Transductive Learning for Reading Handwritten Tibetan Manuscripts

Sivan Keret
School of Computer Science
Tel Aviv University

Lior Wolf
Tel Aviv University and
Facebook AI Research

Nachum Dershowitz
School of Computer Science
Tel Aviv University

Eric Werner
Asia-Africa Institute
University of Hamburg

Orna Almogi
Asia-Africa Institute
University of Hamburg

Dorji Wangchuk
Asia-Africa Institute
University of Hamburg

***Abstract*—We examine the use case of performing handwritten character recognition (HCR) on a newly compiled collection of Tibetan historical documents, which presents multiple challenges, including inherent challenges such as image quality and the lack of word separation, and dataset challenges such as a lack of supervised training data.**

To tackle these challenges, we introduce an end-to-end unsupervised full-document HCR approach composed of unsupervised line segmentation and a convolutional recurrent neural network, trained using solely synthetic data. Various augmentations are applied to these synthesized images, and we compare the effect of each augmentation on the HCR results.

Since we work on a collection of historical manuscripts, we can fit the model to the available test data. During training, our network has access to both the labeled synthetic training data and the unlabeled images of the test set, and we adapt and evaluate four different semi-supervised learning and domain adaptation approaches for transductive learning in HCR.

We test our approach on a set of 167 images from the *Kadam* collection, containing 829 lines. We show that correct data augmentation is crucial for the success of HCR trained solely on synthetic data and that using an effective transductive learning approach drastically improves results.

***Index Terms*—Handwritten Recognition, Historical Document Analysis, CRNN, Transductive Learning, Domain Adaptation, Synthetic Data, Neural Network**

I. Introduction

Historical document analysis and specifically handwritten character recognition (HCR) have both been active fields of research for a long time. In recent years, with the advent of deep learning, the accuracy of HCR systems improved dramatically for whole word [1] and even whole paragraph [2] HCR. However, state-of-the-art algorithms require a vast amount of images with matching transcriptions for training.

While promising results in HCR were obtained, in certain cases they fall short due to the setting of the data. One example is the Tibetan text where syllable and word separation is done by the hard-to-segment *tsheg* ( ) sign. Thus, performing HCR on such data either requires performing word segmentation, a similarly hard task on Tibetan writing, or a vast amount of whole-line images of text, along with their transcriptions. Since transcribing Tibetan manuscripts is a time consuming task, collecting such data is challenging.

In the field of Tibetan character recognition, all research conducted so far involves either online HCR [3], [4], which does not deal with the challenges of text degradation, or offline HCR on upright *uchen* book-form characters [5]. A distinctive feature of the *umê* style we deal with, in comparison to *uchen*, is the absence of a horizontal guide line across the top of the characters, which makes for an additional challenge facing *umê* HCR.

We present an end-to-end framework for unsupervised HCR on *umê* Tibetan handwritten documents, using a projection-based method for line segmentation and a deep neural network for HCR, trained solely on computer-generated data. We follow the success of synthetic data generation for printed word recognition in natural images [6] by creating a mechanism for generating entire lines of handwritten data. We show that by using appropriate data rendering and augmentation, deep learning models trained solely on synthetic data can achieve good performance on original image text lines without the need for additional preprocessing other than line segmentation. In addition, as the training distribution is different from the test distribution, we examine two methods of regularization to allow the model to generalize better.

A key contribution of this paper is the study of transductive methods using images from the test data (without labels) to improve the prediction of the HCR model on the specific test distribution. In transductive learning, the

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC CoG 725974). Research supported in part by German-Israeli Foundation (GIF) grant #0603804893.

learner sees unlabeled samples from the test sets during training. While obtaining transcriptions for historical and handwritten documents is an expensive and complicated procedure, images of the documents themselves are available to researchers. A possible approach when obtaining a new dataset of documents can be fine-tuning existing HCR models for better performance on those documents. Hence, training a model on a synthetic dataset and using transductive methods to improve the results on a specific test set is a viable approach for historical HCR.

Transductive learning is closely related to unsupervised domain adaptation, where we consider the target domain to be represented by the test images. We therefore adapt two domain adaptation algorithms in the context of our HCR model. The first consists of using cycle-consistent adversarial networks (CycleGAN) [7] to either map the training data images to the testing data distribution or vice versa. CycleGANs were shown to work successfully on handwritten character generation [8] and for data augmentation [9]. The second is combining a domain-adversarial neural network approach (DANN) with the convolutional recurrent neural network (CRNN) architecture we use for HCR. DANN methods use adversarial training to find domain–invariant representations in neural networks.

Another approach we try is self-supervision, which was used in semi-supervised HCR in the past [10]. Self-supervision uses the output of the network on unlabeled data as labels for network training, choosing data according to the recognition confidence. We use self-supervision on test data as a method for transductive learning.

Furthermore, we introduce a novel approach for transductive learning. Adversarial training (AT) is a regularization method that uses adversarial examples (the examples that confuse the model the most) by adding a measurement of model accuracy on these examples as a loss during training. In [11], the authors show a method to compute an approximation of the adversarial examples and demonstrate their effectiveness for model regularization. In [12] they propose an extension to this method that enables the calculation of virtual adversarial examples from unlabeled data. We propose to compute the virtual adversarial examples from the test data and add a loss measuring the accuracy of the model on these samples during training. This allows the model to learn a classification that is locally isotropic around the distribution of the testing data. We show this approach to be superior in comparison with the three previous approaches.

Our main contributions are as follows: (1) An end-to-end system for historical handwritten paragraph recognition tested on a novel historical handwritten *umê* Tibetan document dataset. (2) A novel augmentation method for creating synthetic data for handwritten paragraph recognition on documents with overlapping lines. (3) A thorough examination of the effects of different augmentation methods on unsupervised handwritten paragraph recognition. (4) A thorough examination of the effects of different regularization methods on unsupervised handwritten paragraph recognition. (5) A comparison of different domain adaptation methods for transductive learning in the case of unsupervised handwritten paragraph recognition. (6) A novel approach to transductive learning for unsupervised handwritten text recognition by adding virtual adversarial learning loss from the test images (without labels). This is compared to three alternative transductive learning approaches and shown to be superior.

As far as we can ascertain, this is the first time OCR has been performed on *umê* Tibetan scripts and it is also the first attempt to perform line or paragraph-level handwritten OCR by training solely on synthetic data.

II. Previous Work

Tibetan character recognition: Previous work on HCR of Tibetan documents concentrates on character-level recognition of either modern handwritten cursive characters (*gyuk yig*) [3], [4] or scripts containing *uchen* upright book form characters [5]. The latter offers an offline character recognition dataset and gives HCR character-level results.

Handwritten character recognition: There have been some attempts at performing HCR on datasets with little to no transcription using transfer learning [13]. They show promising results, yet still require the use of a small transcribed dataset for training. Another approach for training with little transcribed data is using semi-supervised learning for HCR. In [10], self-training is used as a form of semi-supervised learning by training the network on samples which give confident output.

The literature on training HCR on purely synthetic data is scarce. In [14], a synthetic dataset of English words improves training for word-level HCR. However, this dataset was never tested for unsupervised HCR and no analysis was reported as to which data augmentation step is important for the success of HCR.

An HMM system for training on synthetic data was proposed by [15]. While they show good results, their method heavily relies on the use of a word-level language model and is less relevant to the Tibetan language, where separating words is as hard a task as HCR, since syllables and words are separated using the *tsheg* sign rather than a space. In another work, an HMM for printed Arabic text was learned on synthetic data without using a language model [16].

State-of-the-art methods in HCR include gated CRNNs [1]. A recent method [2] performs paragraph-level recognition for English handwritten documents.

Unsupervised line segmentation: There exist many methods for unsupervised handwritten line segmentation. A straightforward approaches is projection-based segmentation. Specifically, the Radon transform is sometimes used to approximate the locations of text lines [17], [18].

Regularization methods and HCR: In the context of word recognition, dropout has been successful with recur-

rent neural networks [19], and specifically for HCR [20]. We employ methods from the object recognition literature that are based on adversarial examples generated based on labeled [11] or unlabeled [12] data.

Domain adaptation: Transductive learning is far less researched in the context of deep learning than domain adaptation. Modern approaches employ adversarial training, following Ganin et al. [21]. VAT regularization [12] is a method for semi-supervised learning via virtual adversarial training, which was shown to also be relevant to domain adaptation [22].

One can perform domain adaptation by learning a mapping between two image distributions. This was shown to be possible in a completely unsupervised way in [7]. This mapping was shown to be effective for the task of data augmentation [9]. Specifically, Chang et al. [8] show a successful use of this method for Chinese handwritten character generation.

III. Methodology

We perform line segmentation by first using the Radon transform [17], [18] to find the center of each line and then clustering connected components to line centers to find exact line borders in case of curvilinear lines. We then use a convolutional recurrent neural network (CRNN), shown to work successfully for HCR [1]. Network training is done solely on synthetic text line images augmented to better resemble the distribution of historical documents. We note that the lines are rendered from Tibetan text unrelated to the test data. Since we train the network on one distribution and test it on another, we perform regularization to prevent the network from over-fitting to the training distribution. In addition to this completely unsupervised approach, we examine four domain adaptation approaches for using the unlabeled historical images in order to improve HCR training for these images.

A. Synthetic Training Data Augmentations

The ability of a network to generalize depends on the data distribution it trains on. A crucial part of the success of the OCR algorithm lies in the synthetic training data generator's ability to emulate the distribution of Tibetan document images. Computer-generated text does not contain noise or significant variations. Using data augmentations allows the network to be robust to noises and variations in the original data is a key aspect in our approach. An illustration of the rendering and augmentation steps can be seen in Fig. 1.

Multiline image rendering: As consecutive lines in the texts contain overlapping characters, we find that it is critical for the training data to contain such structure as well. Thus, when creating the synthetic data we do not render each line separately, but rather a number of text lines consecutively in each image. On the multiline images we run the same line segmentation algorithm that we run on the original data. As shown in Section V, this process

Fig. 1. Illustration of the text augmentation procedure. The first two images are renderings of text using different font sizes, stretch, weight and spacing; the 3rd is the text after rendering it together with adjacent lines and performing line segmentation; 4th is the result of performing elastic deformation augmentation on the image; 5th is after sinusoidal displacement; 6th is after rotation; 7th is after intensity gradient noise; 8th is after adding white Gaussian noise.

is crucial for the success of the OCR process on historical documents, and that without it the network is unable to generalize from the synthetic data to the original test data.

Geometric line augmentations: Due to the scribe's hand movements, cursive handwritten lines can be curved or rotated. To account for the variability in line structure, we apply two augmentations that change the geometry of the image at the line level: random image rotation and sinusoidal image displacement. The latter applies a vertical displacement to each image column according to a sinusoidal wave form and simulates the curvature of the images in handwritten documents.

Color modification: The digitization of the historical documents requires scanning text written on paper. During image acquisition, factors such as sensor quality and changes in illumination may result in changing pixel values. When creating the synthetic dataset we imitate such noise so as to train the HCR algorithm to become invariant to it. We perform two types of color modification augmentations. Additive white Gaussian noise is used to mimic the effect of the noise of the scanning procedure and the pattern of the paper. In addition, scanned documents often present a gradual intensity change throughout the image, which we simulate by adding a sinusoidal varying factor to the intensity.

Elastic distortion augmentation: In handwritten text there exists high variability in character shapes due to motion of the hand and the different handwriting styles of scribes. As the synthetic data is computer-generated, it has no such variations. To mimic this effect we apply elastic distortion, which has been used successfully for handwritten data augmentation in the past [23].

B. Convolutional Recurrent Neural Network

The base architecture we use for HCR is the convolutional recurrent neural network (CRNN) [24]. Using it, the work described in [1] achieved state-of-the-art results for supervised HCR. The CRNN architecture consists of a convolutional network followed by a recurrent network and a classification layer. We use connectionist temporal classification [25] loss, since it does not require prior alignment between the input and target sequences.

C. Regularization Methods

Since the distribution of the data we train on is different from the test data distribution, regularization is key to avoid overfitting to the training distribution. We examine two methods: conventional dropout [19], which is widely used in OCR [20] and HCR [26]), and virtual adversarial loss [11], which is new to this context. We compare the regularization methods separately and together, and show that indeed using regularization drastically improves the results. The regularization method of [11] considers the direction η in which the classifier h changes the most, and requires that the KL divergence D_{KL} between the network output on the permuted sample $x+\eta$ and the one-hot vector of the label y of the sample x is minimal. The loss term per sample is given by $L_{adv} = D_{KL}(h(x+\eta), y)$, where $h(x)$ is a vector of outcome probabilities, and $\eta = \varepsilon \cdot \text{sign}(\nabla_x D_{KL}(h(x), y))$.

As the output of our network is of varying length and is not aligned with the target sequence, we change the AT loss to be $L_{adv} = L_{CTC}(h(x+\eta), y)$, where $\eta = \varepsilon \cdot \text{sign}(\nabla_x L_{CTC}(h(x), y))$.

IV. Transductive Learning Approaches

When a given document collection is important enough, as in historical documents, spending the extra time to retrain the network for the specific collection is acceptable. The type of learning in which the model has access to unlabeled samples from the testing set is called transductive learning. Since we found no transductive learning method in the literature that is suitable for RNN architectures, we borrow four methods from the fields of domain adaptation and semi-supervised learning, adapt them to HCR and compare their performances regarding our problem.

A. VAT for Transductive Learning

The adversarial training loss of Section III-C was generalized in [12] to be used in an unsupervised manner in virtual adversarial training (VAT). This method introduces the local density smoothness (LDS) loss term, which measures the stability of the network:

$$L_{LDS}(x) = D_{KL}[h(x), h(x + r_{vadv})] \tag{1}$$

where r_{adv} is the adversarial sample with network output the furthest away from x that is at most a distance of ε from it: $r_{vadv} = \arg\max_{r;\|r\|_2 \leq \varepsilon} D_{KL}[h(x), h(x+r)]$.

In practice, D_{KL} is replaced by the L2 norm, and r_{vadv} is computed using the second-order Taylor approximation as $r_{vadv} \approx \varepsilon \frac{g}{\|g\|_2}$, where $g = \nabla_r D[h(x), h(x+r)]$.

In our work, we test the use of this loss for transductive learning by applying it on samples from the testing distribution rather than the training distribution. In addition, we examine an alternative approximation which is based on the L_{inf} norm, which leads to the slightly modified formulation of

$$r_{vadv} \approx \varepsilon \cdot \text{sign}(g) \tag{2}$$

To apply VAT in HCR, we add noise to the image in the same way as in regular VAT, and calculate the distance loss to be the average of D on each output time step:

$$L_{LDS}(x) = \sum_{t=0}^{T} D[h^t(x), h^t(x + r_{vadv})] \tag{3}$$

B. Domain Adaptation with CRNN

The DANN method [21] performs unsupervised domain adaptation between source domain S and target domain T by training three networks. The first network is responsible for feature extraction of the image representation. The two following networks receive the extracted features as input. The domain classification network attempts to correctly classify between the features obtained for samples from the two domains S and T. The label classification network is trained to correctly predict task-specific class labels.

Denoting the feature extraction, label classification and domain classification networks as F, C_{cls} and C_d, respectively, random source samples and matching labels as (X_s, Y_s), and random target domain samples as x_t, DANN solves the following optimization problem:

$$\min_{F, C_{cls}} \mathop{\mathbb{E}}_{(x_x, y_s)} L_{cls}(x_s, y_s) \tag{4}$$

$$\min_{C_d} \max_{F} \mathop{\mathbb{E}}_{x_t}[log(C_d(x_t))] + \mathop{\mathbb{E}}_{x_s}[log(1 - C_d(x_s))] \tag{5}$$

where $L_{cls}(x, y)$ is defined as the task-specific classification loss of the network. The first equation causes the feature extraction network and the classification network to collaborate on minimizing the classification loss on the supervised samples in the source domain. The second loss causes the domain classification network and the feature extraction network to behave adversarially: while the domain classifier tries to distinguish between the two domains, the feature extractor wishes to have the features arising from samples in the two domains as indistinguishable as possible, the logic being that if similar

features are extracted, then the classifier, which is trained only in S, would also fit the samples in T.

We integrate domain adversarial loss in our framework in two locations. One is after the CNN feature extraction and before the RNN, and the other is after the RNN and before the classification layer. We do this by adding additional domain classification networks to the model training. One receives the feature layer output of the CNN network as input, and the other the RNN network output prior to the classification layer. Similarly, two adversarial losses are added to the training of the model: minimax games between the CNN and CNN+RNN parts of the network and the first and second domain classification networks, respectively. We compare the results of training the network in these three scenarios:

- Adding only one domain adversarial network receiving the CNN output as input.
- Adding only one domain adversarial network receiving the RNN output as input.
- Adding both domain adversarial networks, receiving the RNN and CNN outputs as inputs, respectively.

An illustration of the training with both CNN and RNN domain adversarial networks is in Fig. 2.

C. CycleGan for Visual Domain Adaptation

We observe that augmentation that brings the training distribution as close as possible to the testing distribution is crucial. Mapping images between two image distributions, without relying on matching samples in the two domains, is the goal of unsupervised domain mapping methods, including CycleGAN [7]. This method was used in the past to generate handwritten Chinese characters [8].

We tried to use CycleGAN as an augmentation method and as a transductive learning method. As augmentation, it did not perform as well as the engineered augmentation methods, despite producing attractive looking images. As a transductive learning method, we employ it in the other direction, that is, to transform the test images to become visually similar to the training images. The method produces clean binary images, as shown in Fig. 4. These images are better at test time than those obtained by a naive binarization process. However, the results it yields are worse than those obtained using the original images.

D. Self-Supervision for Transductive learning

Previous work [10] has shown impressive results when using self-supervision in the field of semi-supervised handwriting recognition. In this method, the network is first trained on labeled data, followed by several iterations where it is retrained on a subset of unsupervised data, for which its own output was recognized with high confidence, using that output as labels.

We apply self-supervision by retraining the network on confident output on the historical test data. For this, a confidence measure is needed. We experiment with two measures of output confidence:

1) The intensity of the normalized output activation.
2) The consensus among four trained networks, each initialized randomly.

In both experiments we train the networks in iterations. In each iteration we retrain on all test samples with confidence measure above a threshold in addition to labeled train data. In (1) we experiment with threshold values between 0.1 and 0.9. In (2) we train four networks and experiment with thresholds of two to four networks agreeing.

We note that [10] use a validation set to estimate a function of the network accuracy based on the number of networks agreeing and the output of the network. They then use the estimated accuracy as the confidence measure. Since we observed that the network accuracy on synthetic data does not predict accuracy on historical data, we do not employ such a validation set and use the number of networks agreeing directly.

V. Experiments

To test our framework we use a set of 167 images of historical handwritten manuscripts from the *bKa'-gdams* (*Kadam*) collection, containing manuscripts mostly dating from around the 11th to 14th centuries. Each image contains four to five lines, totaling 829 lines. The images are in relatively good condition, yet contain problems seen in historical data such as ink spreading, faded characters and scanning artifacts.

The transcription of the images in the test dataset was created for Tibetan research purposes and not specifically for this research. An example of the transcription in the test dataset compared to the exact letter-by-letter transcription of an example manuscript can be seen in Fig. 3. There are some inconsistencies between the text in the images and the transcription text, caused by two main reasons: (i) The scribes use abbreviations frequently, and in many cases the transcription includes the full form of the intended text and not the abbreviation; and (ii) differences in writing, such as writing a number and spelling a number. As a result, the performance we obtain is a lower bound of the actual results of the models. However, as the errors in the dataset affect all models similarly, it is safe to say that these errors do not affect the comparison between the different methods.

To measure the performance of different methods we use the standard character error rate (CER) measure defined as

$$CER(prediction, label) = \frac{D(prediction, label)}{length(label)}, \qquad (6)$$

where D is Levenshtein distance (the minimum number of single-letter edits required to change one word into the other). Word error rate is less relevant for Tibetan, where most words are disyllabic and the separation between syllables is signified using a *tsheg*.

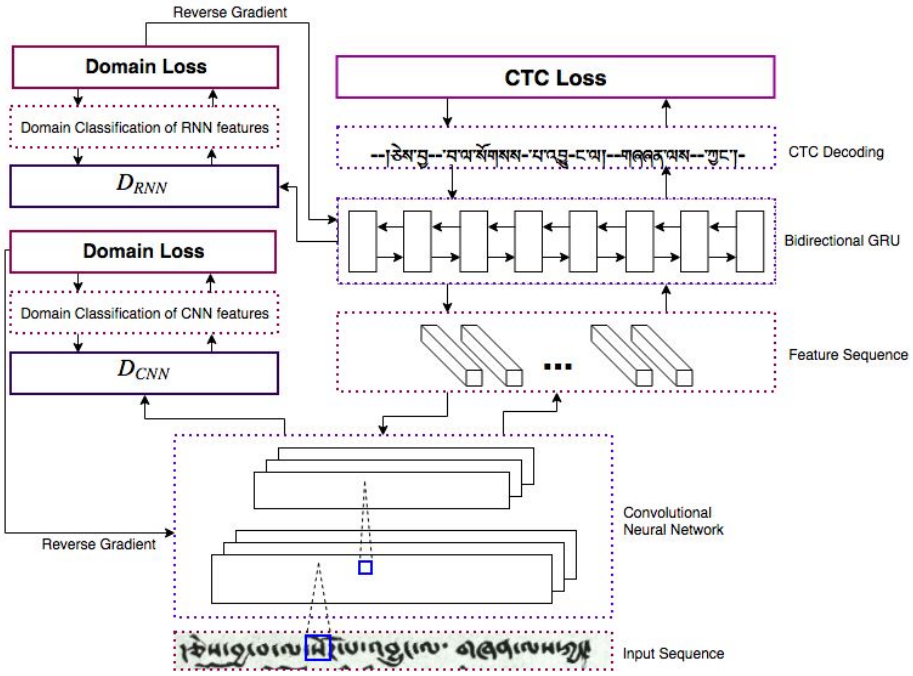

Fig. 2. Illustration of the training procedure with DANN added to the CNN output and RNN output.

Fig. 3. Example of transcription errors. Above is the test image and below is the ground truth transcription. Letters omitted from the original manuscript are marked in pink. Letters that were changed are marked in green.

A. Implementation Details

To support full reproducibility we publish our code[1]. For the convolutional part of the model, we use the first convolutional layer and the first three residual blocks of the ResNet18 architecture. To flatten the height of the images we add max-pooling layers between the first convolution layer and the first residual block, between the second and third residual blocks and after the third residual block. We examined the results of a model pre-trained on ImageNet and a model trained solely on our data and saw that using a pre-trained network does not affect the final results, but does decrease the training time significantly. Thus, to accelerate training, we use a convolutional network pre-trained on ImageNet.

For the RNN part, we use two layers of bidirectional RNN with GRU cells. For all results, except the one showing the performance without dropout, we add dropout layers before and after the RNN. We also tried adding dropout in between the RNN layers, but it did not improve the results. Training is performed with an Adam optimizer with a weight decay of 0.0001.

[1]https://github.com/SivanKe/TransductiveHCR.git

To avoid reaching the point of overfitting while not using the original data for early stopping, we choose to stop training when the error rate on the synthetic data reaches a plateau. To do this we manually look at the error rate of the model on the synthetic validation data and visually choose the plateau point. To give fair and accurate results we average the results of each model on 50,000 iterations after the plateau point and report both the average and the standard deviation of the model.

B. Augmentation Ablation Study

In Table I we compare the results of models trained on data created with one or more augmentations omitted. The most dramatic effect is achieved by the augmentation of rendering multiple lines and using line segmentation to crop each line. This result is probably significant for many other full document recognition systems. Another drastic effect is that of omitting both color augmentations. While they seem to be somewhat interchangeable, as each one by itself causes a much less significant change in results, removing both has drastic effects. This echoes the importance of binarization in classical OCR systems.

TABLE I
TEST CER WITHOUT SPECIFIC TYPES OF AUGMENTATION

Augmentation	Test CER
Multiline omitted	92.58±0.96%
Color tran. omitted	59.38±4.03%
Multi-font properties omitted	32.74±0.98%
Elastic omitted	27.95±0.24%
Geometric tran. omitted	27.49±0.96%
Rotation omitted	26.97±1.48%
Color Gaussian omitted	26.29±1.06%
Geometric sine omitted	25.77±1.00%
Color sine omitted	25.09±0.45%
All augmentations	**24.23±0.93%**

TABLE II
CER FOR VARIOUS REGULARIZATION METHODS

Regularization	Test CER
Only L2 and BN (RNN Hidden 128)	25.80±0.69%
Dropout	24.23±0.93%
Dropout (RNN hidden double)	25.65±0.56%
Dropout + AT	**22.23±0.70%**

C. Regularization Importance

To show the importance of regularization in training HCR on synthetic data, we compare the results without dropout, with dropout, and with dropout and adversarial training loss. In all cases, batch normalization and $L2$ regularization were applied.

After examining the influence of adding dropout at three different stages of the network (between the RNN and the CNN, between RNN layers and after RNN output), we chose to add dropout before and after the RNN, as it gives the best results. As the change in results was not significant, we leave this comparison out. We follow the path of [20], and examine the addition of dropout both with the original RNN hidden size, and with doubling the size of the hidden layer, as dropout effectively causes the number of parameters trained in each step to drop by a factor of two. We find that while adding dropout does improve the results, doubling the size of the hidden layer does not. See Table II for the results.

In addition, we see that adding AT regularization to dropout improves the results. This is of particular interest, as we apply AT using CTC loss (since we have no alignment between the CRNN output and the transcription), which is very different from the loss that is applied in the original article [11].

D. Transductive Methods Comparison

Test side data augmentation: To show the importance of using transductive methods for HCR with sparse data, we compare these methods to the baseline method of test side data augmentation used by [27]. Given a test image, we generate ten variant images using the augmentations described in Section III-A. In [27], the most frequently occurring prediction is chosen as output. Since we perform handwritten recognition on full text lines, it is rare for two augmentations to give the same result. We, therefore, adapted the method of [27] by computing the distances between all pairs of output predictions, choosing the prediction with the lowest median distance as output.

TABLE III
CER FOR VARIOUS TRANSDUCTIVE LEARNING METHODS

Transductive	Test CER
Best without transductive	22.23±0.70%
Naive binarization on test	40.40±2.13%
Test side augmentation	22.68±0.52%
Cycle on test (best obtained)	29.45±0.60%
Cycle on train (best obtained)	22.92±0.30%
Self-supervision score conf. (best obtained)	20.64±0.83%
Self-supervision ensemble conf. (best obtained)	16.53±0.21%
DANN RNN	19.04±0.51%
DANN CNN	18.75±0.67%
DANN CNN+RNN	16.80±0.60%
Sign VAT	**15.44±0.24%**

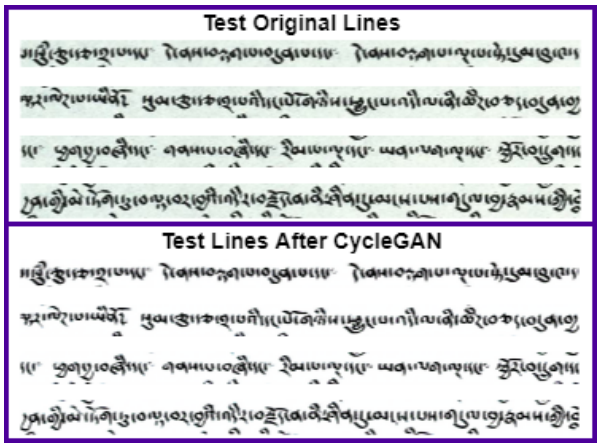

Fig. 4. The effect of the CycleGAN transformation on test images.

Result analysis: In Table III, we compare the four transductive learning methods proposed in Section IV. "Test side augmentation" is the result of the baseline described above. "Cycle on test" is the result of training a method without transductive learning and testing it on the testing dataset transformed by a CycleGAN network trained for 100,000 iterations (the best result we got) on the training and testing datasets. "Cycle on train" is the result of training the model on training data transformed by a CycleGAN network trained on the synthetic and original data for 10^6 iterations. All experiments on CycleGan were done using dropout as the regularization method of the HCR network. "Naive binarization on test" means performing inference on test images after adaptive binarization, and is shown for comparison with using CycleGAN as a form of test data binarization. "Self-supervision score conf." and "Self-supervision ensemble conf." mean self-supervision with confidence measure of output score and ensemble agreement, respectively. "DANN RNN", "DANN CNN", and "DANN CNN+RNN" are done by training

the network adding DANN loss on the CNN, RNN and both CNN and RNN output features, respectively. "Sign VAT" means adding VAT loss to the CTC loss. We call this experiment "Sign VAT" since VAT is calculated using the sign of the direction vector and not the normalized vector, as shown in Eq. (2), which we found to work better than using the original VAT loss. In the three methods of adding DANN, self-supervision and VAT loss, we use a ratio of ten to one between the transductive loss and the CTC loss, as it gives the best results for all three algorithms.

We see that VAT, DANN and ensemble self-supervision methods significantly outperform self-supervision, test-side augmentation and CycleGan. We believe that test-side augmentation gives inferior results as it does not incorporate knowledge on test distribution during training. Output based self-supervision giving inferior results is consistent with the report of [10]. As for the results of CycleGan, this might be due to the fact that the other methods perform domain adaptation directly for the classification task, while CycleGan transfers the style of the images without regard for the labels of the characters. Another important takeaway from these experiments is the significance of transductive learning for unsupervised HCR. We show that by using transductive learning methods we are able to improve results by more than 6%.

In addition, we see that adding DANN both after the CNN feature extraction and after the RNN feature extraction is superior to using only one DANN. Inspired by this, we tried adding VAT loss to different parts of the network, but it did not improve the results.

Using VAT on test samples appears to be superior to using DANN or self-supervision, but not by a large margin, compared to the improvement achieved by using either.

VI. Conclusions

We have presented a novel approach to end-to-end HCR of historical handwritten documents, suited especially to the challenges of the Tibetan language and script. We evtaluated the approach on a novel test dataset consisting of images taken from the historical *Kadam* manuscript collection and showed promising results.

We have shown that using proper augmentation and correctly rendering printed text, a model trained solely on synthetic data can have acceptable performance for HCR on ancient manuscripts. We presented a novel approach to augmentation of entire text lines in the case of HCR on documents with touching lines and showed its importance to the HCR process.

We have seen that strong regularization techniques are important for unsupervised HCR. In addition, we presented four approaches to transductive learning for HCR, and examined their impact. We showed that using transductive learning can significantly improve the results for HCR trained on synthetic data.

References

[1] T. Bluche and R. Messina, "Gated convolutional RNN for multilingual handwriting recognition," in *ICDAR*, 2017.

[2] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention," in *ICDAR*, 2017.

[3] L. Ma and J. Wu, "A recognition system for online handwritten Tibetan characters," in *GREC*, 2013.

[4] L. Ma, H. Liu, and J. Wu, "MRG-OHTC database for online handwritten Tibetan character recognition," in *ICDAR*, 2011.

[5] H. Haung and F. Da, "A database for off-line handwritten Tibetan character recognition," *Journal of Information and Computational Science*, vol. 9, pp. 5987–5993, 2012.

[6] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *CVPR*, 2018.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV*, 2017.

[8] B. Chang, Q. Zhang, S. Pan, and L. Meng, "Generating handwritten Chinese characters using CycleGAN," in *WACV*, 2018.

[9] X. Zhu, Y. Liu, Z. Qin, and J. Li, "Data augmentation in emotion classification using generative adversarial networks," *International Journal of Computer Applications*, p. 51, 2012.

[10] V. Frinken and H. Bunke, "Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition," in *ICDAR*, 2009.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[12] T. Miyato, S.-I. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning." in *PAMI*, 2018.

[13] A. Granet, E. Morin, H. Mouchère, S. Quiniou, and C. Viard-Gaudin, "Transfer learning for handwriting recognition on historical documents," in *Int. Conf. on Pattern Recognition Applications and Methods*, Madeira, Portugal, 2018.

[14] P. Krishnan and C. Jawahar, "Generating synthetic data for text recognition," *arXiv preprint arXiv:1608.04224*, 2016.

[15] M. Kozielski, M. Nuhn, P. Doetsch, and H. Ney, "Towards unsupervised learning for handwriting recognition," in *ICFHR*, 2014.

[16] I. Ahmad and G. Fink, "Training an Arabic handwriting recognizer w/o a handwritten training data set," in *ICDAR*, 2015.

[17] B. M. K. and M. J. Novel, "Approach for baseline detection and text line segmentation." *Int. J. Computer Applications*, 2012.

[18] R. Ptak, B. Żygadło, and O. Unold, "Projection–based text line segmentation with a variable threshold," *Int. J. Applied Mathematics and Computer Science*, 2017.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, 2014.

[20] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *ICFHR*, 2014.

[21] Y. G. et al., "Domain-adversarial training of neural networks," *JMLR*, vol. 17, no. 1, pp. 2096–2030, 2016.

[22] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A DIRT-T approach to unsupervise domain adaptation," in *ICLR*, 2018.

[23] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, 2003.

[24] 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," *T. Pattern Anal. Mach. Intell.*, 2017.

[25] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Int. Conf. Machine Learning*, 2006.

[26] T. Bluche, C. Kermorvant, and J. Louradour, "Where to apply dropout in recurrent neural networks for handwriting recognition?" in *ICDAR*, 2015.

[27] 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 *ICDAR*, 2017.