Integrating Writing Dynamics in CNN for Online Children Handwriting Recognition

Simon Corbillé
Univ Rennes, IRISA lab
F-35000 Rennes France
simon.corbille@irisa.fr

Élisa Fromont
Univ Rennes, IUF, IRISA lab
F-35000 Rennes France
elisa.fromont@irisa.fr

Éric Anquetil
Univ Rennes, IRISA lab
F-35000 Rennes France
eric.anquetil@irisa.fr

Pauline Nerdeux
Univ Rennes, IRISA lab
F-35000 Rennes France
pauline.nerdeux@irisa.fr

***Abstract*—Online handwriting recognition is challenging but an already well-studied topic. However, recent advances in the development of convolutional neural networks (CNN) make us believe that these networks could still improve the state of the art especially in the much more challenging context of online children handwritten letters recognition. This is because, children handwriting is, at an early stage of learning, approximate and includes deformed letters. To evaluate the potential of these networks, we study the early and late fusions of different input channels that can provide a CNN with information about the handwriting dynamics in addition to the static image of the characters. The experiments on a real children handwriting dataset with 27 000 characters acquired in primary schools, show that using multiple channels with CNN, improves the accuracy performance of different CNN architectures and different fusion settings for character recognition.**

***Index Terms*—Online handwriting recognition, Convolutional neural network, Digital learning**

I. Introduction

Digital learning is about associating learning experiences with numerical technology. It is especially popular in the business world as well as in the educational environment. With digital learning, one can benefit both from the traditional active learning methods and from the digital tools which can take various modalities. Our end goal is to recognize and analyse **online children's handwriting** and, in particular, the **letters** they draw when starting to learn how to write, to be able to help them in this crucial step of their development where they have a very approximate graphomotor gesture. In particular, we would like, ultimately, to improve an existing software [1] that has already been deployed in primary schools and allowed us to record a large (27 000 characters) number of children handwriting sequences. This target software gives us a number of constraints: our method should be usable on tablets, work in real-time and be dedicated, for now, to the **Latin alphabet**. Children handwriting differs from the traditional (adult) handwriting analysis problem, since it is often difficult, even for a human eye, to recognize the children letters as depicted in Fig. 1.

A decade ago, Hidden Markov Model based on hand-crafted features [2] were state-of-the-art methods to perform online handwriting recognition. Nowadays, deep neural networks are trained end-to-end and include feature extraction layers mainly based on convolutions filters. The current state-of-art results

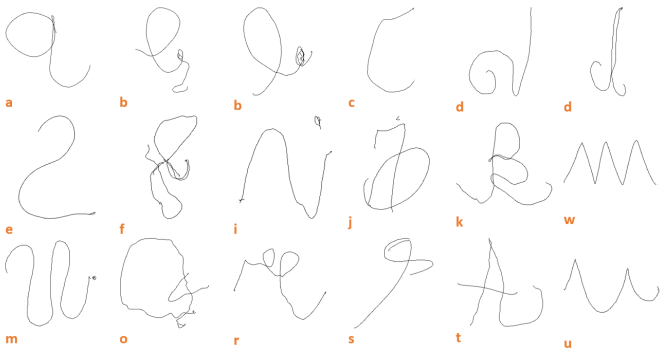

Fig. 1: Examples of deformed children handwritten letters. In black, the drawing corresponding with the real orange letter

in online handwriting recognition are obtained by recurrent neuronal networks [3] [4]. These networks often combine many different sub parts which make them relatively big (in space), slow (at inference time) and sometimes difficult to train. We would like to study how, simpler **convolutional neural networks** (CNN), already known to give excellent results on image analysis, could be used in this context.

In this work, we study three well-known CNN architectures: LeNet-5 [5] (the first well-known CNN, specialized for digit recognition), ResNet-18 [6] and VGG-11 [7]. In particular, we would like to evaluate if using different types of input channels which provide information about the handwriting dynamics like in [8], can improve the performance of the CNN compared to a vanilla setting using only the image channels (3 channels R/G/B). We evaluate an early and a late fusion. The early fusion consists in combining different channels at the input of a single network. The late one consists in using a combination of neural network classifiers trained on the different channels separately to improve the classification performance.

The main contributions of this article are:

- Conversion of online data to image with dynamics information;
- Comparison of three famous CNN architectures and two usual types of fusion.

The paper is organized as follows. The related work is presented in section II. Then, we describe in details in section III our strategy to encode the dynamics of a handwriting sequence

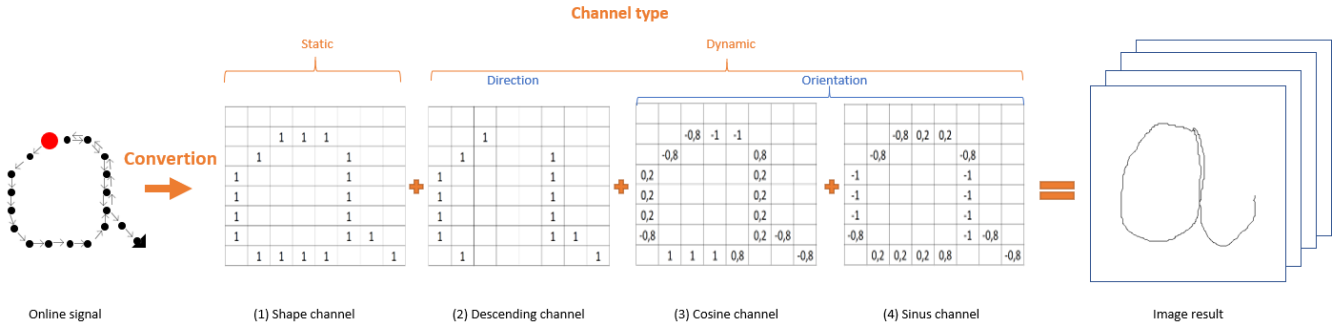

Fig. 2: Transformation of online handwritten characters to static and dynamic channels to product an image. When there is no pencil line, the corresponding value in the matrix is zero.

into a single image. Section IV details the two proposed fusion methods to take into account these complementary inputs. Section V presents the results and show how the use of multiple channels can indeed improve children handwriting recognition. Conclusion and perspectives are given in section VI.

II. Related work

Many recent work ([9], [10], [11], [12], [13] and [14]) use deep learning to automatically recognize offline handwritten texts in, respectively, Amharic, Mongolian, Latin, Bengla and Chinese languages. They mainly use **convolutional** and **recurrent** networks or a combination of both to perform the recognition task. The usual performance metrics when dealing with character or word recognition are the CER (Character Error Rate) and the WER (Word Error Rate) which give respectively the percentage of errors in classification (at the level of the character or at the word level) on a given test set.

CNN are preferred to perform **offline handwriting recognition**. They can be coupled with a text model to increase the performance such as in [15] where a CNN is coupled to a N-gram model. They achieved 3.44% CER and 6.45% WER on the well-known IAM dataset [16] and 1.90% CER and 3.90% WER on RIMES dataset [17].

The current best results in **online handwriting recognition** are obtained with recurrent neural network (see e.g. [3] and its extension [4]). The networks presented in [3] and [4] give, respectively 4.3% CER and 10.4% WER and 2.5% CER and 6.5% WER on IAM-OnDB [18], the online version of the IAM database.

The use of CNN for **online signal recognition** had been studied for the task of **action recognition** in [19] and [20]. In [19], the authors use three types of inputs relative to gesture as input to a CNN. Each input is treated separately (in a branch) of the network and the features extracted are concatenated before being given as input to a last CNN which performs the classification. They used three different channels to represent dynamic information contained in the online signal as input for the CNN. The idea is to used this approach for mixing dynamic information with static ones, which represent the online handwriting, as input for CNN.

Our end goal is to help children learn how to write at school. To fulfill this goal, real-time and modularity are strong requirements for any part of an analysis pipeline. This is the reason why, in this paper, we focus on CNN which are deemed more efficient than recurrent neural network at inference time and easier to integrate in a complex pipeline. Besides, we would like to evaluate how the dynamical information provided by the online recording of the handwriting can be successfully taken into account in such CNN. We thus explore two fusion schemes: an early and a late one that are developed in the following.

III. Online handwriting encoding into channels

Online handwriting can be modelled with a time series where each point is represented by a 4D vectors which encodes the 2D coordinates of the point (x, y), a pressure value and a timestamp. We do not use pressure data in our study. Our goal is to convert the online signal (i.e. the time series) into a set of multichannel images where each channel represents either static or dynamical information about the original signal. Images are mostly represented in black and white, grey-scale or color (R/G/B). In character or word recognition, the color information is not particularly relevant to the recognition process compared to the shape so, the letters are usually encoded into a grey-scale image matrix in a single input channel. In this paper, we explore what additional information that characterize the dynamics of the online signal could bring to the recognition performance of a network. This section describes how each type of information is extracted and encoded in the different channels.

A. Shape information encoding

A few pre-processing steps are necessary to convert an online signal represented by a times series into an image represented by a matrix of values. The whole pipeline of conversion is illustrate in Fig. 3.

First, a **linear normalisation** step sets the coordinates values between a min and max value which represent the bounds of a 32×32 image. Then, a **spatial sampling** step allows to fill in a gap between two points to link them. We

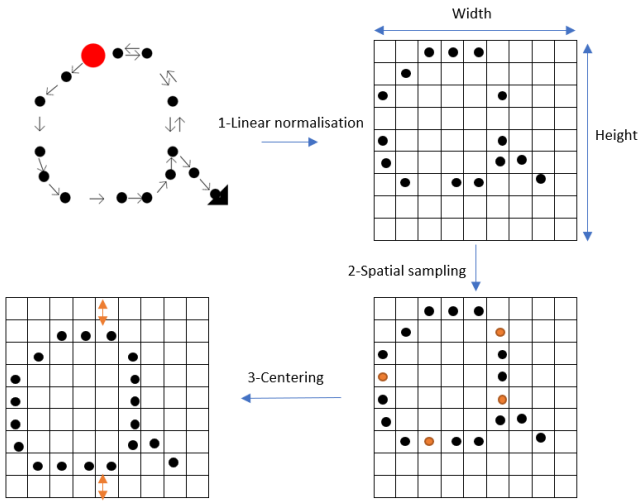

Fig. 3: Data pre-processing of online time series to obtain static images.

center the points to avoid a top right alignment then convert these points to a black and white image. The white value corresponds to a point of handwriting, and the black value corresponds to the background (i. e. one when there is a handwriting, zero if there is not). Thereby, we have converted the list of points to a one channel image.

B. Orientation information encoding

The letter represented by an online signal can be split into ascending and descending lines as shown in Fig. 4. An ascending line represents a handwritten line which has been created bottom up. As soon as the pencil or stylus move downwards, it becomes a descending line. This decomposition provides information about the ductus of the handwritten character. In the following, we only use the information about the **descending strokes** because this characteristic is more discriminating than the other one [21]. The value is one if the descending stroke passes by this point; else it is set to zero.

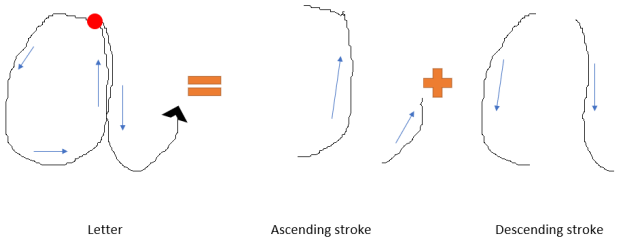

Fig. 4: Decomposition of ascending and descending lines.

C. Direction information encoding

To add some information about the **direction** of the signal, we use the angle between one point and the next point in the time series. To model the angle, we both use its cosine and sinus. The image is representing by a matrix of pixels. Therefore, the next point can be in one of the 8 boxes next to the point which gives us 8 possible angle values as is illustrated in Fig. 5. The cosine and the sinus have values between -1 and 1. Since we already assigned the zero value to indicate the absence of a stroke, we re-normalized the cosine and sinus between 0.2 and 1 to avoid any confusion and multiplied this value by 255 to have values in the color domain. We use the following formula to normalise the value x to x_n :

$$x \in [a, b] \text{ and } x_n \in [c, d]$$
$$x_n = \frac{d-c}{b-a} * (x - a) + c$$
With a = -1, b = 1, c = 0.2, d = 1

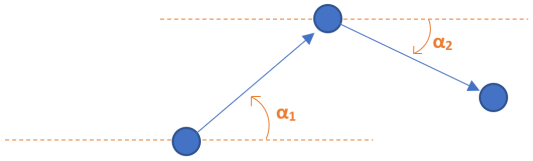

Fig. 5: Example of angle α representation between points in the series.

The cosine and sinus are treated independently in different channels. The different combinations of inputs that we tested are presented in Section V. To summarize, the four types of channel which represent static and dynamical information we use are:

- type 1 : shape of the letter (static information)
- type 2 : orientation represented by descending stroke (dynamical information)
- type 3 : direction represented by cosine value of the angle with the next point (dynamical information)
- type 4 : direction represented by sinus value of the angle with the next point (dynamical information)

IV. Multichannel fusion in convolutional neural networks

We study two different fusion schemes to take into account the online information provided in the channels described in section III: an **early fusion** where the channels are given as input to a single CNN and a **late fusion** where each channel is provided separately to different networks and the predictions are combined *a posteriori*.

A. Early fusion in different CNN architectures

We compare the impact of using different channels on the handwritten character classification performance of three classical CNN architectures: LeNet-5 [5], VGG with 11 layers [7] and ResNet with 18 layers [6]. LeNet-5 is one of the first well-known convolutional neural networks which brought good performance on digit recognition. LeNet-5 is compact (around 60000 parameters) compared to more recent architectures but it is not sufficiently deep to tackle very complex tasks

in computer vision. VGG and ResNet are more recent and deeper networks that have achieved very good performance in the very challenging ImageNet classification challenge [22]. ResNet introduced the residual connections that proved to be key to successfully train very deep networks. We choose a shallow version of both VGG (11 layers for about 28 millions parameters) and ResNet (18 layers for about 11 millions parameters) because we believe that larger ones would be more prone to over-fitting on our, comparatively to ImageNet, rather a small dataset.

B. Late fusion with ensembles

We compare our early fusion approach with a late fusion one where each channel is provided to a different instance of the same architecture (either LeNet, VGG or ResNet). We then build three different ensembles of neural network classifiers which architecture is presented in Fig. 6 and compare their classification and time performance. Each ensemble consists in a set of four networks, one for each type of channel presented before. The predictions of these networks are merged to obtain a global prediction. We tested a naive approach where an equal weight is given to each individual classifier in the ensemble.

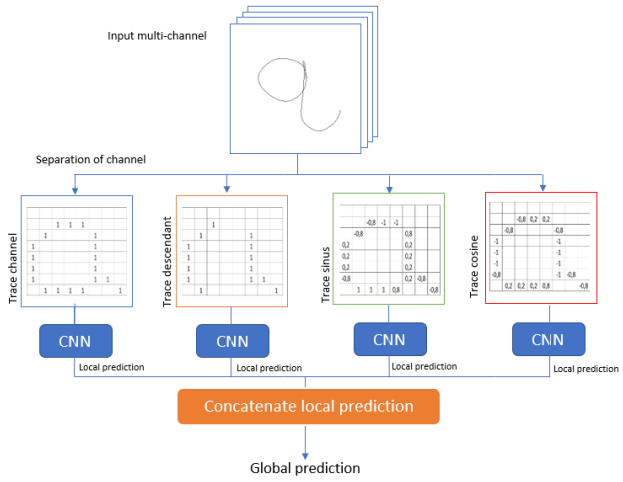

Fig. 6: Ensemble architecture: there is one network per input channel.

V. Experiments

We use accuracy (1-CER) as our performance metric in this section. We also provide inference time (in ms) measures for each configuration (fusion type and architecture). All experiments were made on a computing grid composed of several GPU nodes except the inference time measures that were measured on a CPU. We fixed the *batch size* hyper-parameter to 128, choose ADAM optimizer and set the learning rate to 10^{-4} for all training procedures. The networks were all implemented in Python using Keras[1]. For each configuration of input, we tune the *early stopping* hyper-parameter of each network.

[1]https://keras.io

We tested the four following combinations of channels:

- Config A (baseline) : this configuration uses a single channel which encodes the static pencil line i.e. the shape of the letter (type 1).
- Config B : we add two channels to the previous baseline (A) which encode the dynamic information about the direction of the signal in the form of cosine and sinus values (type 1 + type 3 + type 4).
- Config C : we add one channel to the baseline (A) which encodes the orientation of the signal (type 1 + type 2).
- Config D : we use all four available channels (type 1 + type 2 + type 3 + type 4).

Note that the last configuration (D) is the one that contains the most information about the dynamics of handwriting. Configurations B and C subsume the configuration A but are not comparable since one (C) is using the orientation information but not the direction whereas the other (B) only uses the direction without orientation information.

A. Dataset acquisition

As explained in the previous section, there are some known benchmark datasets for online handwriting recognition [16], [23], but they all provide examples of adult handwriting. Because our work is part of a much bigger project which aims at providing feedback to children learning to write, we designed a new dataset with examples of children handwriting. To do so, we have used an existing platform [1], which is already deployed in some primary schools and has allowed us to collect diverse handwriting sequences made by children writing with a pen on digital tablets[2]. This dataset is private since children handwriting are considered protected personal data. Each handwritten letter (and the entire word if applicable) is recorded as a multivariate time series. This new dataset contains letters naturally distorted on several aspects which may fool a classifier trained on adult handwriting. However, since neural networks are data greedy, we use data augmentation techniques to increase the size of our children handwriting training dataset. Our data augmentation strategy consists in deforming the original letters to create more examples of plausible children handwriting sequences. First, we apply some usual operations such as stretching, inclination and rotation. Then, we use two techniques which are described in [24] which modify the curvature of the stroke and the speed with which the stroke was drawn. We also used other techniques such as stretching and translation at the stroke level. Fig. 7 illustrates some of these techniques.

To create our training dataset, we started from a base of the initial examples from the first version of IntuiScript [1] dataset which contains about 27 000 handwritten characters written by 147 children. Then, we augmented it in order to obtain 5 000 examples per class (10 000 for the ”e” and ”x” letters because we considered two ways to draw them). Since we focus on the Latin alphabet, our training dataset contains **140 000** elements.

[2]See https://www-intuidoc.irisa.fr/children-handwritings-database/ for more information about the dataset.

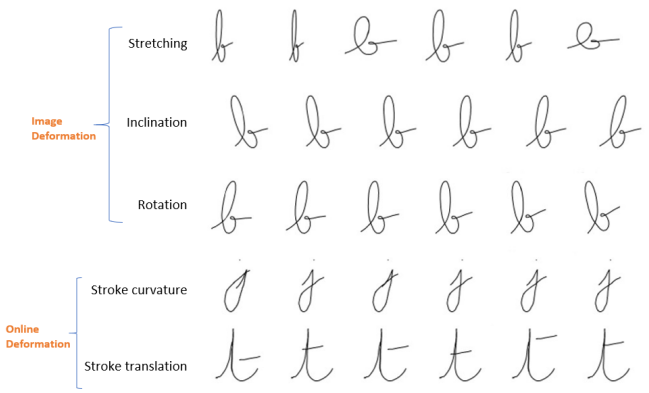

Fig. 7: Example of image and online deformations.

In our test dataset, we do not use augmentation set aside 9 686 letters with no class balancing. We extracted from it 100 examples per class to build a validation set. Our test dataset is composed of **7 096** samples and our validation dataset of **2 600** samples.

B. Early fusion

The results obtained using the different input configurations (A, B, C, D) for the three network architectures (LeNet, VGG and ResNet) are given in Table I. Each experiment is run twice and both results are averaged in the table and presented with their variance.

Input Config	Static A	Static + Dynamic B	C	D
LeNet-5				
Accuracy	90.62	91.35	91.93	**92.66**
Variance	0.0240	0.1599	0.0121	0.0049
ResNet-18				
Accuracy	94.05	94.06	94.33	**94.64**
Variance	0.0100	0.1764	0.0030	0.0030
VGG-11				
Accuracy	94.54	94.64	94.84	**95.00**
Variance	0.0066	0.0580	0.0323	0.0042

TABLE I: Classification results (accuracy) for each architecture and input configuration

These results show first that the best test accuracy (95%) can be obtained with the VGG-11 network which is the one with the highest number of parameters. This accuracy is suitable and very promising for a real-time deployment in an existing platform. Whatever the architecture, when adding some information about the dynamics of the drawing in the network (B,C,D), the results are better than with only the static information (A). With the highest amount of information (D), the results are the best and even more stable (low variance). This shows that it is important to finely encode the dynamical information to improve the performance of a CNN.

C. Late fusion

We use in these experiments the same early-stopping criteria and values as for the early fusion experiments. For each type of architecture, we trained one neural network for each of the 4 channels (type 1 to 4) and merge their probabilities predictions with the mean function with an equal weight to each network an select the prediction with the higher probability. The results are given in Table II.

Input Config	Trace Type 1	Dynamic Type 2	Type 3	Type 4	Ensemble
LeNet-5					
Accuracy	91.10	77.49	90.78	88.15	**93.35**
ResNet-18					
Accuracy	93.35	74.44	93.60	93.19	**94.87**
VGG-11					
Accuracy	94.24	75.30	94.24	93.77	**94.76**

TABLE II: Classification results (accuracy) for each architecture and input configuration

As also noticed for the early fusion, using a late fusion of multiple channels gives better results than all the channels taken separately. Interestingly, we can see that using only the descendant stroke information in a network (type 2) gives very bad performance for all architectures so this feature is not sufficient in itself for classification. The ensemble accuracy results are similar to the ones obtained with an early fusion approach which shows that the fusion scheme is less important than the expert dynamical information that can be encoded in each channel. We can however note that with ensembles of LeNet and ResNet classifiers, we obtain slightly better results than with the early fusion which is not the case for the VGG ensemble even it is again the architecture with the best overall results.

Note that we also tried to learn a weight associated to each classifier in the ensemble but the results were not better than the ones presented in Table II and are thus not shown here.

D. Inference time

We computed the inference time on the test dataset (7086 samples) for the configurations introduced before. The measurements were done without GPU and a Intel i7-7600U, 2.80GHz CPU. We used a batch size of 1 and report in Table III the average processing time for the entire test set.

Architecture \ Input	A	B	C	D	Ensemble
LeNet	0.68	0.71	0.77	0.76	2.80
ResNet	13.33	13.96	14.35	15.33	59.39
VGG	18.89	18.51	18.94	19.46	76.43

TABLE III: Inference time (in **ms**) for each architecture each input configuration (A,B,C,D) and for the late fusion (Ensemble).

This table shows that the inference time is low and meets the real-time requirement of our target software for all early fusion configurations. Unsurprisingly, the inference time for LeNet is much lower than for the two other architectures. The late fusion approach takes much more time than the early fusion one and might not meet the real-time requirement in a more complex pipeline. Adding channels that encode the

dynamics in the early fusion scheme does not significantly increase the inference time. There is a clear trade-off between the complexity of the deep learning model and the inference time. We believe that the ResNet architecture, which is a little bit less accurate than the VGG one, might still be preferred for its better inference time.

VI. Conclusion and perspectives

We studied the early and late fusions of multiple channels with different convolutional neural networks for online children handwriting recognition. We showed that we can improve the performance of CNN in terms of accuracy by adding dynamic information in the input channels of the networks for both the early and late fusion approaches. We achieved 95.00% of accuracy and nearly 20ms of inference time when predicting one letter with the VGG architecture and an early fusion scheme. This is very promising to integrate such a network in a complete analysis pipeline.

Converting the online signal into multiple image channels was one way of using the dynamical information to improve the performance of a CNN. We would like to explore the use of CNN directly on the time series signal as done for example in [25].

References

[1] Damien Simonnet, Nathalie Girard, Éric Anquetil, Mickaël Renault, and Sébastien Thomas. Evaluation of children cursive handwritten words for e-education. *Pattern Recognit. Lett.*, 121:133–139, 2019.

[2] Adrien Delaye and Éric Anquetil. HBF49 feature set: A first unified baseline for online symbol recognition. *Pattern Recognit.*, 46(1):117–130, 2013.

[3] Daniel Keysers, Thomas Deselaers, Henry A. Rowley, Li-Lun Wang, and Victor. Carbune. Multi-language online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1180–1194, 2017.

[4] Victor Carbune, Pedro Gonnet, Thomas Deselaers, Henry Rowley, Alexander Daryin, Marcos Calvo, Li-Lun Wang, Daniel Keysers, Sandro Feuz, and Philippe Gervais. Fast multi-language lstm-based online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 2020.

[5] Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[8] Zhen-Long Bai and Qiang Huo. A study on the use of 8-directional features for online handwritten chinese character recognition. In *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea*, pages 262–266. IEEE Computer Society, 2005.

[9] B. H. Belay, T. Habtegebirial, M. Liwicki, G. Belay, and D. Stricker. Amharic text image recognition: Database, algorithm, and analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1268–1273, 2019.

[10] Yanke Kang, Hongxi Wei, Hui Zhang, and Guanglai Gao. Woodblock-printing mongolian words recognition by bi-lstm with attention mechanism. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 910–915. IEEE, 2019.

[11] Gideon Maillette de Buy Wenniger, Lambert Schomaker, and Andy Way. No padding please: Efficient neural handwriting recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 355–362. IEEE, 2019.

[12] Nishatul Majid and Elisa H. Barney Smith. Segmentation-free bangla offline handwriting recognition using sequential detection of characters and diacritics with a faster r-cnn. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 228–233, 2019.

[13] Yuhuan Xiu, Qingqing Wang, Hongjian Zhan, Man Lan, and Yue Lu. A handwritten chinese text recognizer applying multi-level multimodal fusion network. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1464–1469, 2019.

[14] Dezhi Peng, Lianwen Jin, Yaqiang Wu, Zhepeng Wang, and Mingxiang Cai. A fast and accurate fully convolutional network for end-to-end handwritten chinese text segmentation and recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 25–30, 2019.

[15] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwriting word recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Urs-Viktor Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *IJDAR*, 5(1):39–46, 2002.

[17] E. Grosicki, M. Carré, J. Brodin, and E. Geoffrois. Results of the rimes evaluation campaign for handwritten mail processing. In *10th International Conference on Document Analysis and Recognition (ICDAR)*, pages 941–945, 2009.

[18] Marcus Liwicki and Horst Bunke. Iam-ondb - an on-line english sentence database acquired from handwritten text on a whiteboard. pages 956–961, 2005.

[19] Fan Yang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. Make skeleton-based action recognition model smaller, faster and better. *Proceedings of the ACM Multimedia Asia*, 2019.

[20] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex Kot. Skeleton-based online action prediction using scale selection network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 02 2019.

[21] Éric Anquetil and H. Bouchereau. Integration of an on-line handwriting recognition system in a smart phone device. In *16th International Conference on Pattern Recognition (ICPR)*, pages 192–194, 2002.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[23] Isabelle Guyon, Lambert Schomaker, Rejean Plamondon, Mark Liberman, and Stan Janet. UNIPEN project of on-line data exchange and recognizer benchmarks. In *12th IAPR International Conference on Pattern Recognition (ICPR)*, pages 29–33. IEEE, 1994.

[24] Harold Mouchère, Sabri Bayoudh, Eric Anquetil, and Laurent Miclet. Synthetic On-line Handwriting Generation by Distortions and Analogy. In *in 13th Conference of the International Graphonomics Society (IGS)*, pages 10–13, 2007.

[25] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.*, 33(4):917–963, 2019.