Writer Identification Using Deep Neural Networks: Impact of Patch Size and Number of Patches

Akshay Punjabi, Jose Ramón Prieto, Enrique Vidal
PRHLT Research Center
Universitat Politècnica de València, Spain
email: joprfon@prhlt.upv.es

Abstract—**Traditional approaches for the recognition or identification of the writer of a handwritten text image used to relay on heuristic knowledge about the shape and other features of the strokes of previously segmented characters. However, recent works have done significantly advances on the state of the art thanks to the use of various types of deep neural networks. In most of all of these works, text images are decomposed into patches, which are processed by the networks without any previous character or word segmentation. In this paper, we study how the way images are decomposed into patches impact recognition accuracy, using three publicly available datasets. The study also includes a simpler architecture where no patches are used at all – a single deep neural network inputs a whole text image and directly provides a writer recognition hypothesis. Results show that bigger patches generally lead to improved accuracy, achieving in one of the datasets a significant improvement over the best results reported so far.**

Keywords: writer identification, deep learning, neural networks, CNN, convolutional neural network, handwritten pages, document image processing, historical document writer identification, pattern recognition

I. Introduction

The identification or recognition of the writer who has penned a piece of text is needed in several applications of document analysis and recognition. In particular, it is required for the organization of historical manuscripts in archives and libraries. Traditional methods used to rely on a segmentation of the written text into individual characters, followed by more or less heuristic analysis of the shape of the characters [1], [2]. This approach has provided some results in simple cases where the text is clear enough to allow the detection and extraction of individual characters. However, it fails dramatically when the writing is sloppy and/or when ancient scripts are considered.

Therefore, modern works follow holistic approaches where the writing style is analyzed without character or word detection or extraction. Moreover, machine learning techniques are proving highly adequate to tackle the writer's identification problem in a fully holistic, segmentation-free way. In this direction, deep convolutional neural networks (DCNN) are the most promising models tried so far [3], [4], [5], [6].

In this paper, we explore in detail the use of DCNN for the classification of a handwritten image into several classes corresponding to possible hands which have written the text in the image. After informally exploring several DCNN architectures, the present work focuses on the so-called ResNet model [7] and explores how the way images are decomposed into patches impacts the recognition accuracy. We also study a simple architecture where no patches are used at all and a single DCNN input a whole text image and directly provides a writer recognition hypothesis.

We report results of extensive experiments with three public-domain datasets: IAM [8], Firemaker [9] and ICDAR17 [10], the dataset used in the ICDAR 2017 writer identification competition. Both IAM and Firemaker are well-known benchmark datasets consisting of modern documents which allow us to compare our results with the state of the art reasonably precisely. ICDAR17, on the other hand, is particularly interesting because it consists of a large number of images of *historical* documents. However, it was originally proposed for writer identification experiments, which a training/test partition that ensures no writer of the training set has any page image in the test set. However, this dataset can be very naturally partitioned also for experimentation on writer recognition. So we did and adopted this dataset as a good representative of the writer recognition challenges entailed by historical handwritten text.

The results of our experiments are generally satisfactory and, for the two benchmark datasets, they are close to, or better than the best results achieved so far.

II. Related works

This section provides an exhaustive analysis of the state of the art on writer identification. Our analysis is mainly based on [11] which does a review of methods between 2011 and 2016. The methods depicted in [11] follow traditional handcrafted approaches. In this paper, we review these handcrafted approaches and extend it with deep learning approaches. All results are listed on Table I.

Traditional handcrafted methods are categorized into structure-based methods, texture-based and grapheme-based.

A. Texture-based methods

One of the first texture-based methods was of Said et al. [12] where the texture feature was extracted by Gabor filters and grey level co-occurrence matrix (GLCM). He et al. [13] used Hidden Markov Tree (HMT) model in the wavelet domain for feature extraction. Helli and Moghaddam [14] used Gabor and XGabor filter for feature extraction. Bertolini et al. [15] create a texture representation image by compressing handwriting utilizing overlapping individual connected components. After that, Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) are used for feature extraction, and classification is done by a Support Vector Machine (SVM).

In [6] the authors propose the identification of different writers by making use of novel direction, curvature, and tortuosity based geometrical features. Furthermore, the paper proposed the improvement of state of the art edge-based directional features by using a filled moving window instead of edge moving window and chain code-based features by using a fourth-order chain-code list to improve discriminative power.

B. Structure-based methods

There is one interesting study [16] in which they used the width of ink traces for writer identification. They created the Quill-Hinge feature, which is a probability distribution of the relation between the ink direction and the ink width.

In [1] the authors extract SIFT descriptors (SDs) and the respective scale and orientations (SOs) from segmented words. From the SDs and SOs, they obtain signatures and histograms respectively, which then are used to distinguish the style between writers. The same authors also propose [2], where they modify descriptors, SDs and triangular descriptor (TD), to incorporate orientation information and named them as modified SD or MSD. Similar to before, they create histograms from the descriptors, in this case, MSDs histograms (MSDH) and TD Histogram (TDH), to identify the writers. Experimental results show that this method is more effective than the previously mentioned with the unmodified SIFT descriptor.

C. Grapheme-based methods

Jain and Doermann [17] proposes the generation of features for the codebook with K-Adjacent Segment (KAS). The KAS method has better performance as a single feature descriptor than other methods where many features are combined to achieve better results.

Kirli et al. [18] proposes a novel dynamic windows based feature extraction model. This method can adapt to any kind of handwriting. They use the dynamic windows to extract features from three special writing zones. As for classification for the writer identification, they use k-nearest neighbor (K-NN), Gaussian Mixture Models (GMM), and Normal Density Discriminant Function (NDDF) Bayes classifiers.

Although graphemes features are typically used for the codebook generation, Fiel and Sablatnig [19] extracted SIFT descriptors. After that, the histogram of occurrences of a new document is then compared to the ones in the codebook. Finally, classification is done with the nearest neighbor method. Using local SIFT descriptors has the added advantage of not needing binarization preprocessing, avoiding information loss.

Ghiasi and Safabakhsh [20] introduce two new efficient methods for the generation of codebooks from contours. The first method uses the actual pixel coordinates of contour fragments while the other method computes the linear piecewise approximation using segment angles and lengths. For feature extraction, they use the occurrence histogram of the shapes in the codebook.

Khalifa et al. [21] apply an ensemble of codebooks complemented with kernel discriminant analysis using spectral regression (SR-KDA). SR-KDA is chosen as a way to reduce dimensionality and avoid over-fitting problems that arise with combining multiple codebooks. The method improved classification accuracy using many distinct codebooks from randomly generated grapheme features. The ensemble of codebooks has shown an 11% increase compared to the use of a single codebook.

Garz et al. [22] approach uses a set of novel descriptors extracted from geometrical interest points at various scales like from strokes, junction, endings, and loops. The proposed descriptors reduce the compute time compared to other methods and are more straightforward and efficient to use.

More recently, Khan et al. [23], used a combination of SIFT and RootSIFT descriptors in a set of Gaussian mixture models (GMMs). With their method, they have obtained the current best results in the Firemaker [24] dataset.

D. Combination of structure and grapheme based methods

Some researchers have developed methods that work both at the texture level and the allograph level. Bulacu and Schomaker [25] use, at the texture level, contour-based joint directional probability distribution functions (PDFs) that encode orientation and curvature information of the writing style. At the allograph level, they use a random pattern generator of ink-trace fragments. The base codebook is generated by segmenting the allograph into several fragments. Finally, the writer is identified with the PDF of the patterns of the writing. The authors have also proposed several new features, such as edge direction distribution, edge-hinge distribution, and directional co-occurrence. Siddiqi and Vincent [26] also extracted features at the allograph level and texture level. At the allograph level, they create a codebook of the most similar shapes used. At the texture level, they extract features from the chain code of the handwriting contours. Finally, KNN is used for classification.

E. Deep Learning based methods

Deep learning methods have been gaining significant momentum in all types of fields, including writer identification. One of the first attempts at writer identification with deep learning methods was proposed by Fiel and Sablatnig [3] in 2015. The authors proposed to train a CNN over the input images, which consisted of segmented words and line images, and used the second last fully connected layer as a feature vector after training is finished. The obtained feature vector is then compared to precalculated feature vectors of the dataset by nearest neighbor classification. In the same year, [4] used a very similar approach differing in a few respects. They also used the second to last layer of a CNN as a feature vector. However, in their case, the feature vector that formed the second last layer was encoded to form a global feature vector through Gaussian mixture models (GMM) super vector encoding. They obtained better results than the model proposed by Fiel and Sablatnig.

TABLE I
SUMMARY OF STATE OF THE ART IN WRITER IDENTIFICATION METHODS ON IAM [8] AND FIREMAKER [24] DATASETS WITH RESPECT TO THE TYPE OF APPROACH. IN BOLD BEST RESULTS IN THE RESPECTIVE DATASET.

STRUCTURE-BASED APPROACHES

Year	Features	Classifier	Ref.	Dataset	Writers	Top1 (%)
2014	MSDH + TDH	KNN	[2]	IAM	657	97.1
2014	SDS + SOH	Euclidean	[1]	IAM	657	**98.5**
				Firemaker	250	92.4
2012	Quill–Hinge	NN	[16]	IAM	657	97
				Firemaker	251	86

TEXTURE-BASED APPROACHES

Year	Feature	Classifier	Ref.	DB	Writers	Top1
2016	Chain code	KDA	[6]	IAM	650	82.7
2013	Texture LPQ	SVM	[15]	IAM	650	96.7

GRAPHEME-BASED APPROACHES

Year	Feature	Classifier	Ref.	DB	Writers	Top1
2019	SIFT + RootSIFT	GMM	[23]	IAM	650	97.85
				Firemaker	250	**97.98**
2016	p(Is,Iθ), p(IBOS)		[22]	IAM	657	86.9
2015	Graphemes	SR-KDA	[21]	IAM	657	92
2013	Connected	KNN, x2	[20]	IAM	650	94.8
				Firemaker	250	95.2
2012	SIFT	x2	[19]	IAM	650	93.1
2011	KAS	SVM	[17]	IAM	650	92.1
2011	Global and local	KNN, GMM, Bayes	[18]	IAM	93	98.76

COMBINATION OF STRUCTURE AND GRAPHEME BASED METHODS

Year	Feature	Classifier	Ref	DB	Writers	Top1
2010	Codebook and contour	KNN	[26]	IAM	650	91
2007	Contour PDFs and ink trace	PDFs	[25]	IAM	650	89
				Firemaker	250	83

DEEP LEARNING-BASED APPROACHES

Year	Feature	Classifier	Ref.	DB	Writers	Top1
2020	CNN with word fragments		[31]	IAM	657	96.3
				Firemaker	250	97.6
2019	CNN with tuples of images		[30]	IAM	650	93.14
				Firemaker	250	93.56
2016	Multi-stream CNN		[29]	IAM	657	97.3

Another interesting work is of [5] where, instead of extracting local features like previous work, they extract global features of the entire image. As the other research mentioned before, they also extract features with a CNN but use a joint bayesian technique to extract the writer. The results reported in this paper are the best results for the ICDAR13 [27] dataset and the CVL [28] dataset.

One successful approach that involves classification and not feature extraction was of DeepWriter [29] a deep multi-stream convolutional neural network consisting of two branches sharing the convolutional layers. Nguyen et al. [30] propose another end to end CNN classifier with two branches. A subregion level branch targets individual writing strokes features. A character-level branch captures character shape features. Finally, the extracted local features are aggregated into global features and sent to a softmax classifier to make a prediction.

A very recent study [31] suggests a novel DNN called FragNet that uses word images. FragNet has two branches: a feature pyramid branch and a fragment branch. The feature pyramid branch is a traditional CNN used to extract feature maps in different scales from an input word image. The fragment branch receives fragments that are segmented from the input word image and feature maps on the feature pyramid. The combined pieces of evidence of all fragments are taken into account to make the final prediction of the writer authorship. One of the major advantages of FragNet is that it can be interpreted.

III. PROPOSED APPROACH

As we have seen in the previous section there have been few deep learning works in the field of writer identification. Interestingly, all of the approaches used as an input training sample for the neural network, a small image patch. To the best of our knowledge, none of the works uses full-page neither big patches of images as training input. As a result, we propose a model that works with full-page images as well as comparisons with versions of the model working with image patches.

We use the well-known ResNet [7] neural network architecture, in specific the ResNet18 version. Some preliminary tests were done on other popular models like VGG[32], AlexNet [33], and DenseNet[34]. Similarly, other larger versions of ResNet have also been tested. Since the results are similar to the smaller version, this one has been chosen.

The ResNet model used in all of the experiments is a pretrained version using ImageNet. Thus, fine-tuning is done by changing only the last fully connected layer to fit with our number of classes, and training it from the beginning.

Additionally, we fine-tune two variants of our models based on the input image. The first variant uses a full page as input in which we do a special preprocess explained in Section IV-D. The second variant uses patch images. In this variant, different sizes are tested, ranging from 100×100 to 1500×1500, and are shown in Section V-A.

Concerning our models that use patch images as input, as training input we extract randomly from a page image, n number of square patches with size $a \times a$, where n and a are hyperparameters of the model. Random extraction of patches implies that any patch could be extracted from all the possibilities, meaning that there could be heavily overlapped patches.

As testing and validation input, we extract the same sized patch $a\times a$ but with stride $a\times a$ so that no overlapping patches are extracted and all the image information is received as input. We could have followed the same procedure as in the training phase, extracting random patches, but we believe obtaining all possible non-overlapping patches is a more consistent and robust manner to tackle the testing phase since it removes the random factor.

Finally, in the patch model, a voting scheme is used to fusion all the patches results. The maximum predicted writer for each patch is considered as a vote. As a result, the writer who receives the most votes is considered as the predicted writer. A general view of the scheme can be seen in Figure 1.

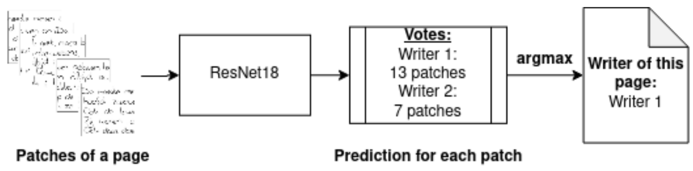


Fig. 1. Visual illustration of the voting scheme used for our patch models.

IV. Datasets

The datasets used in this work are described below. We followed the experimental protocol used in literature like in [25], [1] for IAM and [30] for Firemaker. For ICDAR17 we propose a new benchmark partition. We also explain how these datasets are adapted for the experiments presented in Sec. V. Table II summarizes the features of each dataset and the partitions adopted in the work reported in this paper.

TABLE II
Datasets and partitions

Dataset	Writers	Training pages	Test pages	Total pages
Firemaker	250	1	1	500
IAM	657	1 or 1/2	1 or 1/2	1314
ICDAR17	720	4	1	3600

A. IAM Handwriting Database

The IAM Handwriting Database was first published at the ICDAR 1999 and is described in [8][1]. It contains 1539 pages of handwritten text from 657 writers, each providing a different amount of pages. The dataset was rather artificially produced during the late 1990s. Writers were asked to copy fragments of English printed text into sheets of paper, using any kind of pen of their choice.

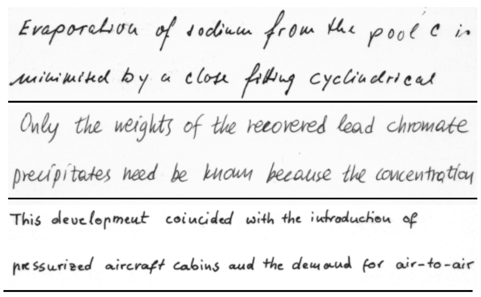
It is worth noting that this dataset was produced for experimentation on multi-writer and/or writer-independent handwritten text recognition. However, since the collection contains page images of many writers, several authors have adapted IAM for its use as a benchmark dataset in writer recognition experiments. Here we follow the setting adopted in [25], [1]. Since the number of pages of each writer ranges from 1 to 58 pages, the original dataset is modified to have exactly two pages for each writer. For writers who contributed more than two pages, only the first two are kept, and for writers with only one page, the respective pages are cut in half.

Examples of handwritten text from different writers of the IAM dataset are shown in Fig. 2.

B. Firemaker

This dataset [24][2] contains text written in Dutch by 250 writers, each providing four handwritten pages. As with IAM, this dataset was rather artificially produced. Each page of each writer has a different writing condition. The first page consists

[1]http://www.fki.inf.unibe.ch/databases/iam-handwriting-database
[2]https://zenodo.org/record/1194612#.XqHAb8szbmE


Evaporation of sodium from the pool c is
minimized by a close fitting cylindrical

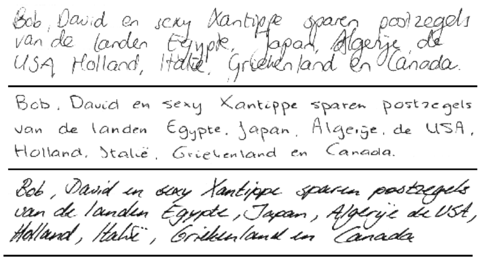
Only the weights of the recovered lead chromate
precipitates need be known because the concentration

This development coincided with the introduction of
pressurized aircraft cabins and the demand for air-to-air

Fig. 2. Three IAM examples of text from different writers.

of a copy of a specific text so that each of the 250 writers are copying the same text. On the second page, writers use the same text mentioned before but written in upper case. The third page again contains the text copied before, but writers were given the condition of writing in a different style than their natural style. The fourth and final page is a description of a cartoon comic image; therefore, each writer provides a different text. As in [30], we use the first page of each writer for training and validation, the fourth page for testing, and the other two pages are discarded.

Examples of handwritten text from different writers of the Firemaker dataset are shown in Fig. 3.


Bob, David en sexy Xantippe sparen postzegels
van de landen Egypte, Japan, Algerije, de
USA, Holland, Italië, Griekenland en Canada.

Bob, David en sexy Xantippe sparen postzegels
van de landen Egypte, Japan, Algeije, de USA,
Holland, Italië, Griekenland en Canada.

Bob, David en sexy Xantippe sparen postzegels
van de landen Egypte, Japan, Algerije de USA,
Holland, Italië, Griekenland en Canada

Fig. 3. Three Firemaker examples of text from different writers.

C. ICDAR17

This dataset was proposed for the 2017 ICDAR Competition on Historical Document Writer Identification in the International Conference on Document Analysis and Recognition [10][3]. The objective of the competition was to retrieve the documents written by the same writer of a given document, considered as a "query by example". Documents consist of historical handwritten pages from the 13th to 20th century. The dataset encompasses a training set of 394 writers which provide three pages each and a test set of 720 writers which provide five pages with each. In contrast with IAM and Firemaker, this dataset is a good representation of real challenges entailed by writer recognition for historical manuscripts.

The writers of these two sets are disjoint; so training data is only potentially useful for learning to extract relevant features from images which allow distinguishing among writers (rather than other features of the text images) – which can be seen as a case of unsupervised feature learning.

[3]https://zenodo.org/record/1324999#.XwsCchFw1H5

Therefore the official training/test partition provided for the ICDAR 2017 competition is not adequate in the writer recognition scenario adopted in the present work. In this scenario, the objective is to classify each page image in several known classes, each associated with a different writer. So, as with IAM and Firemaker, here the system should be supervisedly trained on training images written by the writers which are expected to be seen in the test images.

For these reasons, in the present work, we propose a different benchmark partition, similar to those of IAM and Firemaker. Specifically, we use only the 3600 page images of the official ICDAR 2017 test set, written by 720 writers. For each writer, the first three pages are used for training, the next one for validation and the last one for testing.

Examples of handwritten text from different writers of ICDAR17 are shown in Fig. 4.

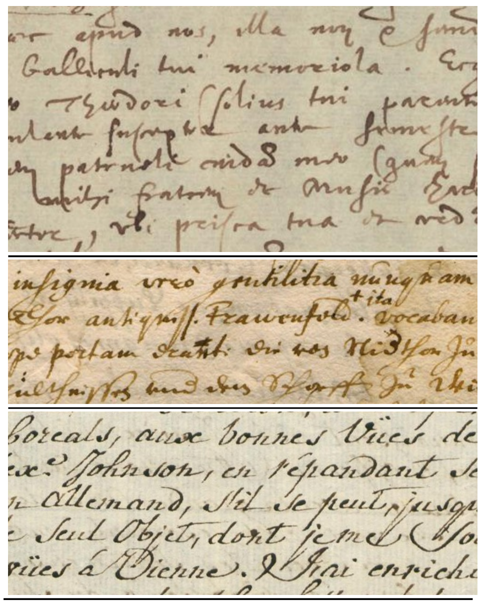
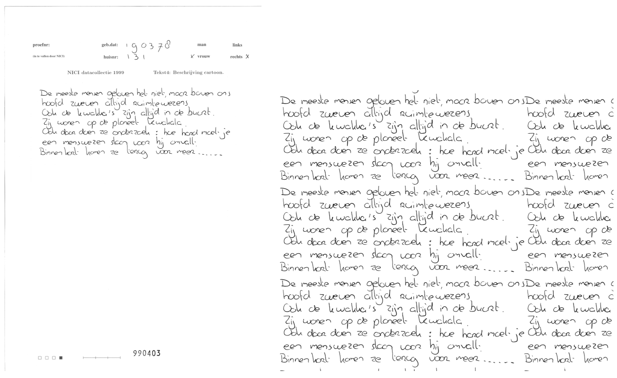


Fig. 4. Three ICDAR17 examples of text from different writers.

D. Image Preparation

Every page image of each dataset is cropped to avoid unnecessary blank backgrounds and to obtain handwritten images as compact as possible. As a result, we obtain different sized cropped images. Fixed-size images are needed for neural networks. For this purpose, we perform *text padding*. First, a cropped image is padded with white background to the maximum height and width of all cropped images. Then, the blank space is filled with fragments of the original text, performing a copy from left to right and top to bottom, as shown in Fig. 5. This can also be seen as a form of "data augmentation". So, the aim of this technique is twofold: getting fixed image sizes and data augmentation.

After geometry normalization, standard image intensity and color normalization is also applied to every dataset by shifting and scaling RGB values of each image using the mean and standard deviation of pixel RGB values of all the images in the dataset.



(a) Raw image (b) Cropped and text padded

Fig. 5. Example of the proposed text padding. (a) original page (b) cropping and applying text padding to (a).

V. Experiments and Results

In this section, we will do a series of experiments to end up with a good performing model. We start with an initial experiment working with commonly used patch image sizes in literature. Next, we will extend this experiment, trying even bigger patch sizes that never have been tried in literature. Furthermore, in Section V-C, we investigate the performance of our models in historical manuscripts, which is known to be a challenging task. All results of the patch-based models are presented in Table III. After that, we will experiment with our proposed full-page input. Finally, we present the results of our best models and compare them with the state of the art in Section V-E.

A. Experimental Settings

The convolutional weights of the model have been restored from the pre-trained model with ImageNet, as mentioned in the approach section III. The weights of the modified linear layer initialized following [35]. Patch-based experiments are trained according to the cross-entropy loss for 20 epochs using stochastic gradient descent (SGD), with a learning rate of 0.01 and a batch size of 32. Full-pages experiments are also trained with cross-entropy and SGD but with a learning rate of 0.002 and a batch size of 9. The selected train and test partitions have been explained in the previous sections, each in its respective dataset section. In all the experiments, accuracy over test partition is reported and used as a metric to compare the different results.

All the models were trained with two Nvidia GeForce RTX2080 GPUs and Pytorch v1.5 [36]. All the code is publicly available in order to reproduce all the experiments done in this work. [4]

B. Performance of different patch sizes

Our initial experiments started with small-sized image patches. To do so, we employ the patch sizes found in literature: 32×32 [4], 64×64 [30], 113×113 [29], [31], 224×224 [5] and 256×256 used by the University of Fribourg

[4]https://github.com/akpun/writer-identification

TABLE III
ACCURACY (IN %) FOR INCREASING PATCH SIZES. THE NUMBER OF PATCHES WAS 300 FOR IAM AND FIREMAKER AND 64 FOR ICDAR17

Patch size	IAM	Firemaker	ICDAR17
100	84.8	78.0	22.6
256	95.7	98.0	63.8
500	**96.8**	98.0	78.3
600	**96.8**	98.4	81.0
800	94.6	98.4	**83.6**
1000	95.0	**99.2**	-
1200	95.4	-	82.5

on the competition of ICDAR17. The proposed model was not capable of discerning any valuable features with patch sizes smaller than 100, so the results were not included. With a patch size of 256, it reaches 96% and 98% accuracy for IAM and Firemaker, respectively. The voting scheme seems to improve accuracy noticeably.

Furthermore, we noticed an increase in accuracy with a larger number of patches. We have done several experiments with different numbers of patches ranging from 10 to 300. Overall, we obtain good results with 100 and 200 patches and obtain the best results with 300 patches. For the sake of clarity and conciseness, we only include the results with 300 patches in IAM and Firemaker and 64 for ICDAR17. Additionally, it is better to minimize the number of patches to achieve an efficient model. For that reason, we have not experimented with more than 300 patches as it requires higher training costs.

With the steep upward trajectory of the accuracy with bigger patch sizes, we tested bigger patch sizes to see until what size the accuracy peaks. The chosen image sizes [5] to test were of 300, 400, 500, 600, 800, 1000 and 1200. Table III shows our best results.

For the IAM dataset, the accuracy peaks with a patch size of 600 achieving 96.8% accuracy. With bigger sizes, it plateaus or slightly decreases.

For the Firemaker dataset, the accuracy already reaches 98% with a size of 256; however, it keeps improving with increasing sizes. With the biggest size, we were able to obtain an incredible result of 99.2%.

Overall, with bigger patches, we have obtained excellent results. It seems that having big patches provides a lot of information to the neural network. Combining it with the voting scheme, the model has less margin of error and obtains better results.

C. Performance on Historical Documents

Previous experiments were done on IAM and Firemaker, which are contemporary datasets. Historical manuscripts have proven to be more challenging for writer identification. These types of documents are quite noisy, suffer deterioration, and are typically written in cursive, which is more difficult to understand. In these circumstances, it is more than interesting to know how well can our model cope with these types of datasets. We use the ICDAR 2017 dataset for this purpose.

As seen before in the contemporary datasets, the big patches based model performs better. Our best results are obtained with a patch size of 800 with an accuracy of 83.6%. Increasing more the patch size does not improve the performance; on the contrary, it slightly decreases. Based on our results, we can confirm the difficulty that faces writer identification on these types of datasets. In general, the accuracy does not seem to go beyond the 85% mark. A comparison could be made with the results obtained by the team of researchers of the University of Fribourg on the competition of ICDAR 2017 [10] with 47.8% accuracy or with [37] with 88.9% accuracy. Nevertheless, it is not comparable, as our test set is different because their test set was provided for writer retrieval purposes.

D. Performance of using full-page images as input

As mentioned before, all the deep learning approaches up to date use small image patches from the original page. Observing that no studies use full pages as input images, we decided to use full pages to see how well can the neural network perform with a full page.

We obtain a top 1 test accuracy of 91.3% in IAM. 98.3% in Firemaker and 83% in ICDAR 2017. The results seem to be quite satisfactory. Firemaker, in particular, achieves the state of the art results.

The results indicate that our proposed page model is less effective than the patch-based models. The voting scheme used for the patch models helps significantly it to perform better. The reason is that when testing, the model has a bigger margin of committing errors in patch images, while in the case of pages there is a single image.

E. Results

In this section, we compare our results with the state of the art methods on the IAM and Firemaker datasets. In Table I we had included all methods up to date, as far as we have studied, including handcrafted approaches and deep learning approaches. In Table IV we present our best results and compare them with the best results in the state of the art in Table I. Our methods have obtained a new state of the art results in Firemaker and competent results in IAM.

The experiments show a clear performance improvement of using bigger patch sizes. We hypothesize that features like the spacing between lines or the spacing between words are well captured by our models and may play a role in our improved results. Further studies need to be done to determine the plausible reasons for the achieved performance.

VI. CONCLUSION

In this paper, we presented a comprehensive summary of writer recognition approaches, which included both traditional methods and deep learning methods. We focused especially on papers that included their results in benchmark datasets, and we list all these results in Table I. We realized during this review that the proposed deep learning approaches have

[5] All the sizes refer to a square patch. To simplify we state a single value.

TABLE IV
BEST RESULTS WITH OUR PROPOSED PATCH-BASED MODELS AND FULL-PAGE MODELS.

	FULL IMAGE	PATCHES			BEST FROM
DATASET	Accuracy (%)	Accur, (%)	Patch Size	# Patches	Table I
IAM	91.3	96.8	600	300	98.5 [1]
FireMaker	98.3	99.2	1200	300	98.0 [23]
ICDAR17	83.0	83.6	800	64	–

worked only in patch images as input data, instead of using the full pages as input data. Thus, one of our approaches included experimenting with full pages.

To preprocess our images, a simple but effective data augmentation method, which we refer to as *text padding*, is also proposed. This method also provides image size normalization and, instead of performing padding with blank values to an image, it fills the padded area with the handwritten text copied from parts of the original image.

We presented two variant of models that perform end to end writer identification. Our different variants of models vary depending on the type of data, which can be patch images of different sizes or page images. The patch-based models compute the most probable writer for each patch and determine the writer for the full page based on the most voted writer over all the patches. The page-based model uses a single image of the full page to determine the writer.

We tested these methods on the IAM and Firemaker dataset and obtained competitive accuracy for the IAM dataset and better than the state of the art for the Firemaker dataset. Additionally, we have proposed a new, challenging benchmark for the writer recognition, called ICDAR17. It consists of a reorganization of the corpus proposed in the ICDAR 2017 competition on Historical Document Writer Identification, where the dataset training and test partition is adequately redefined for writer recognition experimentation. We also tested our models in ICDAR17 to assess their performance in historical datasets. The results indicate that our models are competent even for historical manuscripts but also prove that these type of handwritten images are more challenging than those of contemporary writing.

Finally, our experiments demonstrated the promising performance of using bigger patch sizes. Features like the spacing between lines or the spacing between words may play a role in our improved results. However, further studies need to be done to determine the real reasons behind the improved performance.

ACKNOWLEDGMENTS

Work partially supported by the BBVA Foundation through the 2018–2019 Digital Humanities research grant “Hist-Weather – Dos Siglos de Datos Climáticos”. and by the EU JPICH project “HOME – History Of Medieval Europe” (Spanish PEICTI Ref. PCI2018-093122). The second author’s work was partially supported by the Universitat Politècnica de València under grant FPI-I/SP20190010. Computing infrastructure was provided by the EU-FEDER Comunitat Valenciana 2014-2020 grant IDIFEDER/2018/025.

REFERENCES

[1] X. Wu, Y. Tang, and W. Bu, “Offline text-independent writer identification based on scale invariant feature transform,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 526–536, 2014.

[2] Y. Tang, W. Bu, and X. Wu, “Text-independent writer identification using improved structural features,” in *Chinese Conference on Biometric Recognition*. Springer, 2014, pp. 404–411.

[3] S. Fiel and R. Sablatnig, “Writer identification and retrieval using a convolutional neural network,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2015, pp. 26–37.

[4] V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, “Offline writer identification using convolutional neural network activation features,” in *German Conference on Pattern Recognition*. Springer, 2015, pp. 540–552.

[5] Y. Tang and X. Wu, “Text-independent writer identification via cnn features and joint bayesian,” in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 566–571.

[6] S. Al-Maadeed, A. Hassaine, A. Bouridane, and M. A. Tahir, “Novel geometric features for off-line writer identification,” *Pattern Analysis and Applications*, vol. 19, no. 3, pp. 699–708, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] U.-V. Marti and H. Bunke, “The iam-database: an english sentence database for offline handwriting recognition,” *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 2002.

[9] L. Schomaker, Lambert; Vuurpijl, “Firemaker image collection for benchmarking forensic writer identification using image-based pattern recognition,” 2000, https://zenodo.org/record/1194612#.XqHAb8szbmE.

[10] S. Fiel, F. Kleber, M. Diem, V. Christlein, G. Louloudis, S. Nikos, and B. Gatos, “Icdar2017 competition on historical document writer identification (historical-wi),” vol. 01. IEEE, 2017, pp. 1377–1382.

[11] G. J. Tan, G. Sulong, and M. S. M. Rahim, “Writer identification: A comparative study across three world major languages,” *Forensic science international*, vol. 279, pp. 41–52, 2017.

[12] H. Said, T. Tan, and K. Baker, “Personal identification based on handwriting,” *Pattern Recognition*, vol. 33, no. 1, pp. 149 – 160, 2000.

[13] Z. He, X. You, and Y. Y. Tang, “Writer identification of chinese handwriting documents using hidden markov tree model,” *Pattern Recognition*, vol. 41, no. 4, pp. 1295 – 1307, 2008.

[14] B. Helli and M. E. Moghaddam, “A text-independent persian writer identification based on feature relation graph (frg),” *Pattern Recognition*, vol. 43, no. 6, pp. 2199 – 2209, 2010.

[15] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, “Texture-based descriptors for writer identification and verification,” *Expert Systems with Applications*, vol. 40, no. 6, pp. 2069–2080, 2013.

[16] A. Brink, J. Smit, M. Bulacu, and L. Schomaker, “Writer identification using directional ink-trace width measurements,” *Pattern Recognition*, vol. 45, no. 1, pp. 162–171, 2012.

[17] R. Jain and D. Doermann, “Offline writer identification using k-adjacent segments,” in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 769–773.

[18] Ö. Kırlı and M. B. Gülmezoğlu, “Automatic writer identification from text line images,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 2, pp. 85–99, 2012.

[19] S. Fiel and R. Sablatnig, “Writer retrieval and writer identification using local features,” in *2012 10th IAPR International Workshop on Document Analysis Systems*, 2012, pp. 145–149.

[20] G. Ghiasi and R. Safabakhsh, “Offline text-independent writer identification using codebook and efficient code extraction methods,” *Image and Vision Computing*, vol. 31, no. 5, pp. 379–391, 2013.

[21] E. Khalifa, S. Al-Maadeed, M. A. Tahir, A. Bouridane, and A. Jamshed, “Off-line writer identification using an ensemble of grapheme codebook features,” *Pattern Recognition Letters*, vol. 59, pp. 18–25, 2015.

[22] A. Garz, M. Würsch, A. Fischer, and R. Ingold, “Simple and fast geometrical descriptors for writer identification,” *Electronic Imaging*, vol. 2016, no. 17, pp. 1–12, 2016.

[23] F. A. Khan, F. Khelifi, M. A. Tahir, and A. Bouridane, "Dissimilarity gaussian mixture models for efficient offline handwritten text-independent identification using sift and rootsift descriptors," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 289–303, 2019.

[24] L. Schomaker and L. Vuurpijl, "Forensic writer identification: A benchmark data set and a comparison of two systems, technical report." *Nijmegen University, Tilburg, The Netherlands*, 2000.

[25] M. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural and allographic features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 701–717, 2007.

[26] I. Siddiqi and N. Vincent, "Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features," *Pattern Recognition*, vol. 43, no. 11, pp. 3853–3865, 2010.

[27] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "Cvl-database: An off-line database for writer retrieval, writer identification and word spotting," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 560–564.

[28] G. Louloudis, B. Gatos, N. Stamatopoulos, and A. Papandreou, "Icdar 2013 competition on writer identification," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 1397–1401.

[29] L. Xing and Y. Qiao, "Deepwriter: A multi-stream deep cnn for text-independent writer identification," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 10 2016, pp. 584–589.

[30] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkhya, and M. Nakagawa, "Text-independent writer identification using convolutional neural network," *Pattern Recognition Letters*, vol. 121, pp. 104–112, 2019.

[31] S. He and L. Schomaker, "Fragnet: Writer identification using deep fragment networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3013–3022, 2020.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," no. NeurIPS, 2019.

[37] V. Christlein, M. Gropp, S. Fiel, and A. Maier, "Unsupervised feature learning for writer identification and writer retrieval," in *2017 14th Internationa Conference on Document Analysis and Recognition*, 11 2017, pp. 991–997.