Language Model Supervision for Handwriting Recognition Model Adaptation

Chris Tensmeyer*,†‡, Curtis Wigington*,†‡, Brian Davis†, Seth Stewart†, Tony Martinez† and William Barrett†
†Dept. of Computer Science - Brigham Young University. Provo, USA
‡Adobe Research - San Jose, CA
tensmeyer@byu.edu

Abstract—Not all languages and domains of handwriting have large labeled datasets available for training handwriting recognition (HWR) models. One way to address this problem is to leverage high resource languages to help train models for low resource languages. In this work, we adapt HWR models trained on a source language to a target language that uses the same writing script. We do so using only labeled data in the source language, unlabeled data in the target language, and a language model in the target language. The language model is used to produce target transcriptions to allow regular example based training. Using this approach we demonstrate improved transferability among French, English, and Spanish languages using both historical and modern handwriting datasets.

Keywords—Handwriting Recognition; Language Model; Transfer Learning; Bootstrap; Historical Document Analysis

I. INTRODUCTION

State-of-the-art offline handwriting recognition (HWR) models are based on deep Convolutional Neural Networks (CNNs) and Bidirectional Long-Short Term Memory (BLSTM) networks and are trained on large amounts of labeled line images [1]. Obtaining such large annotated training sets is expensive and time consuming, though viewed as necessary because trained HWR models often fail to sufficiently generalize to domains, languages, and writers that were not observed during training. Eliminating or lessening this requirement is the goal of unsupervised HWR and related approaches.

Prior work has attempted to address the lack of a large labeled training set for low resource languages/domains through several means. Training on synthetic data is an appealing direction because an arbitrary amount of labeled data may be generated with little human effort. In some works, synthetic data is obtained by applying annotation preserving transformations to real data in order to simulate the natural variability in handwriting [2]–[4]. However, these methods depend on the availability of some labeled data, which is not always the case. Other works have modeled the writing process for isolated characters using character prototypes for Chinese [5] and Korean [6] characters, though it is not clear how such models could be extended to cursive scripts. Elarian et al. proposed a concatenative model for handwritten Arabic, though it relies on a database of pre-segmented characters and the concatenation procedure is specific to Arabic [7].

An alternative semi-supervised formulation of the problem assumes that there is a small labeled training set and a larger unlabeled training set. The main methodology involves propagating annotations from the labeled set to the unlabeled set through model prediction. Subsequent models then train as if the noisy predicted labels were ground truth annotations. Frinken et al. explored this method for isolated word image recognition in the framework of co-training, where Hidden Markov Models (HMM) and BLSTM models made predictions that were used to further train the other model [8]. In separate work, Frinken and Bunke use an ensemble of BLSTM networks for self-training, where high confidence ensemble predictions on the unlabeled data are subsequently used as ground truth to further train the ensemble. Ball and Srihari used a similar idea to adapt writer specific HWR models from a general model by iterative updating segmented character prototypes after performing recognition on unlabeled data [9].

In this work, we propose a transfer learning methodology that allows us to train a HWR model for a target language for which we have no labeled images. Our method only requires a labeled training set of line image in a sufficiently similar source language, a trained Language Model (LM) in the target language, and a set of unlabeled images in the target language. A source language is sufficiently similar to the target language if the character set of the two languages has high overlap. For example, Latin based languages are all similar enough because they all use the written Latin script. The LM can be obtained from digital text in the target language that is unrelated to the unlabeled images. Digital text for training a LM is much more commonly available than labeled line images and so our methodology helps extend automated HWR to lower resource languages.

After training a HWR on the source language, our proposed method begins a hybrid training procedure that mixes training on source data and target data. When the model encounters target data, it predicts a transcription which is fed to the LM to produce a corrected prediction, which we then use as ground truth in training the model.

We perform several experiments to analyze the behavior of our proposed transfer learning methodology for HWR. These experiments are performed using 4 datasets and
3 languages: English, Spanish, and French. We examine factors such as how long the source model is trained, LM decoding parameters, and what percentage of target data is used during hybrid training.

II. LANGUAGE TRANSFER LEARNING

We formulate our problem as follows. Suppose we wish to obtain a trained HWR model for a target language Y that has no labeled training data available, but there are many unlabeled text line images in this language. We do, however, have segmented text line images with corresponding ground truth transcriptions for a language X. Noting that languages X and Y have similar scripts, we want to use the data in both languages to produce the HWR model for language Y. We also suppose that we have sufficient digital text in language Y, such that we can train a Language Model (LM) for Y.

We also suppose that we have sufficient digital text in language X and Y have similar scripts, we want to use the data in both languages to produce the HWR model for language Y. We also suppose that we have sufficient digital text in language Y, such that we can train a Language Model (LM) for Y.

Though our discussion uses the term language, our methodology is also applicable to transfer learning problems where there is a difference in domains (e.g. modern vs historical) or writers. We demonstrate this later by transferring between two English datasets: one many-author modern dataset and one single-author historical datasets.

A. Source Model Training

We begin transfer learning by training a state-of-the-art HWR model on the source language for which we have ground truth transcriptions. We use a CNN-BLSTM architecture that learns high level features using convolution operations that are vertically collapsed to a 1D horizontal sequence of feature vectors that are fed to a 2-layer BLSTM. In the BLSTM, context is propagated both forwards and backwards along the sequence. Separate linear character classifiers are each applied frame-wise to the output of the CNN and the output of the combined CNN-BLSTM. Both classifiers are trained using Connectionist Temporal Classification (CTC) loss [10] which automatically aligns frame-wise outputs with the ground truth transcriptions.

The CNN classifier is considered an auxiliary classifier and it is discarded after the training procedure, meaning that the model predictions only consider the output of the CNN-BLSTM classifier. We found that using the auxiliary classifier improves transferability of the model, likely because it forces the CNN visual features to be discriminative. When transferring between languages, the visual difference of some shared characters is small, so the CNN should be robust to the language difference. In contrast, the BLSTM considers the whole sequence, so it is more sensitive to transferring between datasets.

The precise architecture of our HWR model is based on the model presented in [4]. The size of the the input image is $W \times 60$, where $W$ is the image width, which can dynamically vary. The CNN is composed of 6 convolutional layers with 3x3 learnable kernels, and there are 64, 128, 256, 256, 512, and 512 feature maps respectively for the 6 layers. We apply Batch Norm (BN) after layers 4 and 5, and 2x2 Max-Pooling (MP) with 2x2 stride is applied after layers 1 and 2. After layers 4 and 6, we vertically collapse features by using 2x2 MP with a vertical stride of 2 and a horizontal stride of 1. To form the input for the BLSTM and for the CNN auxiliary classifier, we concatenate features in the same column to form a 1D horizontal sequence of 1024-dimensional feature vectors. The BLSTM has 2-layers each with 512 hidden nodes that have a 0.5 probability of node dropout. A linear classifier is applied to each time step to produce the final prediction, which is a probability distribution over characters at each timestep.

The model is trained using CTC loss over both the main classifier and the auxiliary classifier:

$$L(X; Y) = \sum_{i} \lambda L_{CTC}(\phi(X_i), Y_i) + (1 - \lambda) L_{CTC}(\psi(X_i), Y_i)$$

where $X$ represents the input images, $Y$, the corresponding ground truth transcriptions, $L_{CTC}$ is the CTC loss [10], $\phi$ is the auxiliary CNN classifier, and $\psi$ is the BLSTM classifier. We empirically set $\lambda = 0.25$ based on cross validation using validation data.

B. Language Model Decoding

The HWR model produces a sequence of probability distributions over characters, but CTC decoding may produce linguistically improbable sequences of characters. Decoding with a Language Model (LM) instead of CTC during inference combines the individual predicted character probabilities with language specific knowledge of how probable sequences of characters occur in that language.

Our LM decoding implementation uses the Kaldi Speech Recognition Toolkit [11], which has been used in previous HWR models [12]. Similar to [13], we use a 10-gram character LM, which estimates $p(c_i|c_{i-1}c_{i-2} \ldots c_{i-9})$ from digital text. Because not all 10-gram character sequences are observed, we smooth the empirical 10-gram sequence distribution and employs backoff, where n-grams shorter than 10 are used to estimate the probability of infrequent 10-grams [14].

Decoding is finding the most likely sequence of hidden states in a Hidden Markov Model (HMM), where the emission probabilities are determined by the HWR model and the transition probabilities are determined by the LM:

$$\hat{h} = \text{arg max}_h \prod_{i}^{N} p(h_i|h_{j<i})p(x_i|h_i)^{w}$$

where $h$ is the sequence of hidden states corresponding to characters, $h_{j<i}$ indicates all states prior to $h_i$, $x$ is the observed data, and $w$ determines the relative importance of the CNN-BLSTM and LM predictions. Because characters can span multiple output frames, we model each character
using 3 states (corresponding to character start, middle, and end) as is commonly done in speech recognition [15]. The LM directly encodes the \( p(h_i|x_i) \) term, but the CNN-BLSTM outputs \( p(h_i|x_i) \). Using Bayes Rule, we have

\[
p(x_i|h_i) = \frac{p(h_i|x_i)p(x_i)}{p(h_i)} \tag{3}
\]

We can estimate \( p(h_i) \) by examining the CNN-BLSTM outputs, but \( p(x_i) \) is unknown. Following Bluche et al., we approximate \( \frac{p(x_i)}{p(h_i)} = p(h_i)^{-\alpha} \), where \( \alpha \) is a hyperparameter [12].

An exact solution to Eq. 2 can be intractably slow to compute, so in practice, we use a beam search which efficiently searches the state-space, but in some cases may not find the exact maximal sequence of characters.

C. Hybrid Training

Our hybrid training procedure leverages the recognition performance achieved by the source model on the source language to then learn recognition over the target language. The overall process is shown in Algorithm 1.

The main difference between hybrid and source training is in the data used for learning. During hybrid training part of the data comes from the source dataset (typically 50%) with the rest coming from the target dataset for which there are no ground truth transcriptions. However, the training loss for hybrid training is that same as in source training (Eq. 1), which means that to train we need to provide some target transcriptions. We obtain target transcriptions by applying the LM of the target language to the predictions of the network. The intuition is that due to the similarity of the source and target languages, the predictions of the network will be much better than random, though still quite poor at first. Applying the LM will improve the poor predictions to make better targets, which in turn helps the network to learn the target language better.

Because LM decoding depends on the marginal distribution of CNN-BLSTM outputs, \( p(h_i) \) in Eq. 3, we need to periodically update this quantity. This is done in lines 3-7 in Algorithm 1. In normal HWR model training, this is unnecessary because the LM is applied only as post-processing and not as part of the training process.

III. DATASETS

In this work we use 4 datasets: IAM [16], Rimes [17], Rodrigo [18], and Bentham (2014 HTR competition) [19] collections. Each dataset is composed of a number of line images with corresponding ground truth transcriptions.

Rodrigo is a single author, 853 page Spanish manuscript written in 1545 with 20000 segmented line images. We used the first 750 pages as training data, the next 50 pages as validation data, and the remaining pages as test data. The annotations contain some meta information that we preprocessed to exclude. Some examples of this include symbols that indicate that whitespace should be inserted or deleted for correctness, i.e., the manuscript author did not conform to modern usage of whitespace.

The Bentham collection are the writings of the English philosopher Jeremy Bentham, though some images may be handwritten copies of his works produced by others [19]. For preprocessing, we deskewed the line images and performed height normalization.

For IAM, we use the standard split, merging the two defined validation sets. For Rimes, there is only a defined train/test split, so we used a subset of the training data for a validation set.

Each image collection has different low level differences (e.g., color, texture), so we opted to binarize each dataset so eliminate those differences. This allows our analysis to focus on adapting to salient differences in languages and style rather than on adapting to low level domain differences. For IAM, Rimes, and Rodrigo, we used Otsu binarization [20] but for Bentham, we used adaptive Wolf binarization [21] because it produced visually better binarizations for this dataset.

To train the LMs for each dataset used in most experiments, we used the transcriptions of the training data. Though this corresponds to having an optimal LM for hybrid training, we also explore using LMs trained on unrelated data. For these LMs, we sampled 50000 sentences from the United Nations proceedings subset of the Europarl machine translation dataset [22] in Spanish, English, and French.

To obtain the character classes predicted by our models, we take the union of the character sets of each dataset.
Table I: CER of source models evaluated on each dataset. Bentham and IAM are English, Rimes is French, and Rodrigo, Spanish.

| Train Data | Test Data | Bentham | IAM | Rodrigo |
|------------|-----------|---------|-----|---------|
| Bentham    | 5.1       | 58.1    | 54.9| 45.8    |
| IAM        | 38.3      | 10.9    | 30.2| 42.7    |
| Rimes      | 47.0      | 28.7    | 5.1 | 51.9    |
| Rodrigo    | 64.8      | 74.0    | 69.6| 8.4     |

Because of this, during source model training, the classifiers output distributions over all characters, not just those characters contained in the source dataset. This way if the target dataset has additional characters, we do not need to modify the classifiers before hybrid training.

IV. EXPERIMENTAL RESULTS

In the following experiments, we use the following protocol. For source models, we trained 4 models for each dataset for 10 epochs using the ADAM optimizer to perform weight updates [23]. We then selected the best model using the Character Error Rate (CER) on the validation set after performing LM decoding using the dataset-specific LM. All reported numbers for source models are on the designated test splits for each dataset. These source models were used as the initial models in all hybrid training experiments, except where noted.

For hybrid training, we also trained 4 models where each hybrid model is initialized with weights learned on the source dataset. Hybrid models are trained for approximately 12000 weight updates using mini-batches of 8 images. Mini-batches contain both source and target images. To report metrics, we select the best model based on the validation set for the target data and then evaluate this model on the target data. While in practice this is not feasible because target data would not have ground truth transcriptions available, this allowed us to fairly compare different methods of hybrid training. A method for selecting the best model without using ground truth is left to future work.

A. Source Model Evaluation

Table I shows the CER of source models when evaluated on each dataset. As expected, source models obtain low CER when the test data matches the training data and high CER when there is a mismatch. Though this result may be obvious, it demonstrates the need for our hybrid training methodology in order to transfer models from one language to another. We also note that even though IAM and Bentham are both English datasets, models trained on one do not perform well on the other and have need of transfer learning.

The CERs obtained are competitive when compared with previous results reported in the literature. For example, [12] reports CER of 3.9 and 3.8 for IAM and Rimes respectively. In [24], a CER of 3.0 is reported on the Rodrigo dataset, though this number is not directly comparable to our reported results because they use a different data split and transcription preprocessing. Additionally, we binarized our data for transferability and generally CNN-BLSTM models perform better when using grayscale inputs. The best CER on Bentham reported in the 2014 ICFHR HTR competition is 5.0 for the restricted track [19]. Our reported numbers are also on source models that have not been trained to convergence on the source task as this improves hybrid training and further training of the source models results in even lower CERs.

B. Hybrid Training

In hybrid training, we varied 4 factors to gain a better understanding of the sensitivities of the method:

- Length of source model training
- Proportion of source and target data
- Data used to train the LM
- The $w$ and $\alpha$ LM parameters

Table II shows the CER after hybrid training for all language pairs for all experimental settings. Here we explain the column semantics of Table II. Source Epochs indicate how long source models were trained before hybrid training begins. We also varied the data used for LM training - either the ground truth training set transcriptions, Europarl corpus subset, or no LM was used. The next 2 columns respectively indicate the LM hyperparameters parameters and percentage of source data used in hybrid training. Remaining columns indicate Source-Target dataset pairs, where the first header row indicates the source language with target languages listed below. For example, the first data column is Bentham as the source with IAM as the target. The last column shows the average performance of the 6 language pairs involving Bentham, IAM, and Rimes. For this average, we excluded Rodrigo because of the extremely high CERs of unsuccessful transfers, which would dominate the average. For comparison, the last row shows performance of the source models before hybrid training.

If we consider the first row, pairwise transfers between Bentham, IAM, and Rimes are extremely successful, achieving CERs near to those obtained with full supervised training. It is interesting that while these three datasets can somewhat transfer to Rodrigo with CERs of about 18%, the reverse is not true. Only Rodrigo to Rimes hybrid training managed to significantly improve the CER over the source model, though the resulting 27% CER is still very high.

When the source model is trained to convergence, i.e. trained for 50 epochs instead of 10, CER on the source data does improve by about 1-2% (data not shown), but the transferability of the models decreases for all language pairs that successfully transfer. This is because after training for so long, the models can overfit the source data and may have difficulty unlearning some factors unique to the source dataset.
Table II: CER of hybrid trained models for all language pairs across a variety of experimental settings. See text for details on column semantics.

| Experimental Conditions | Bentham (English) | IAM (English) | Rimes (French) | Rodrigo (Spanish) |
|-------------------------|-------------------|---------------|----------------|-------------------|
|                         | Source LM LM Amount | Source | Ben. | Rim. | Rod. | Ben. | IAM | Rod. | Ben. | IAM | Rim. | Avg. |
| 10 Train 0.4 / 0.5 50%  | 7.8 | 5.8 | 18.0 | 5.6 | 3.2 | 18.2 | 6.8 | 6.8 | 17.0 | 61.3 | 77.6 | 27.2 |      |
| 50 Train 0.4 / 0.5 50%  | 8.5 | 4.0 | 18.0 | 5.9 | 3.6 | 18.9 | 7.0 | 6.7 | 17.7 | 61.6 | 74.5 | 68.5 |      |
| 10 Train 0.4 / 0.5 75%  | 8.0 | 4.1 | 18.2 | 5.5 | 3.1 | 17.1 | 7.4 | 6.3 | 17.5 | 55.2 | 61.9 | 25.4 |      |
| 10 Train 0.4 / 0.5 25%  | 8.9 | 4.1 | 18.4 | 6.6 | 3.4 | 17.0 | 6.5 | 6.9 | 17.0 | 66.4 | 72.1 | 30.4 |      |
| 10 Train 0.8 / 0.4 50%  | 8.0 | 4.1 | 18.2 | 5.5 | 3.1 | 17.1 | 7.4 | 6.3 | 17.5 | 55.2 | 61.9 | 25.4 |      |
| 10 Train 1.2 / 0.3 50%  | 8.9 | 4.1 | 18.4 | 6.6 | 3.4 | 17.0 | 6.5 | 6.9 | 17.0 | 66.4 | 72.1 | 30.4 |      |
| 10 None - 50%           | 58.1 | 54.9 | 45.8 | 38.3 | 30.2 | 42.7 | 47.0 | 28.7 | 51.9 | 54.8 | 74.0 | 69.6 |      |

Next we examined what proportion of source and target data is used during hybrid training. We found that using equal proportions works better overall than using 25% or 75% source data, though there are some language pairs that benefit from unequal proportions.

Next we examined the LM parameters $w$ and $\alpha$ used during LM decoding (Eqs. 2, 3). We determined our default setting of $w = 0.4$ and $\alpha = 0.5$ by cross validation to optimize the CER of source models evaluated on the datasets that they were not trained on (i.e., the off-diagonal entries of Table I). For example, Figure 1 shows heatmaps for the source IAM model evaluated on Bentham and Rimes. When evaluating the IAM model on Bentham or Rimes, we see better performance when $w \approx 0.4$ and $\alpha \approx 0.5$, but when we evaluate on IAM, $w = 1.2$ and $\alpha = 0.3$ perform best. We saw a similar trend when evaluating the Bentham source model on the other datasets.

A similar trend also holds for hybrid training. The default parameters of $w = 0.4$, $\alpha = 0.5$ work better than $w = 0.8$, $\alpha = 0.4$ and $w = 1.2$, $\alpha = 0.3$ for most language pairs. In particular, transferring to Rodrigo becomes unsuccessful with these alternate parameters. However, it is interesting that these parameters settings do greatly improve transfer from Rodrigo to Rimes (achieving 10.6 and 5.6 CER).

We conclude our experiments by varying the data used to train the LM. If we do not apply LM decoding during hybrid training (or equivalently use a LM where all sequences of characters are equally likely), we see that some language pairs will improve over the source model performance, though some do not improve or get worse. When we use the Europarl trained LMs, we see degraded performance with respect to the LMs trained on the dataset training sets, but this is expected to some degree. The Europarl corpus uses very formal language, and the modern Spanish is very different from the historical Spanish used in the 1545 Rodrigo manuscript. Transferring Bentham to IAM and vice versa, using the Europarl English LM greatly improves CER compared to using no LM at all. The same is also true for IAM and Rimes.

V. CONCLUSION

In this work we have proposed a methodology that trains HWR on a target language without using any labeled data in that language. It does so by leveraging labeled images in a closely related source language and a language model in the target language. After training a source model, we train on both the source and target data, imputing target labels using the current model predictions decoded by the LM. We demonstrate that our approach is successful on many pairs of languages using the IAM, Rimes, Bentham, and Rodrigo datasets. We explored the design choices of our hybrid training approach and make conclusions about the LM training data, LM hyperparameters, amount of source data in hybrid training, and length of source model training.
REFERENCES

[1] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855–868, 2009.

[2] P. Y. Simard, D. Steinkraus, J. C. Platt et al., “Best practices for convolutional neural networks applied to visual document analysis.” in International Conference on Document Analysis and Recognition, vol. 3, 2003, pp. 958–962.

[3] T. Varga and H. Bunke, “Perturbation models for generating synthetic handwritten data in handwriting recognition,” in Machine Learning in Document Analysis and Recognition. Springer, 2008, pp. 333–360.

[4] C. Wigington, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen, “Data augmentation for recognition of handwritten words and lines using a cnn-lstm network,” in International Conference on Document Analysis and Recognition (ICDAR), vol. 1. IEEE, 2017, pp. 639–645.

[5] C.-H. Tung, Y.-J. Chen, and H.-J. Lee, “Performance analysis of an ocr system via an artificial handwritten chinese character generator,” Pattern Recognition, vol. 27, no. 2, pp. 221–232, 1994.

[6] D.-H. Lee and H.-G. Cho, “A new synthesizing method for handwriting korean scripts,” International Journal of Pattern Recognition and Artificial Intelligence, vol. 12, no. 01, pp. 45–61, 1998.

[7] Y. Elarian, I. Ahmad, S. Awaida, W. G. Al-Khatib, and A. Zidouri, “An arabic handwriting synthesis system,” Pattern Recognition, vol. 48, no. 3, pp. 849–861, 2015.

[8] V. Frinken, A. Fischer, H. Bunke, and A. Foornes, “Co-training for handwritten word recognition,” in International Conference on Document Analysis and Recognition. IEEE, 2011, pp. 314–318.

[9] G. R. Ball and S. N. Srihari, “Semi-supervised learning for handwriting recognition,” in International Conference on Document Analysis and Recognition. IEEE, 2009, pp. 26–30.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in International Conference on Machine Learning. ACM, 2006, pp. 369–376.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011, iIEEE Catalog No.: CF11SRW-USB.

[12] T. Bluche, H. Ney, and C. Kermorvant, “A comparison of sequence-trained deep neural networks and recurrent neural networks optical modeling for handwriting recognition,” in International Conference on Statistical Language and Speech Processing. Springer, 2014, pp. 199–210.

[13] P. Voigtlaender, P. Doetsch, and H. Ney, “Handwriting recognition with large multidimensional long short-term memory recurrent neural networks,” in International Conference on Frontiers in Handwriting Recognition. IEEE, 2016, pp. 228–233.

[14] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Computer Speech & Language, vol. 13, no. 4, pp. 359–394, 1999.

[15] M. Mohri, F. Pereira, and M. Riley, “Speech recognition with weighted finite-state transducers,” in Springer Handbook of Speech Processing. Springer, 2008, pp. 559–584.

[16] 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.

[17] E. Augustin, J.-M. Brodin, M. Carr, E. Geoffrois, E. Grosicki, and F. Priteux, “RIMES evaluation campaign for handwritten mail processing,” in Workshop on Frontiers in Handwriting Recognition, no. 1, 2006.

[18] N. Serrano, F. Castro, and A. Juan, “The rodrigo database,” in International Conference on Language Resources and Evaluation, 2010, pp. 19–21.

[19] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal, “Ichr2014 competition on handwritten text recognition on transcriptionist datasets (htts),” in International Conference on Frontiers in Handwriting Recognition. IEEE, 2014, pp. 785–790.

[20] N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, 1979.

[21] C. Wolf, J.-M. Jolion, and F. Chassaing, “Text Localization, Enhancement and Binarization in Multimedia Documents,” in Proceedings of the International Conference on Pattern Recognition, vol. 2, 2002, pp. 1037–1040.

[22] P. Koehn, “Europarl: A parallel corpus for statistical machine translation,” in Machine Translation (MT) summit, vol. 5, 2005, pp. 79–86.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[24] E. Granell, E. Chammas, L. Likforman-Sulem, C.-D. Martinez-Hinarejos, C. Mokbel, and B.-I. Cistera, “Transcription of spanish historical handwritten documents with deep neural networks,” Journal of Imaging, vol. 4, no. 1, p. 15, 2018.