Can One Deep Learning Model Learn Script-Independent Multilingual Word-Spotting?

Mohammed Al-Rawi*, Ernest Valveny, and Dimosthenis Karatzas
Computer Vision Center
Campus UAB, Edifici O, 08193 Cerdanyola del Valls, Barcelona
ms.alrawi@gmail, ernest@cvc.uab.es, dimos@cvc.uab.es

***Abstract*—Word spotting can be used to extract textual information from handwritten documents and scene-text images. Current word spotting approaches are designed to work on a single language and/or script. Building intelligent models that learn script-independent multilingual word-spotting is challenging due to the large variability of multilingual alphabets and symbols. We used ResNet-152 and the Pyramidal Histogram of Characters (PHOC) embedding to build a one-model script-independent multilingual word-spotting and we tested it on Latin, Arabic, and Bangla (Indian) languages. The one-model we propose performs on par with the multi-model language-specific word-spotting system, and thus, reduces the number of models needed for each script and/or language.**

Keywords - Scene text images; Handwriting; Pyramidal Histogram of Characters; PHOC; Multitasking, ResNet

I. Introduction

Word-spotting is a matching task between one or multiple query word-images or strings and a set of candidate word-images in documents and/or scenes. Word-Spotting has received unprecedented attention in recent years as it can be used in information retrieval from handwriting documents and scene text images [10]. Deep Convolutional Neural Networks (DCNNs) have been used recently to build word-spotting systems that are able to achieve staggering performance [15, 32, 33, 34]. Usually, the word image is used as input to a DCNN and the string corresponding to the word is embedded into some form, using for example the Pyramidal Histogram of Characters (PHOC) [5, 33, 27, 30].

Text spotting and recognition in a natural environment is a key component of many applications [10] and can be used, for example, as a complementary component in intelligent transport. Another vital area is social media and networks that contain huge amount of text composited onto images and on Instagram [7, 12]. By the time of writing this manuscript, text found in the wild has tremendous size [15] and the Instagram hashtag #handlettering enlists $7,275,272$ images and the hashtag #lettering enlists $9,776,820$ images, containing printed and handwritten captions composited onto images [3]. Samples are shown in Fig. 1. Scene text word-spotting can therefore help remove violent, threatening or even terrorist information from the Internet and take action if necessary. It can also be used by social media and networks to gather critical information about their users' personal preferences, similar to other information being collected now in different domains [31]. Word-spotting of scene text images will therefore bring tremendous value in the near future.

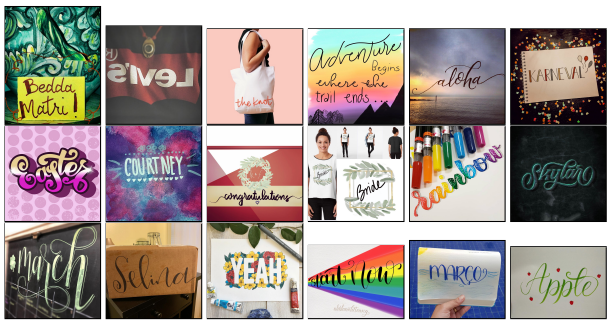

Figure 1. Samples from the Instagram #handlettering images that we used in the experiments. Best viewed in color.

Different deep learning models have been used to build word-spotting systems, some of which are tailored to cope with the specific requirements of word-spotting, as described in [32]. Recent models, however, are based on state-of-the-art DCNNs [30]. Moreover, data augmentation methods have recently been applied to enhance the performance of word-spotting systems based either on introducing distortions [34, 30] to the training set or on using a large corpus of synthetic text images [17, 16].

We aim at building one model that can perform script independent multilingual word-spotting in handwriting and scene-text images. We thus propose a unified model aimed at performing word-spotting on English, French, German, French, Arabic and Bangla (Indian) languages. For illustrative purposes, we present in Fig. 2 the two modalities that

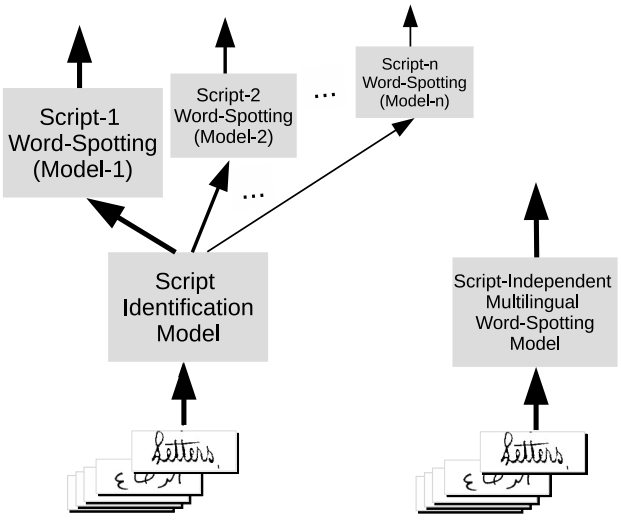

Figure 2. The proposed (joint) script-independent multilingual model (right) and the possible multi-model based system built with several uni-script models (left).

Figure 3. Latin, Arabic and Bangla samples from scene-text MLT images. Best viewed in color.

can be used to build script independent multilingual word-spotting systems. The joint model we are proposing should be able to perform script independent word-spotting, without any prior script identification step. To compare the performance of proposed multi-lingual model to the multiple uni-script word-spotting models, we have also implemented a baseline model using a DCNN to identify/discriminates the script type followed by several uni-script word spotting models, one for each script. In our experiments, we have used several datasets covering different domains of application of word spotting: for scene text, the Multilingual Transcription (MLT2017) dataset [23], that contains scene text images in modern cities, and a collected set of Instagram handlettering images. For handwriting, the IAM and GW English, and IFN Arabic handwriting datasets. Finally, we introduce a new dataset containing handwritten text on scene images that we have synthetically generated. This latter is a highly challenging handwriting learning task as the background of the handwriting word is a natural scene and not a smooth background. This dataset was inspired from the Instagram handlettering scene-text problem domain [13], the MLT-2017 dataset and scene handwriting images that are beginning to appear increasingly on social networks and media. Samples from the MLT-2017 dataset are shown in Fig. 3.

To the best of our knowledge, this is the first work that addresses building one deep learning model to perform script-independent multilingual word-spotting applied for document handwriting, scene text and scene handwriting scenarios. The importance of this work is therefore twofold: 1) pushing one DCNN to learn multilingual word-spotting as a multitasking, like human being, in normal handwriting and scene text in the wild; and 2) reducing the number of models needed in multilingual applications as one model will learn several languages regardless of their scripts.

II. Related Work

The major focus of the literature has been devoted to script identification problems [24, 29, 6, 14, 9, 35]. Hence, the aforementioned techniques rely on analyzing only a single script by measuring the characters' variations, using for example, analysis of connected components and character distributions as in [14] or image moment methods as in [9]. Furthermore, Google Vision API also adopts language / script identification model(s) prior to recognizing text in images [11]; the model/technique has not been opened to the public. Although script independent multilingual word-spotting via deep learning has not been reported in the literature, there has been some excellent works that used statistical and Bayesian approaches to tackle script-independent multilingual word-spotting [36, 18]. Regardless of the technique used for script identification, the natural evolution to perform multi-script word-spotting is to use a script identifier and then to feed the word to the respective word-spotting system that matches the language of the identified script, as shown in left part of Fig. 2. This approach, although might be successful, is tedious and redundant as the whole system will rely on $n+1$ models: one model to work as script identifier and n word-spotting models. What we are suggesting, shown in the right part of Fig. 2, is beyond state of the art as it uses one model to perform word-spotting for the n languages without the use of script identification model.

Regarding the baseline performance, word-spotting models have achieved remarkable results in terms of Mean

Average Precision (mAP) when used on handwriting data. See for example [30] that achieved 98.4% [32] using Query by String (QbS) on the George Washington (GW) English handwriting dataset [1] and 92.1% mAP(QbS) on IFN/ENIT Arabic handwriting dataset [2, 26]. Nonetheless, no word-spotting results have been reported on the MLT-2017, nor Instagram and the scene-handwriting datasets. For MLT-2017, however, there has been some recent works on script identification and OCR [23].

III. Methods

We build our word-spotting models using PyTorch framework [25, 28] and we make the source code publicly available at https://github.com/morawi/MLPHOC. Below we provide in-depth details about creating a word-spotting model and setting up datasets.

A. Word encoding via PHOC

The Pyramidal Histogram of Characters (PHOC) has frequently been used to convert the word string to a vector (label) that can used to supervise the learning of word-spotting models. The PHOC algorithm, $\zeta = \psi_m(l)$, simply divides the word into a few regions (m levels) and then finds the histogram of characters falling within each region [5]. The concatenation of all these histograms gives the PHOC vector ζ, which is the label that can be used to supervise the learning.

In the current word spotting systems, different scripts are associated with different PHOC representations with an independent set of symbols. Hence, each uni-script model is trained using the PHOC representation specific for that language/script. In this work, we propose to build a single PHOC representation for multiple scripts, basically by concatenating all symbols from all languages into a single set of symbols. In this way, a single model can learn words from any script. More formally, this can be formulated as follows:

$$\zeta = \psi_m(\cup_{k=1}^{n} l_k \cup \xi), \tag{1}$$

where m is the PHOC level, l_k is a set denoting the k_{th} language that has alphabets and numerals for $k = 1, 2, ..., n$, ξ is a set containing non-alphanumeric symbols, and n is the number of languages used. In this work, we use up to five PHOC levels, *i.e*. PHOC levels used are {2 ,3, 4, 5}, and bigrams or trigrams are omitted.

B. Retrieval protocol

We use a retrieval protocol similar to [5]. This protocol has also been adopted by all previous word-spotting works, see for example [32, 33, 30, 34]. We evaluate the performance of word-spotting systems using mean Average Precision (mAP), which is the standard evaluation method under retrieval problems. Similar to other previous works, for example [5, 32], we test the proposed word-spotting system using two measures: Query by Example (QbE) that was first proposed by Manmatha *et al*. in [19], and Query by String (QbS) that was first proposed by Edwards *et al*. in [8]. For QbE, we only use the unique strings in the test set as queries; each word image in the test set is used once as a query to rank the remaining word images in the test set. For QbS, however, we only take the unique words in the test set and use their PHOC representation as queries. As a distance measure, we use the cosine similarity distance.

C. Deep learning model

We use DCNNs to predict from an input image the PHOC vector encoding of each string denoting the text in that image. Without the need to build novel DCNNs specially dedicated to this supervised learning task, we opt to try a few of the already available DCNNs, *e.g*., ResNets, VGGs, SqueezeNets, DenseNets, DualPathNets, *etc*. Our exploratory ablation analysis showe that ResNets outperform the majority of the other DCNNs. ResNets not only excel in performance compared to other DCNNs, but they are also memory efficient.

We use transfer-learning DCNNs that have been trained with ImageNet in which PyTorch (via Torchvision) supports quite a few models [28]. The procedures and hyper-parameters that we use in the experiments are as follows: SGD, learning-rate of 0.1, lr was divided by 10 according to the following milestones $\{40, 80\}$, momentum $= 0.9$, weight-decay $= 10e$-14, dropout-probability $= 0$ (dropout has not been used), and a training batch size equal to 2. Working with small batch sizes has great generalization advantages as illustrated in [22]. Each image was re-scaled to 256×128. We use random shuffling during training, in addition to time randomizers to make variations for different training instances. We use Binary Cross Entropy loss with Logits; which combines a $Sigmoid$ layer and has thus a better numerical stability than using a $Sigmoid$ function followed by a loss function [28] during training. In most experiments, unless stated otherwise, we use a maximum of 100 epochs. After training is finished, the $Sigmoid$ function is deployed on the outputs to bound them between 0 and 1. Furthermore, we use rounding to convert the real-valued outputs to binary values.

D. Script identification deep learning model

The the use of Script identification is usually followed by multiple uni-script word-spotting models; a system that can be used as a baseline to compare with the unified model we are proposing. We use a binary classifier to identify the type of script. To do this, we replaced the final layer that outputs a PHOC sized vector with labels for each language/script. Our classifier is based on ResNet-18 and training batch of size 10, Cross Entropy Loss function to train the script iden-

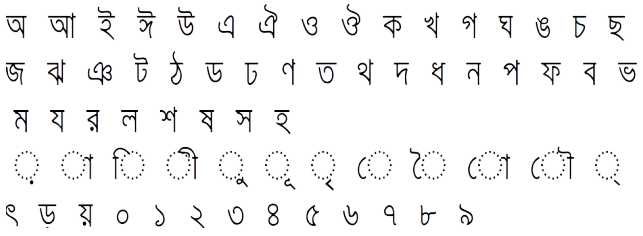

Figure 4. Bangla alphabets in MLT-2017 dataset. The total number of characters, however, is 109 as MLT contains other non-Bangla symbols.

tification model, otherwise, the rest of the parameters were similar to those used in our word-spotting model that are detailed in sections *E.* and *C.*.

E. Datasets

We use the Institute of Informatics and Applied Mathematics English handwriting dataset, which is known as the IAM dataset [21, 20], George Washington dataset, which is known as the GW dataset [1], and IFN/ENIT (Arabic) dataset [26, 2]. The multi-script Transcription (MLT) dataset [23] contains challenging scene-text images of eight different languages; English, German, French, Italian, Arabic, Bangla, Korean, and Japanese. Due to lack of space, we only present the Bangla alphabets in Fig. 4. We did not use Korean and Japanese due to annotation errors that we detected during our exploratory analysis on the MLT-2017 dataset. We also use Instagram images by annotating more than 600 scene-text images of the #handlettering hashtag (which contains handwriting, printed, calligraphy text, among others). IAM, GW, IFN, Instagram and MLT-2017 datasets have Arabic numerals and non-alphanumeric symbols (MLT also has Indian and Bangla numerals). All Latin letters are used in a case-insensitive manner.

To validate the word-spotting model prior to testing it with Instagram dataset, we generated synthetic scene handwritten dataset. We use scene images from the STL-10 dataset (which has $100,000$ images) [4] to form the background of the handwriting words taken from GW, IAM and IFN datasets. For the generation to be realistic, we randomly picked the STL-10 images for each iteration in the training and testing and we randomly changed the handwriting color in a set of experiments. To illustrate the scene-handwritten image generation, a few images are depicted in Fig. 5.

A final word on the datasets is that the Arabic language has 28 standard letters that change shape according to their position in the word; *i.e*. *beginning*, *middle*, *end* or *isolated* letters. We treat each letter as its representative form regardless of its position; meaning, every letter in Arabic is mapped to its representational letter code, also in accord with some previous works, *e.g*. [32].

Figure 5. Scene-handwritten examples: the word 'weight' from the GW and 'though' from the IAM datasets are overlaid on STL-10 images; last row shows image examples of the word 'though' after random coloring of the handwriting stroke. Best viewed in color.

IV. Results

We performed several experiments to measure the average performance of the proposed model. The GW dataset has around $5,000$ word images, the IFN dataset has around $30,000$ word images, and the IAM dataset has $47,981$ training, $7,554$ validation, and $20,304$ testing samples. The MLT-2017 dataset has a total of $68,613$ samples. We use 75% for training and the rest for testing for MLT-2017, IFN and GW datasets. We treat the IAM dataset similar to [30] by merging the training and validation sets and using the merged set for training. To cope with previous works, we removed the stop words from the query for the IAM dataset [16].

A. Script identification

We build the script identifier using ResNet-18. This model learned to identify English from Arabic script in a few epochs, reaching 99.9% classification accuracy for the handwriting datasets. ResNet-152 also reached an accuracy of 99.9%, but ResNet-18 has an advantage over ResNet-152 as it is smaller in size. We also explored using SqueezeNet, which is much smaller than ResNet-18, but did not converge. For MLT-2017 dataset (containing English, French, German, Italian, Bangla, and Arabic), a ResNet-18 script identifier achieved 65% accuracy, and the reasons are: 1) the alphabet overlap between the Latin languages and the numerals, 2) non-alphanumeric symbols overlap between the different languages and 3) the complex scene text images, indoors and outdoors, having sometimes rotation and perspective views.

B. Uni-script word spotting

Uni-script word-spotting denotes using a model to perform word spotting for each single language. Our proposed deep learning model, described in *C.*, achieves remarkable word-spotting performance, as illustrated in Tables 1 and 3. Clearly, our results are on par and sometimes better than those reported in previous state-of-the-art works.

C. One model based script-independent multilingual word-spotting

In this case, there is no need to use script identification model and will thus provide $n+1$ models savings when used with n languages. Hence, we tested our Script-Independent Multilingual wOrd-Spotting (SIMOS) model using several languages. In fact, we use the same deep learning model, described in *C.*, which we also use for the uni-script word spotting. That is, there is no need to do any alterations to the deep learning model to test if it has the capacity to learn multilingual scripts. In the first experiment, we calculate the word-spotting retrieval performance of the SIMOS model using GW, IFN, and IAM datasets. The results presented in Table 2 show that the SIMOS model is capable of yield mAP values slightly less but on par with the uni-script model tested separately for each language. We also train and test the SIMOS model on the MLT-2017 dataset that contains English, French, German, Italian, Arabic, and Bangla languages. The results in Table 3 demonstrate that the SIMOS model is capable of learning in this challenging and dynamic dataset.

We also aim to test the generalization ability of the SIMOS model on the Instagram dataset. As Instagram #handlettering images are quite challenging containing printed and handlettering words, we train a SIMOS model to capture both features using MLT-2017, IFN and IAM datasets. From the results presented in Table 3, we can see the the QbS and QbE values are way above chance, although less than those of the MLT, IFN, and GW datasets. One reason is that the Instagram handlettering data is highly challenging as it contains scene, text, handwriting, calligraphy, different texture backgrounds; in addition to computer generated indoor and outdoor images. The Instagram dataset is a novel dataset and no baseline yet exists for comparison. To have an idea about the complexity of the Instagram dataset, we perform a few experiments using the Google Vision API [11], which is based on reliable optical character recognition models, and our analysis confirms the high complexity of this dataset. For example, the simple image containing 'Apple' calligraphy shown at the bottom right part of Fig. 1 was identified as leaf by Google Vision API, yet a simpler image in the same figure containing the word 'KARNEVAL' was identified correctly.

D. Scene handwriting word spotting

We present in Table 2 the QbS and QbE values after training and testing the SIMOS model with the scene-handwriting dataset, described in *E.*. To investigate the ability of the SIMOS model to deal with colored handwriting, we used random coloring of the handwriting stroke, both during training and testing. Moreover, our handwriting coloring algorithm does not consider the overall color of the scene background, which is an additional challenge as people usually use bright colors to write on dark scenes and vice versa. For additional complexity, we made artistic effects by creating gaps in the handwriting strokes. The total number of the used handwriting color combinations is 256^3. The handwriting in most of these generated images is hard to recognize even by a human observer. The results using the SIMOS model in Table 2 are slightly lower than, if not on par with those of the normal handwriting images. The SIMOS model was able to capture the handwriting even when the background is a complex scene and the handwriting stroke is randomly colored.

V. Discussion

This work addressed the following research questions: 1) can deep learning simultaneously learn script-independent multilingual word-spotting 2) can it adapt to the increased dimensionality of the PHOC resulted from using those languages? 3) can it learn scene handwriting and/or scene-text words? and 4) what future adaptations or proposals are needed to address a larger number of languages towards a fully multi-script word-spotting model? To start, our proposed model has been successful in learning the multi-script word spotting, but less success in tri-script word-spotting. Using one model in such scenarios is more efficient than using three models to determine the type of script in addition to additional other models used to implement word-spotting for each script. Table 4 shows this; the total number of parameters needed for each word-spotting system, which clearly demonstrates the advantage of using only one model as the number of parameters increases linearly with the number of languages when using one word-spotting model a script identification model for each language. Clearly, increasing the dimension of the output layer, which depends on the increase in the PHOC dimension of the multi-script word-spotting model did not have a significant impact on its size.

Due to lack of space, we only present IAM/IFN/GW loss evolution, shown in Fig. 6, that gives a nice indication about learning the representation of different datasets and script/language scenarios. Clearly, the IFN (Arabic) dataset has a smoother curve compared to the GW and IAM datasets. This smoothness of the loss evolution of the IFN dataset also copes with the results, where the IFN gave the highest retrieval performance denoted by QbS and QbE. For the multi-script word spotting system, denoted in Fig. 6 by the IAM+IFN and GW+IFN titles, we can see that the IAM+IFN has a smoother curve compared to the GW+IFN. This is because of the poor digitization and thresholding for the GW word images, which is due to the low quality of George Washington letters dating back to the 18^{th} century.

The proposed script identification model, based on ResNet-18, achieved over 99% and 65% accuracy in the handwritten and scene-text images, respectively. We ought

Table 1: Mean Average Precision (mAP)[%] of the proposed uni-script word spotting model on standard datasets compared to state-of-the art methods. Except our method, all other methods used data augmentation of a very large volume which can reach 500,000 samples, either using affine deformation models or synthetic data.

Method	**IAM**		**GW**		**IFN**	
	QbS	**QbE**	**QbS**	**QbE**	**QbS**	**QbE**
Proposed	88.5	89.4	96.1	**98.6**	**99.0**	**99.0**
Deep-Embed [16]	94.0	90.3	97.4	98.1	–	–
PHOCResNet [30]	**94.6**	88.3	**98.4**	97.2	–	–
PHOCNet [32]	82.9	72.5	92.6	96.7	92.1	96.1

Table 2: Mean Average Precision (mAP)[%] of the proposed multi-script word spotting system in normal and scene handwritten.

Image Background	**Query Dataset**	**Training Dataset GW+IFN**	
		QbS	**QbE**
Black	GW+IFN	95.5	98.8
	GW	94.2[96.1]†	97.7[98.6]
	IFN	98.8[99.0]	98.9[99.0]
Scene	GW+IFN	95.6(92.1)*	99.0(98.4)
	GW	93.6(92.1)	97.9(97.3)
	IFN	99.0(98.7)	99.1(98.7)

		IAM+IFN	
		QbS	**QbE**
Black	IAM+IFN	91.1	94.6
	IAM	89.3[88.5]	90.0[89.4]
	IFN	98.9[99.0]	98.8[99.0]
Scene	IAM+IFN	90.4(87.3)	94.7(93.6)
	IAM	88.8(84.7)	90.1(88.0)
	IFN	98.7(98.9)	98.8(98.7)

†Between square brackets are the QbS and/or the QbE obtained via the uni-script model (these values are copied from the first row of Table 1). *Results obtained using ***randomly colored*** handwriting composited onto STL-10 scene dataset are shown between parentheses.

to mention that the script identification model should only be used for the multi model based case, *i.e*. one word-spotting model for each language as illustrated in Fig. 2, and it is not needed for the proposed (single) multi-script word-spotting model. To emphasize the advantages of the proposed SIMOS model when used for n languages, the word-spotting system that is based on multi models requires n times the size that the proposed multi-script (single) model based needs, as shown in Table 4.

Last but not least, MLT-2017 script identification (English, French, German, Italian, Bangla, and Arabic) achieved 65% accuracy. This indicates that the performance of the multi-script word-spotting built using a single model is much better than the a word-spotting built using script identifiers and several uni-script word-spotting models. This is because the 35% script identification error in estimating the script language will affect the retrieval performance by the same rate; compared to 86.5% QbS and 92.0% QbE obtained for the multi-script model of the six languages we used from the MLT-2017 dataset.

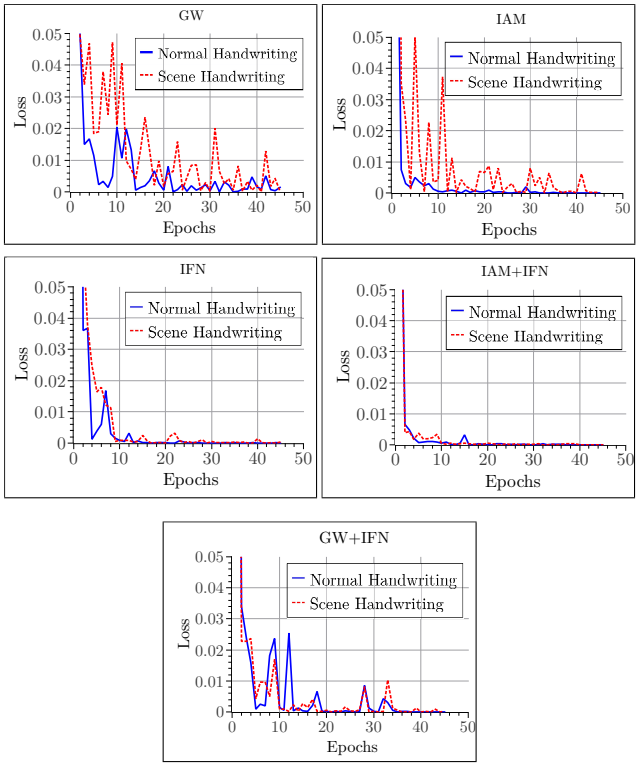

Figure 6. The loss evolution over the course of training.

VI. Conclusions

Despite the fact that data augmentation have not been used in this work, mAP results of the proposed word-spotting system are close to, and sometimes better than, those reported in other recent state-of-the-art works that are heavily

Table 3: Mean Average Precision (mAP)[%] of the proposed multi-script word-spotting system using MLT-2017 dataset.

Language/Model	QbS	QbE
English / uni-script	89.3	93.0
Arabic / uni-script	87.4	92.5
Bangla / uni-script	84.0	96.6
English+Arabic / bi-lingual bi-script	86.5	91.6
English+Arabic+Bangla / tri-lingual tri-script	83.7	90.8
English+French+German+Italian (Latin) / quad-lingual	88.0	91.9
English+French+German+Italian (Latin) + Arabic / quint-lingual bi-script	87.3	92.1
English+French+German+Italian (Latin) + Arabic + Bangla / sex-lingual tri-script‡	86.5	92.0
MLT(Latin) + IFN(Arabic) + GW-STL10(English)⋆; Model tested with Instagram Dataset †	53.2	67.6

‡ Compared to less than 57% mAP reported in [36], although they tested a statistical approach with far less keywords.
⋆ This is the only instance where IFN & GW used in training.
† Zero-shot learning for Instagram dataset.

Table 4: Number of trainable parameters of multi-script word-spotting systems. The one we propose uses only one ResNet-152, while the other system (which others might suggest to refute our one-model based system) uses several ResNet-152(s), one for each language, in addition to a third model based on ResNet-18 for script identification/separation.

Dataset(s)	Proposed Script-Independent Multilingual Model	Uni-Script Multi Model System
bi-lingual	60.5M	130.0M
tri-lingual	60.7M	190.2M
quad-lingual	60.9M	250.4M
quint-lingual	61.2M	310.7M
sex-lingual	61.4M	371.1M

dependent on data augmentation. Correspondingly, the proposed word-spotting model resulted better mAP results than the other works in the literature that did not consider using data augmentation. However, the greatest achievement of this work is that it has shown that the one deep learning model can be used efficiently and accurately in building script-independent multilingual word-spotting system that can reach the performance of each single model per script. Another achievement is the ability of the proposed word-spotting system to recognize handwriting words composited on scene images with performance close to the normal handwriting composited over flat background as normally appear in documents. The latter has been a real challenge due to the complexity of the scene handwriting used in this work that sometimes cannot easily be recognized by a human observer. We also verified that the proposed multi-script word-spotting system is resilient to the random temporal color changes, not to mention that the handwriting strokes in this experiment has random distortions in the forms of gaps. The multi-script scene-text and handwritten problems coupled with script-independent multilingual modelling is a challenging research area that can be further improved and we expect the scientific community to make non-trivial efforts on it. Ways of improvement may include using attention, augmentation, and hyper-tuning of the model parameters.

Acknowledgment

This work has received funding from from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 665919, research grants TIN2017-89779-P and 2017-SGR-1783 from the Spanish and Catalan governments, respectively.

References

[1] GW and IAM English handwriting datasets. http://www.fki.inf.unibe.ch/databases/. [Online; accessed: 25-June-2018]. 3, 4

[2] IFN/ENIT Arabic handwriting dataset. http://www.ifnenit.com/. [Online; accessed: 20-may-2018]. 3, 4

[3] Instagram's handlettering hashtag: Seven million images and increasing. https://www.instagram.com/explore/tags/handlettering/?hl=en. [Online; accessed 4-June-2018]. 1

[4] The STL-10 dataset. http://cs.stanford.edu/~acoates/stl10. [Online; accessed 2-May-2018]. 4

[5] J. Almazan, A. Gordo, A. Forns, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(12):2552–2566, 2014. 1, 3

[6] P. Barlas, D. Hebert, C. Chatelain, S. Adam, and T. Paquet. Language identification in document images. In *Document Recognition and Retrieval*, pages 1–16. Ingenta, 2016. 2

[7] Q. Decaillet. How to add handwritten text to your Instagram pictures. https://fstoppers.com/composite/how-add-handwritten-text-your-instagram/. [Online; accessed 3-June-2018]. 1

[8] J. Edwards, Y. W. Teh, D. A. Forsyth, R. Bock, M. Maire, and G. Vesom. Making latin manuscripts searchable using ghmms. In *NIPS*, pages 385–392, 2004. 3

[9] A. M. Elgammal and M. A. Ismail. Techniques for language identification for hybrid Arabic-English document images. In *ICDAR*, 2001. 2

[10] L. Gomez, M. Rusinol, and D. Karatzas. Lsde: Levenshtein space deep embedding for query-by-string word spotting. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 499–504. IEEE, 2017. 1

[11] Google. Google Vision API. https://www.nltk.org/. [Online; accessed 8-Jan-2019]. 2, 5

[12] L. Gulyas. Quickly add handwritten captions to your instagram photos. https://helpx.adobe.com/lightroom-cc/how-to/add-text-travel-photos.html. [Online; accessed 10-August-2018]. 1

[13] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Transactions on Image Processing*, 25:2529–2541, 2016. 2

[14] D. Hebert, P. Barlas, C. Chatelain, S. Adam, and T. Paquet. Writing type and language identification in heterogeneous and complex documents. In *ICFHR*, pages 411–416. IEEE Computer Society, 2014. 2

[15] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116:1–20, 2015. 1

[16] P. Krishnan, K. Dutta, and C. V. Jawahar. Word spotting and recognition using deep embedding. *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 1–6, 2018. 1, 4, 6

[17] P. Krishnan and C. V. Jawahar. Generating synthetic data for text recognition. *CoRR*, abs/1608.04224, 2016. 1

[18] G. Kumar and V. Govindaraju. A bayesian approach to script independent multilingual keyword spotting. *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 357–362, 2014. 2

[19] R. Manmatha, C. Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *CVPR*, 1996. 3

[20] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *In Proc. Int. Conf. on Document Analysis and Recognition*, pages 705–708, 1999. 4

[21] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 2002. 4

[22] D. Masters and C. Luschi. Revisiting small batch training for deep neural networks. *CoRR*, abs/1804.07612, 2018. 3

[23] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khlif, M. M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1454–1459, 2017. 2, 3, 4

[24] S. M. Obaidullah, C. Goswami, K. C. Santosh, C. Halder, N. Das, and K. Roy. Separating indic scripts with 'matra' - A precursor to script identification in multi-script documents. In *CVIP (1)*, volume 459 of *Advances in Intelligent Systems and Computing*, pages 205–214. Springer, 2016. 2

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 3

[26] M. Pechwitz, S. S. Maddouri, V. Morgner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. In *In Proc. of CIFED 2002*, pages 129–136, 2002. 3, 4

[27] A. Poznanski and L. Wolf. CNN-N-Gram for handwriting word recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2305–2314, 2016. 1

[28] PyTorch. Tensors and Dynamic neural networks in Python with strong GPU acceleration. https://pytorch.org/. [Online; accessed 24-April-2018]. 3

[29] K. Roy, A. Alaei, and U. Pal. Word-wise handwritten persian and roman script identification. In *ICFHR*, pages 628–633. IEEE Computer Society, 2010. 2

[30] E. Rusakov, S. Sudholt, F. Wolf, and G. A. Fink. Exploring architectures for cnn-based word spotting. *CoRR*, abs/1806.10866, 2018. 1, 3, 4, 6

[31] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger. Social media analytics - challenges in topic discovery, data collection, and data preparation. *Int J. Information Management*, 39:156–168, 2018. 1

[32] S. Sudholt and G. A. Fink. PHOCNet: A deep convolutional neural network for word spotting in handwritten documents. In *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*, pages 277–282. IEEE, 2016. 1, 3, 4, 6

[33] S. Sudholt and G. A. Fink. Evaluating word string embeddings and loss functions for cnn-based word spotting. In *ICDAR*, pages 493–498. IEEE, 2017. 1, 3

[34] S. Sudholt and G. A. Fink. Attribute CNNs for word spotting in handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 21:199–218, 2018. 1, 3

[35] K. Ubul, G. Tursun, A. Aysa, D. Impedovo, G. Pirlo, and T. Yibulayin. Script identification of multi-script documents: A survey. *IEEE Access*, 5:6546–6559, 2017. 2

[36] S. Wshah, G. Kumar, and V. Govindaraju. Statistical script independent word spotting in offline handwritten documents. *Pattern Recognition*, 47:1039–1050, 2014. 2, 7