of objects by UAVs, in: Computer Vision and Pattern Recognition Workshops, IEEE, 2019.

[38] W. Fang, L. Wang, P. Ren, Tinier-YOLO a real-time object detection method for constrained environments, IEEE Access 8 (2020) 1935–1944.

[39] H. Song, H. Liang, H. Li, Z. Dai, X. Yun, Vision-based vehicle detection and counting system using deep learning in highway scenes, Eur. Transp. Res. Rev. 11 (51) (2019) 1–16.

[40] J. Chen, Z. Liu, H. Wang, A. Nunez, Z. Han, Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network, IEEE Trans. Instrum. Meas. 67 (2) (2018) 257–269.

[41] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Computer Vision and Pattern Recognition, IEEE, 2016.

[42] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Computer Vision and Pattern Recognition Workshops, IEEE, 2016.