Data Augmentation for Recognition of Handwritten Words and Lines using a CNN-LSTM Network

Curtis Wigington, Seth Stewart, Brian Davis, and Bill Barrett
Brigham Young University
wigington@byu.net, sstewar2@studentbody.byu.edu,
briandavis@byu.net, barrett@cs.byu.edu

Brian Price and Scott Cohen
Adobe Research
bprice@adobe.com, scohen@adobe.com

***Abstract*—We introduce two data augmentation and normalization techniques, which, used with a CNN-LSTM, significantly reduce Word Error Rate (WER) and Character Error Rate (CER) beyond best-reported results on handwriting recognition tasks. (1) We apply a novel profile normalization technique to both word and line images. (2) We augment existing text images using random perturbations on a regular grid. We apply our normalization and augmentation to both training and test images. Our approach achieves low WER and CER over hundreds of authors, multiple languages and a variety of collections written centuries apart. Image augmentation in this manner achieves state-of-the-art recognition accuracy on several popular handwritten word benchmarks.**

***Keywords*-Data Augmentation, Handwriting Recognition, Deep Learning, Elastic Distortion, CNN, LSTM**

I. Introduction

The need to transcribe archives of handwritten documents has accelerated the development of Deep Learning networks for automated handwriting recognition (HWR). As shown in Fig. 1, handwriting varies widely from author to author (row a) as well as from instance to instance for a single author (row b). While modern neural networks show good performance at HWR, available training data is often not sufficient to capture this variation.

As a result, we introduce a more robust augmentation technique to model the variation of a given author. This is illustrated in Fig. 1 where augmentation is performed by distorting the boxed instance 3 times (row c) with overlay in the 4th column. This overlay is comparable to the overlay in row b which shows the natural variation of the single author.

Recently, Convolutional Neural Networks (CNNs) have been shown to produce impressively low error rates for large, multi-author handwritten word datasets [1]. Such networks have made use of reduced feature representations with deep feature embedding and augmented training to perform word spotting as well as recognition [2]. Recurrent neural networks (RNNs) have also been applied successfully to HWR, producing top results in the recent competition on German handwriting recognition [3].

To improve the state-of-the-art in neural-network-based HWR, we introduce two novel data augmentation and normalization techniques that should allow any HWR neural network to improve generalization. We achieve very accurate recognition at both the word and line level with results that eclipse current best approaches at the word level: (1) profile normalization of both word and line images, and (2) distortion of existing words using random perturbations on a regular grid aligned to the baseline. We apply normalization and augmentation to both training and test images.

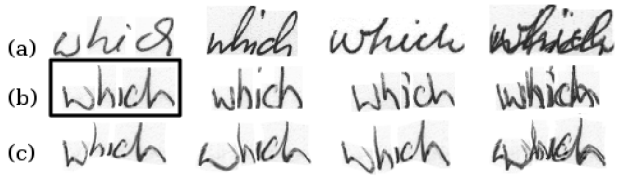


Fig. 1. (a) 3 different IAM authors + overlay (4th column) (b) Individual IAM author + overlay (c) 3 distortions of boxed instance using our approach. Overlaid distortions (row c) closely model natural variations in row b overlay.

We evaluate our augmentation and normalization techniques using a CNN-LSTM architecture [4] to perform HWR. Our choice of neural network architecture is motivated by simplicity and the flexibly to recognize on both word and line images.

We present the lowest Word Error Rates (WER) to date over thousands of authors and multiple languages written centuries apart. This includes the READ dataset consisting of historical German documents [3], and large multi-author datasets (IAM [5] and RIMES [6]).

II. Related Work

A. Handwriting Recognition

HWR is a long-standing computer vision problem [7], [8], [9]. More recently, deep learning approaches have yielded very low error rates for large, contemporary multi-author handwritten datasets.

Poznaski and Wolf [1] perform word-level recognition by employing a fixed-size CNN architecture that evaluates binary lexical attributes over word images, such as whether a given portion of the image contains a certain unigram or bigram (PHOC [10]). The correct transcription is determined by the word in a lexicon closest to this representation. Krishnan et al. [2] also employ a fixed-size CNN architecture to learn features for the PHOC representation for embedding the text and images into a common subspace.

Another general approach uses RNNs for HWR. These have been widely adopted with the introduction of CTC (connectionist temporal classification) [11], particularly using

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.110

the popular LSTM (long short-term memory) units. They are capable of being trained on, and thus recognize, a line of text without any other segmentation information (i.e. they do not require word-level segmentation), which makes them very appealing for the application of recognizing handwriting in documents. Doetsch et al. [12] use a 3-layer BLSTM (bidirectional LSTM) with PCA-based features. Bluche et al. [13] use four systems with ROVER [14]: deep multi-layer Perceptrons on handcrafted features, deep multi-layer Perceptrons on pixel values, BLSTM on handcrafted features, and BLSTM on pixels. Both systems have achieved state-of-the-art results for line level recognition on the IAM and RIMES Databases. In a recent competition on 14th through 18th century German handwriting [3] (Fig. 4), the top three methods used architectures generally consisting of convolutional layers and LSTM (bidirectional or multidirectional) layers [4], [15]. We follow this basic approach in the design of our evaluation network.

B. Data Augmentation

Deep learning networks typically require large amounts of data. However, for many datasets, especially historical documents, the data is fixed, so augmentation must be performed by modifying the original data.

For image recognition, augmentation is applied using simple transformations such as flipping images horizontally, scaling, or sampling subwindows of the images [16]. For handwriting, slight affine transformations are often used [1]. However, affine transformations over word images fail to capture variations of slant and size that occur at the character level.

Shen and Messina [17] use a corpus of segmented handwritten Chinese characters to create new line images for line recognition. This is done by concatenating characters, with variation in spacing and alignment, or replacing characters in a document image with new (normalized) characters. This augmentation method is effective, but requires a character-level handwritten dataset to build from.

Krishnan and Jawahar [18] present a method similar to [17] of pretraining a network using a synthesized dataset of cursive fonts rather than handwriting to synthesize word images. By varying inter-character spacing, stroke width, and foreground-background pixel distributions, they create a convincing synthetic dataset, used to train a network that is later fine-tuned on the real target dataset. This same synthetic dataset and methodology is applied in the previously discussed work [2]. However, this technique is only effective when the fonts can closely model the handwriting, which in the case of historical documents may not be possible. Fonts also fail to fully capture the wide variations in handwriting style such as those shown in Fig. 1.

Our augmentation technique builds upon, but is substantially different from that described in [19]. Simard et al. [19] show improvements over affine transformations by using random elastic distortions over single character images. This is done by creating a random displacement field followed by Gaussian smoothing. This technique was originally used for single character recognition on small 28x28 handwritten digit images, and, to our knowledge, has not been applied to word or line images.

The distortions of [19] are based on two parameters: σ and α. The authors give recommendations for σ and α values for 28x28 images. However, tuning σ and α is nonintuitive, and for our higher resolution images (variable width x80) we had to iterate over many possible values and select from the examples that looked the most plausible.

III. Augmentation and Normalization

We introduce novel methods for augmentation and normalization to improve HWR by allowing the network to be more tolerant of variations in handwriting. Normalization adjusts for differences in the scale of the handwriting. The augmentation models the natural character-to-character variation and improves the network's accuracy and ability to learn.

A. Profile Normalization

Images are normalized to compensate for variations in the size of the handwriting. We normalize word images using the difference d between the upper and lower baseline provided in the IAM Database (Fig. 2) and the standard deviation σ of their horizontal profiles. We define ratio r as

$$r = \frac{\bar{d}}{\bar{\sigma}} = \frac{1}{|A|} \sum_{i=1}^{|A|} \frac{\bar{d}_i}{\bar{\sigma}_i} \tag{1}$$

where $\bar{d}_i$ is the average baseline difference for author i, $\bar{\sigma}_i$ is the average standard deviation of the horizontal profile, and A is the set of authors. $r \approx 1.75$ for the IAM Database.

We normalize all images by a scale factor $s = \frac{16}{\bar{\sigma}_i r}$ since $\bar{d}_i \approx 16$ pixels for most of the authors. Figure 2 shows words from two authors with their respective horizontal profiles in blue. Using s, the top images are scaled to a roughly equivalent size.

Even if author identifiers are not labeled, in many cases same authorship can be easily inferred (i.e. same sentence, same page, etc.). In RIMES, authors are not labeled, but we knew that each page contained only handwriting from a single author, so we treated each page as a unique author even though the same author may have written multiple pages. In addition, since no lower and upper baselines are provided for the RIMES Database, these are also scaled using r from IAM. After height normalization, images are then centered according to the center of mass.

B. Novel Grid-Based Distortion Augmentation

Previous methods have sheared or rotated an entire word image to generate augmented data [1]. However, we observe that naturally occurring variations in handwriting are *not* usually manifested as uniform slants across the entire word (an affine transformation), but more as slight differences in scale and slant from character to character within the word. We employ a random grid mesh to capture this.

Random warp grid distortion (RWGD) is performed as follows. (1) Place control points on a regular grid such that

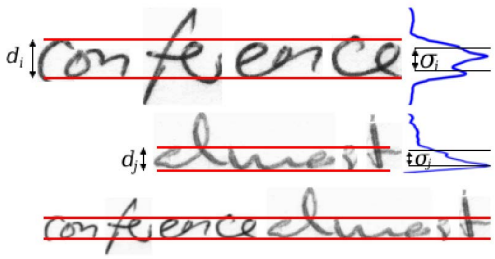


Fig. 2. Image profiles (top, middle) and resulting normalization (bottom) from baseline b and and standard deviation σ.

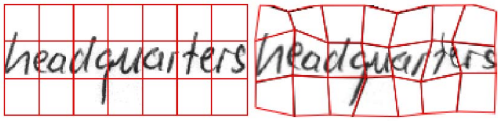


Fig. 3. Word image with uniform grid superimposed. 2nd image (right) with distorted grid and image distorted accordingly.

they align to the baseline. For IAM/RIMES we used a 26-pixel interval. (2) Perturb each control point in the x and y direction by randomly sampling from a normal distribution. For IAM/RIMES we used a standard deviation of 1.7 pixels. (3) Warp the image according to the perturbed control points.

Given the profile normalization and center of mass alignment, the middle row of the grid approximates the height of the lowercase characters. This allows the distortion grid to apply the warp over entire characters, minimizing the creation of kinks or creases within a character, thereby creating a more natural looking distortion. Figure 3 shows a word image being distorted according to the random grid.

Our technique is based on two parameters, the control points placement interval and standard deviation by which to randomly displace the control points. In our experiments, we found that adjusting these parameters was intuitive to visualize and tune for the specific handwriting sets. We place the control points on intervals of 26 pixels (slightly larger than the average baseline height) and perturbed the points about a normal distribution with a standard deviation of 1.7 pixels. These parameters are for images with a height of 80 pixels. Both values scale linearly with the height of image and would need to be scaled if different sized images were used.

C. Test-side Augmentation

Similar to Poznanski and Wolf [1], we employ test-side geometric augmentation using our described techniques. Test-side augmentation is performed by (1) generating N augmented examples for each word/line image in the test set (N=20 in our experiments), (2) performing recognition on the N augmented images, and (3) choosing from the N predictions the one that produces the lowest CTC loss based on a lexicon. If using lexicon-free decoding, because there is not an associated loss, we select the most commonly occurring prediction. In contrast to [1], our network uses recurrent layers, allowing us to process line images of arbitrary length. Therefore, instead of averaging feature vectors prior to classification, we use the predictions for each image variant as described in step 3.

IV. The CNN-LSTM Network

Our CNN-LSTM network, based on [4], uses 6 convolutional layers: 64, 128, 256, 256, 512, and 512 (3x3) filters respectively in the forward direction. Batch normalization is applied after the 4th and 5th layers. Max pooling (2x2 window), stride 2 in both directions are applied after the 1st and 2nd layers. Max pooling (2x2 window) and vertical stride of 2 and horizontal stride of 1 is applied after the 4th and 6th layer. Two BLSTM layers follow with 512 and 256 hidden nodes respectively with dropout rate of 0.5 before each. A fully connected layer reduces the output to the character set size and a softmax is applied. It is trained using the CTC loss and the ADADELTA optimizer.

Our CNN has $w \times h$ input nodes where h is a fixed pixel height dependent on the dataset (German: 60, RIMES & IAM: 80) and w is the corresponding width where the aspect ratio of the image is preserved. We use a single input channel (grayscale image) with the exception of the German dataset where we use two additional binarizations as input channels: the thresholding scheme specified in [3] and Howe's binarization [20].

V. Results and Discussion

We first present results of our raw network output with and without the use of a lexicon (Table I). This provides a baseline for comparing our results with prior work and the improvement obtained using augmentation and lexical correction.

To demonstrate the effectiveness of our augmentation and normalization strategies, we compare the results of our network with and without augmentation and normalization (Table III), and how these results compare with those obtained using more traditional augmentation methods (Table IV).

We also compare our results with prior state-of-the-art methods (Table V). We evaluate over datasets that vary in language, authorship, content, and general appearance. These include large multi-author datasets (IAM and RIMES) and historical 14th through 18th century German documents, first featured at ICFHR 2016. Depending on the dataset, we appropriately report results at word-level and/or line-level.

Because we have found some variation in the evaluation methods used previously, when comparing our results with prior work, we take careful consideration to report parameters such as case sensitivity, inclusion of punctuation, what lexicon was used, and what portion of the testing set was actually used. We do this with the intention of making our comparisons as clear, fair and accurate as possible.

A. Lexicon and Lexicon-free Network Decoding

The top line of Table I gives the results of our network output without the use of a lexicon (Lexicon Free Decoding). With the use of our augmentation techniques (training and test augmentation) and without lexical correction, our network yields a WER/CER of 19.07/6.07 for the IAM dataset and

Method	IAM		RIMES	
	WER	CER	WER	CER
Lexicon Free	19.07	6.07	11.29	3.09
Lexicon	5.71	3.03	2.85	1.36

TABLE I
LEXICON FREE DECODING VS LEXICON-BASED DECODING

11.29/3.09 for RIMES. Line 2 of Table I shows results from augmentation and lexical decoding.

For both lexicon (LD) and lexicon-free decoding (LFD) of the network, we employ the same method described in [4]. For lexicon-free decoding, at each time step we select the character with the highest activation. The lexicon-free decoding applies to both line-level and word-level recognition. We apply lexicon decoding to word-level but not line-level recognition. We find the word in the lexicon that produces the lowest CTC loss for the output of the network. We prune the lexicon to words within a certain edit distance of the lexicon-free decoding to avoid computing the CTC loss for all words in the lexicon.

Most previous work compares results after applying some form of lexical correction. However, we find it useful to compare network output before applying the lexicon, because variations in the size and the out-of-vocabulary (OOV) rate of the lexicon can significantly affect performance.

B. Qualitative Results

Table II shows an example (IAM) word image after various augmentation techniques have been applied. The red grid is to help visualize the effects of the augmentation. Profile normalization has been applied to all of the images. The images in the right-hand column show five overlaid instances of the specified augmentation techniquesimilar to the three overlaid instances shown in Figure 1 (4th column) where we compare our technique to the natural variation in handwriting.

Figure 4 shows a comparison of each augmentation technique applied to a noisy historical document. The first example in each group shows one distortion of the original on the top row. The second shows five similar distortions overlaid, as in Table II. In these examples the augmentation looks even more convincing because the handwriting already has significant variation. The last image in Figure 4 shows that our method produces more variation than the other techniques, while the single image is still a plausible exemplar. We believe the reason we have the lowest WER across all datasets is because our augmentation more closely models and accounts for the natural variation in handwriting from instance to instance and from author to author.

Affine and rotation generates variation for the ascender and descender parts of the characters, but generates minimal variation along the baseline of the characters, and thus fails to model the more natural within-character variation that typically occurs in handwriting. In our experiments we used $\pm 5^\circ$ for shear and rotation.

Given our parameterization, [19] produces localized distortions. However, we found that σ and α were difficult to tune and discovered these values ($\sigma = 8$ and $\alpha = 64$)

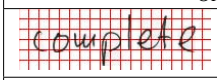
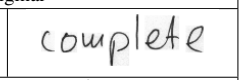
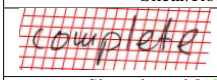
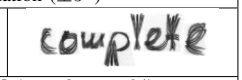
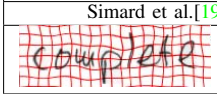
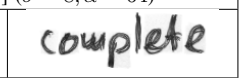
Single Example	Five Overlaid Examples
Original	
	
Shear/Rotation ($\pm 5^\circ$)	
	
Simard et al.[19] ($\sigma = 8, \alpha = 64$)	
	
Ours	
	

TABLE II
QUALITATIVE EXAMPLES OF THE AUGMENTATION TECHNIQUES.

only by iterating through many possible parameterizations and selecting from those that looked most plausible. In addition, its use up to this point has been limited to single digit images and its ability to produce natural variation in the sizes of the characters and the inter-character spacing is limited.

In contrast, our technique uses a warp grid where the control points align with the height of the baseline characters, producing more natural variation in the sizes of the characters and inter-character spacing. Also, only a single parameter needs to be tuned: the standard deviation for perturbation of the control points. This parameter is in units of pixels so it is simple to conceptualize and select.

C. Ablation Study: Elastic Distortion; Profile Normalization

Table III contains ablation results on the IAM and RIMES Databases, showing network performance without and with varying amounts of augmentation and normalization. All results in Table III make use of lexical correction.

The best results are obtained when Random Warp Grid Distortion (RWGD) and Profile Normalization (PN) are applied to both training and test images. With RWGD and PN, WER/CER drop to 5.71/3.03 for the IAM dataset and 2.85/1.36 for RIMES.

As can be seen, the drop in error rates is not strictly monotonic. For example, if RWGD is applied to the Training Set, but not the Test Set, the error increases slightly for RIMES. Similarly, the error increases if PN is used without RWGD.

However, marked improvements are achieved when RWGD and PN are used together. The unique combination of mesh-based elastic distortion and baseline centering using profile normalization achieves state-of-the-art results on these and other datasets. For both IAM and RIMES, WER and CER drop by about 40%. For IAM, WER decreases from 9.27 to 5.71, and CER from 5.14 to 3.03. For RIMES, WER drops from 4.98 to 2.85 and CER from 2.38 to 1.36.

Fig. 4. Augmentation examples for German handwriting. Top: single example. Bottom: five overlaid examples. Ground truth transcription: "Zū Abfertig: vnd Contentier"

RWGD		PN	IAM		RIMES	
Train	Test		WER	CER	WER	CER
			9.27	5.14	4.98	2.38
✓			7.88	4.39	5.12	2.43
✓	✓		6.09	3.33	4.72	2.23
		✓	9.87	5.35	7.53	3.69
✓		✓	7.18	3.93	3.84	1.82
✓	✓	✓	**5.71**	**3.03**	**2.85**	**1.36**

TABLE III
ABLATION STUDY (WITHOUT PUNCTUATION AND CASE). RWGD = RANDOM WARP GRID DISTORTION, PN = PROFILE NORMALIZATION.

D. Random Warp Grid Distortion Compared with Other Geometric Augmentation Techniques

For the IAM and RIMES Databases, we compare our novel random warp grid distortion augmentations (RWGD) to 36 affine transformations used in the work of [1], where predetermined slight rotation and shear operations are applied to word images. We also compare our technique to the elastic distortion by Simard et. al. [19]. Profile normalization is used for all techniques to facilitate comparison. Our technique demonstrates a 12-25% improvement over the next best approach in Table IV, even after our PN has been used with those techniques. We believe this is because our RWGD more closely models the natural character-to-character variation we see in handwriting.

E. IAM Handwriting Database

The IAM Handwriting Database [5] is a multi-author handwriting recognition database of 115,320 word-level images from 500 authors. The database provides a standard split for training, validation, and test sets. The data sets are mutually exclusive with regard to the authors; each author contributes to only one set. There are two tasks associated with this dataset: word-level recognition and line-level recognition. We only report results for word-level recognition.

Method	Train	Test	IAM		RIMES	
			WER	CER	WER	CER
None			9.87	5.35	7.53	3.69
Rotate/Shear	✓		7.63	4.16	5.09	2.25
Rotate/Shear	✓	✓	6.71	3.56	3.92	1.97
Simard et al.[19]	✓		7.64	4.11	4.03	1.85
Simard et al.[19]	✓	✓	6.57	3.45	3.78	1.66
Ours	✓		7.18	3.93	3.84	1.82
Ours	✓	✓	**5.71**	**3.03**	**2.85**	**1.36**

TABLE IV
COMPARISON OF AUGMENTATION METHODS. ALL EVALUATIONS USED PROFILE NORMALIZATION.

The IAM test set consists of 17,614 word images. 3,863 of these have ground truth or segmentation errors, so we discard these from the test set, reducing the test set to 13,751 words. Discarding images with only punctuation further reduces the test set to 11,601 words. This test set reduction is consistent with prior work.

For word-level recognition, previous work is generally lexicon-based. In some cases the lexicon is made up of words from the training and test sets, while in other work, the lexicon contains only words from the test lexicon. When punctuation is not evaluated, it is removed from the ground truth of word images that contain a combination of alphanumeric characters and punctuation.

1) IAM Word-level Recognition: Our network is trained to recognize punctuation and capitalization, even when not considered during evaluation. We train on all of the word images, even the ones marked with segmentation errors. Every word on a line is marked with an error if a single word on the line has an incorrect segmentation, even if most of the words are segmented correctly. Our system is robust to these errors and benefits from the additional training data. Previous work reveals three variations of evaluations on the IAM word-level recognition task. Table V is divided into three sections (Top, Middle, Bottom) to compare our results with different evaluation methods and lexicons. **Top:** Punctuation and case are considered in the evaluation. However, images that contain only punctuation are discarded. The lexicon contains words from the training and test sets. **Middle:** Punctuation and case are not considered in the evaluation. The lexicon contains words from the training and test sets. **Bottom:** Punctuation and case are not considered in the evaluation. The lexicon contains words only from the test set. State-of-the-art results are shown in bold.

In Table V, Top, we demonstrate a significant improvement (.72) in WER compared to [21]. However, our CER is higher by .16. This can mostly be explained by the difference in technique for applying the lexicon. [21] will reject a word if it is not a close enough match to a word in the lexicon and then apply an alternative decoding. Our approach, and the other's in the results tables, always selects a word from the lexicon.

With Punctuation & Case, Training/Test Set Lexicon		
Method	WER	CER
Bruno et al.[21]	6.55	**2.99**
Ours	**5.83**	3.15
Without Punctuation & Case, Training/Test Set Lexicon		
Method	WER	CER
Poznanski and Wolf[1]	6.45	3.44
Ours	**5.71**	**3.03**
Without Punctuation & Case, Test Set Lexicon		
Method	WER	CER
Krishnan et al. [2]	6.69	3.72
Ours	**4.97**	**2.82**

TABLE V
IAM DATABASE: WORD-LEVEL RECOGNITION

With Punctuation & Case, Competition Lexicon		
Method	WER	CER
Menasri et al.[22]	4.75	-
Ours	**3.69**	**1.69**
Without Punctuation & Case, Competition Lexicon		
Method	WER	CER
Poznanski and Wolf[1]	3.90	1.90
Bruno et al.[21]	3.48	**1.34**
Ours	**2.85**	1.36

TABLE VI
RIMES DATABASE: WORD-LEVEL RECOGNITION

As such, our technique favors a low WER where [21] favors a low CER.

When punctuation and case are not considered (Table V, Middle and Bottom), compared to previous state-of-the-art results, our approach yields significant improvement in both WER and CER, whether the lexicon is comprised of both the test set and the training set (Middle) or only the test set (Bottom).

F. The RIMES Database

The 2011 version of the RIMES Database [6] has over 60,000 French words with over 1,300 authors. A 5,744 word lexicon is used for word-level recognition. In the official competition, capitalization and punctuation were considered in the WER. CER was not computed in the official competition.

Results on RIMES (Table VI) are divided into 2 sections: where punctuation and upper/lower case is considered and where it is not. The lexicon originally provided by the competition is used in both sections.

Section 2 of Table VI provides results when punctuation and case is not considered. We include the results from [1] in this section, assuming they do not consider punctuation and case, as in Table V. CER with our approach is about the same as that reported in [21], even though [21] favors a low CER over WER, as noted in Table V, above.

G. 14th to 18th Century German

This dataset, consisting of 400 pages [3] of handwriting from German authors between 1470 and 1805, is the most challenging dataset because of its antiquity, flourishing writing style, archaic vocabulary, and significant ink bleed-through.

The training set consists of 350 pages, 8,367 lines, and 35,169 running words with a lexicon of 6,985 words. The test set consists of 50 pages, 1,043 lines, and 3,994 running words with a lexicon of 1,526 words.

Method	WER	CER	Average
RWTH	20.9	**4.8**	12.85
BYU	21.1	5.1	13.10
A2IA	22.1	5.4	13.75
LITIS	26.1	7.3	16.70
Ours	**19.7**	5.0	**12.35**

TABLE VII
GERMAN DATABASE: LINE-LEVEL RECOGNITION

The results of the competition are summarized in Table VII where there is a third column that represents the average of the WER and the CER. The winner of the competition was designated as the group that had the lowest average.

Our CNN-LSTM network with image augmentation and normalization produced a WER of 19.7, 1.2 less than the winner of the competition. Our CER was still .2 greater than the winner. However, our average (12.35) is 0.5 less than the winner. We believe that using a German language model (as did the winner) could reduce this error even further.

VI. CONCLUSION

We have introduced two new data augmentation and normalization techniques and have demonstrated their use with a CNN-LSTM to produce the lowest word error rate (WER) to date over hundreds of authors, multiple languages, and thousands of documents including challenging, medieval, historical documents with noise, ink bleed-through, and faint handwriting. Because these techniques are independent of the network used, they could also be applied to enhance the performance of other networks and approaches to HWR.

REFERENCES

[1] A. Poznanski and L. Wolf, "Cnn-n-gram for handwriting word recognition," *Proc. CVPR*, 2016.

[2] P. Krishnan, K. Dutta, and C. V. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," *The 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016.

[3] J. A. Sanchez, V. Romero, A. H. Toselli, and E. Vidal, "Icfhr2016 competition on handwritten text recognition on the read dataset," *ICFHR*, 2016.

[4] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *CoRR*, vol. abs/1507.05717, 2015. [Online]. Available: http://arxiv.org/abs/1507.05717

[5] 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, no. 1, pp. 39–46, 2002. [Online]. Available: http://dx.doi.org/10.1007/s100320200071

[6] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, and E. Geoffrois, "Rimes evaluation campaign for handwritten mail processing," *Workshop on Frontiers in Handwriting Recognition*, 2006.

[7] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.824821

[8] S. N. Srihari, "Recognition of handwritten and machine-printed text for postal address interpretation," *Pattern Recogn. Lett.*, vol. 14, no. 4, pp. 291–302, Apr. 1993. [Online]. Available: http://dx.doi.org/10.1016/0167-8655(93)90095-U

[9] S. N. Srihari and K. Singer, "Role of automation in the examination of handwritten items," in *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition*, ser. ICFHR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 619–624. [Online]. Available: http://dx.doi.org/10.1109/ICFHR.2012.263

[10] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[11] A. Graves, S. FernÃąndez, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[12] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, Sept 2014, pp. 279–284.

[13] T. Bluche, H. Ney, and C. Kermorvant, *A Comparison of Sequence-Trained Deep Neural Networks and Recurrent Neural Networks Optical Modeling for Handwriting Recognition*. Cham: Springer International Publishing, 2014, pp. 199–210. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11397-5_15

[14] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, Dec 1997, pp. 347–354.

[15] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," *The 14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014.

[16] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *Proc. BMVC*, 2014.

[17] X. Shen and R. Messina, "A method of synthesizing handwritten chinese images for data augmentation," *The 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 114–119, 2016.

[18] P. Krishnan and C. V. Jawahar, "Matching handwritten document images," *The 14th European Conference on Computer Vision (ECCV)*, 2016.

[19] J. P. Patrice Y. Simard, Dave Steinkraus, "Best practices for convolutional neural networks applied to visual document analysis." Institute of Electrical and Electronics Engineers, Inc., August 2003.

[20] N. R. Howe, "Document binarization with automatic parameter tuning," *International Journal on Documant Analysis and Recognition*, 2012, to appear; DOI: 10.1007/s10032-012-0192-x.

[21] B. Stuner, C. Chatelain, and T. Paquet, "Cohort of lstm and lexicon verification for handwriting recognition with gigantic lexicon," *arXiv preprint arXiv:1612.07528*, 2016.

[22] E. Grosicki and H. El-Abed, "ICDAR 2011: French handwriting recognition competition," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2011, pp. 1459–1463.