Start, Follow, Read: End-to-End Full-Page Handwriting Recognition

Curtis Wigington[1,2], Chris Tensmeyer[1,2], Brian Davis[1], William Barrett[1], Brian Price[2], and Scott Cohen[2]

[1] Brigham Young University
[2] Adobe Research
wigingto@adobe.com code at: https://github.com/cwig/start_follow_read

Abstract. Despite decades of research, offline handwriting recognition (HWR) of degraded historical documents remains a challenging problem, which if solved could greatly improve the searchability of online cultural heritage archives. HWR models are often limited by the accuracy of the preceding steps of text detection and segmentation. Motivated by this, we present a deep learning model that jointly learns text detection, segmentation, and recognition using mostly images without detection or segmentation annotations. Our Start, Follow, Read (SFR) model is composed of a Region Proposal Network to find the start position of text lines, a novel line follower network that incrementally follows and preprocesses lines of (perhaps curved) text into dewarped images suitable for recognition by a CNN-LSTM network. SFR exceeds the performance of the winner of the ICDAR2017 handwriting recognition competition, even when not using the provided competition region annotations.

Keywords: Handwriting Recognition, Document Analysis, Historical Document Processing, Text Detection, Text Line Segmentation.

1 Introduction

In offline handwriting recognition (HWR), images of handwritten documents are converted into digital text. Though recognition accuracy on modern printed documents has reached acceptable performance for some languages [28], HWR for degraded historical documents remains a challenging problem due to large variations in handwriting appearance and various noise factors. Achieving accurate HWR in this domain would help promote and preserve cultural heritage by improving efforts to create publicly available transcriptions of historical documents. Such efforts are being performed by many national archives and other organizations around the world, but typically use manual transcriptions, which are costly and time-consuming to produce. While this work focuses discussion on one of the most difficult HWR domains, i.e. historical documents [9], our proposed methods are equally applicable to other HWR domains.

For most HWR models, text lines must be detected and segmented from the image before recognition can occur. This is challenging for historical documents

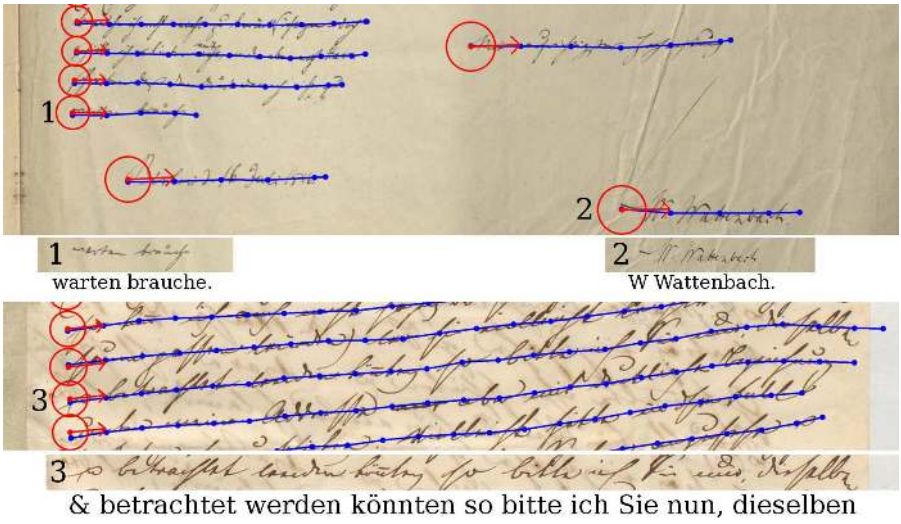


Fig. 1: Start, Follow, Read on two document snippets. Red circles and arrows show the Start-of-Line finder network's detected position, scale, and direction. Blue lines show the path taken by the Line Follower network to produce normalized text lines; three lines are shown with the HWR network's transcription.

because they may contain significant amounts of noise, such as stains, tears, uneven illumination, and ink fade, seepage, and bleed-through. Errors in the detection or segmentation of text propagate to the recognition stage, and as noted in [25], the majority of errors in complete HWR systems are due to incorrect line segmentation rather than incorrect character or word recognition. Despite this, line detection and segmentation are commonly performed by separate algorithms in an independent fashion and many HWR models are designed, trained, and evaluated only in the context of ground truth line segmentations [18,29].

A few works have attempted to combine detection, segmentation, and recognition. Bluche et al. proposed a recurrent model that detects and recognizes text lines using a soft-attention mechanism [3]. However, this method is slow because the model processes the whole image twice to transcribe each text line. Furthermore, the method does not allow for preprocessing detected lines of text (e.g. normalize text height), which is shown to improve HWR performance [11]. In contrast, our proposed model efficiently detects all text lines in a single pass and uses learned preprocessing before applying the HWR model on each line independently, allowing each line to be recognized in parallel.

In this work, we present Start, Follow, Read (SFR), a novel end-to-end full-page handwriting recognition model comprised of 3 sub-models: a Start-of-Line (SOL) finder, a Line Follower (LF), a line-level HWR model. The SOL finder is a Region Proposal Network (RPN) where the regions proposed are the start positions and orientations of the text lines in a given document image. The LF model starts at each predicted SOL position, incrementally steps along the text

line, following curvature, and produces a normalized text image. Finally, a state-of-the-art HWR model predicts a transcription from the normalized line image. Fig. 1 shows how the SOL, LF, and HWR networks process document images.

One main contribution is our novel LF network, which can segment and normalize curved text (e.g. Fig. 1 bottom) that cannot be segmented with a bounding box. Though [19] previously used a SOL network, we propose a new architecture and a new training scheme that optimizes recognition performance. Another contribution is the joint training of the three components on a large collection of images that have transcriptions only, which allows the SOL finder, LF, and HWR to mutually adapt to, and supervise, each other. In particular, we demonstrate that the LF and HWR networks can be used to derive and refine latent targets for the SOL network; this method only requires pre-training on a small number of images (e.g. 50) with additional segmentation labels.

We demonstrate state-of-the-art performance on the ICDAR2017 HWR competition dataset [25]. This competition represents a common scenario where a collection is manually transcribed, but segmentations are not annotated. While the best previous result is 71.5 BLEU score using the provided region annotations (57.3 BLEU without), SFR achieves 73.0 BLEU with region annotations, and performs only slightly worse with a 72.3 BLEU score without regions.

2 Related Work

Though segmentation and recognition are critical components of HWR, most prior works solve these problems independently: text lines are detected, segmented, and preprocessed into rectangular image snippets before being transcribed by a recognition model. Errors in the detection, segmentation, or preprocessing steps often lead to poor recognition. In contrast, SFR jointly performs detection, segmentation, preprocessing, and recognition in an end-to-end model.

Text Line Detection/Segmentation. Often, peaks in vertical projection profiles (summing pixels along rows) are used to detect transitions from dark text to lighter inter-line space [13,1,26]. However, these methods are sensitive to images with noise and curved handwriting (e.g. the image in Fig 1). Additionally, such methods assume that distinct text lines cannot be horizontally adjacent, an assumption that is violated in practice. The recursive XY cut algorithm also considers the horizontal projection profile to make vertical image cuts along detected white space, but requires manually tuning of threshold values [14].

Seam carving [2] based methods improve on projection profile methods because seams can follow the curves of text lines. Boiangiu et al. use a pixel information measure for computing an energy map for seam carving [5], while Saabni and El-Sana use a signed distance transform to compute the energy [24]. The winner of the ICDAR2017 handwriting recognition competition [25] corrected the output of a seam carving method by using a Convolutional Neural Network (CNN) to predict if lines were over-segmented or under-segmented.

Tian et al. [31] use a Region Proposal Network (RPN), similar to Faster-RCNN [23], to predict bounding boxes for text in the wild detection. However,

unlike Faster-RCNN, their RPN predicts many small boxes along the text line in order to follow skewed or curved lines. These boxes must be clustered in a separate step, which may result in over- or under-segmentation.

Handwriting Recognition. Some early handwriting recognition models used machine learning models such as neural networks and Support Vector Machines (SVM) to learn whole word, character and stroke classifiers using handcrafted features [32,17]. However, such methods required further segmentation of text line images into primitives such as characters or strokes, which itself was error prone. Hidden Markov Model (HMM) approaches similar to those used in speech recognition then became popular because they were able to perform alignment to refine segmentation hypotheses [20]. These approaches are often combined with a Language Model (LM) or lexicon to refine predictions to more closely resemble valid natural language [6].

The introduction of the Connectionist Temporal Classification (CTC) loss [10] allowed recurrent neural network (RNN) character classifiers to perform alignment similar to HMMs, which led to the current dominance of RNN approaches for HWR. Long-Short Term Memory (LSTM) networks combined with convolutional networks, CTC, and LM decoding represent the current state-of-the-art in HWR [11]. Additional improvements, such as Multi-Dimensional LSTMs [12], neural network LMs [34], and warp based data augmentation [33] have also been proposed. Preprocessing text lines to deslant, increase contrast, normalize text height, and remove noise is also a critical component of many HWR systems [11].

Combined Segmentation and Recognition. Moysset et al. proposed predicting SOL positions with a RPN and then applying a HWR network to axis-aligned bounding boxes beginning at the SOL [19]. However, the two models are trained independently and bounding box segmentations cannot handle curved text. Recurrently computing an attention mask for recognizion has been applied at the line-level [3] and the character level [4] and though these methods are computationally expenseive, they have been shown to successfully follow slanted lines on clean datasets of modern handwriting with well-separated text lines. In contrast, we demonstrate our work on a more challenging dataset of noisy historical handwritten documents.

3 Proposed Model: Start, Follow, Read

In order to jointly learn text detection, segmentation, and recognition, we propose the SFR model with three components: the Start of Line (SOL) network, the Line Follower (LF) network, and the Handwriting Recognition (HWR) network. After pre-training each network (Sec. 3.3) individually, we jointly train the models using only ground truth (GT) transcriptions (with line breaks) (Sec. 3.3).

3.1 Network Description

Start-of-Line Network Our Start-of-Line (SOL) network is a RPN that detects the starting points of text lines. We formulate the SOL task similar to [19],

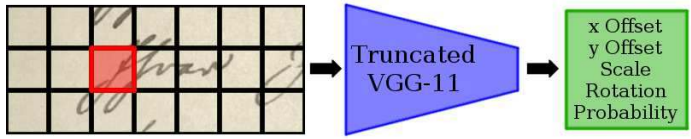
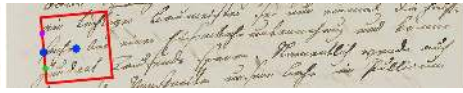


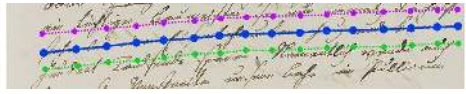
Fig. 2: The SOL network densely predicts x and y offsets, scale, rotation angle, and probability of occurrence for every 16x16 input patch. Contrary to left-right segmentation methods, this allows detection of horizontally adjacent text lines.

(a) SOL position and first viewing window



(b) Predicted next position

(c) Next viewing window



(d) Resulting LF center line path

(e) Normalized handwriting line

Fig. 3: The LF begins at a SOL (a) and regresses a new position indicated by the second blue dot in (b). The next input is a new viewing window (c). This process repeats until it reaches the image edge. The purple and green lines in (d) show the segmentation that produces the normalized handwriting line (e).

but we use a truncated VGG-11 architecture [27] instead of an MDLSTM architecture to densely predict SOL positions (Fig. 2). For an image patch, we regress (x_0, y_0) coordinates, scale s_0, rotation θ_0, and probability of occurrence p_0. For image patches with a SOL (e.g. red box in Fig. 2), the network should predict $p_0 = 1$, otherwise 0. We remove the fully connected and final pooling layers of VGG-11 for a prediction stride of 16x16 and, similar to Faster R-CNN [23], predicted (x, y) coordinates are offsets relative to the patch center. The scale and rotation correspond to the size of handwriting and slant of the text line.

Line Follower After identifying the SOL position, our novel LF network follows the handwriting line in incremental steps and outputs a dewarped text line image suitable for HWR (see Fig. 3). Instead of segmenting text lines with a bounding box (e.g. [19]), the LF network segments polygonal regions and is capable of following and straightening arbitrarily curved text.

The LF is a recurrent network that given a current position and angle of rotation (x_i, y_i, θ_i), resamples a small viewing window (red box in Fig. 3a) that is fed to a CNN to regress $(x_{i+1}, y_{i+1}, \theta_{i+1})$ (Fig. 3b). This process is repeated until the image edge (Figs. 3c and 3d), and during training we use the HWR

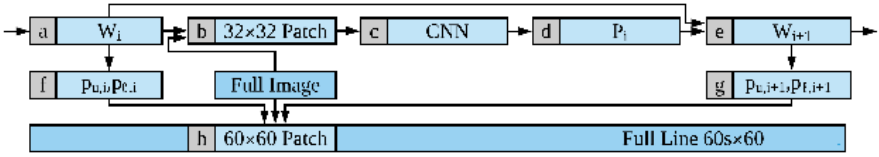


Fig. 4: Using the current transformation W_i (a), we resample a 32×32 patch (b) from the input image. A CNN regresses a transform change (d) used to compute the next transformation (e). Using the upper and lower points (f,g) of the LF path, we resample a 60×60 patch to be part of the normalized, segmented line.

network to decide where the text line ends. The initial position and rotation is determined by a predicted SOL. The size of the viewing window is determined by the predicted SOL scale and remains fixed.

Resampling the input image to obtain the viewing window is done similarly to the Spatial Transform Network [15] using an affine transformation matrix that maps input image coordinates to viewing image coordinates (see Fig. 4). This allows LF errors to be backpropagated through viewing windows. The first viewing window matrix, $W_0 = AW_{SOL}$, is the composition of the mapping defined by a transformation SOL matrix W_{SOL} (defined by values of the SOL network prediction) and a look-ahead matrix A:

$$W_{SOL} = \begin{bmatrix} \frac{1}{s_0} & 0 & 0 \\ 0 & \frac{1}{s_0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_0) & -\sin(\theta_0) & 0 \\ \sin(\theta_0) & \cos(\theta_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0.5 & 0 & -1 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The look-ahead matrix gives the LF network enough context to correctly follow lines. For each step i, we extract a 32×32 viewing window patch by resampling according to W_i. When resampling, the (x, y) coordinates in the patch are normalized to the range $(-1, 1)$. Given the $(i-1)^{\text{th}}$ viewing window patch, the LF network regresses x_i, y_i and θ_i, which are used to form the prediction matrix P_i. We then compute $W_i = P_i W_{i-1}$ with

$$P_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

To obtain the output image for HWR, we first represent the normalized handwriting line path as a sequence of upper and lower coordinate pairs, $p_{u,i}$ and $p_{\ell,i}$ (green and purple lines in Fig 3d), which are computed by multiplying the upper and lower midpoints of predicted windows by their inverse transformations:

$$p_{u,i}, p_{\ell,i} = \begin{bmatrix} x_{u,i} & x_{\ell,i} \\ y_{u,i} & y_{\ell,i} \\ 1 & 1 \end{bmatrix} = W_i^{-1} A \begin{bmatrix} 0 & 0 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \tag{3}$$

We extract the handwriting line by mapping each $p_{u,i}$, $p_{\ell,i}$, $p_{u,i+1}$, and $p_{\ell,i+1}$ to the corners of a 60×60 patch. We concatenate all such patches to form a full handwriting line of size $60s \times 60$ where s is the number of LF steps.

The architecture of the LF is a 7-layer CNN with 3x3 kernels and 64, 128, 256, 256, 512, and 512 feature maps on the 6 convolution layers. We apply Batch Normalization (BN) after layers 4 and 5 and 2x2 Max Pooling (MP) after layers 1, 2, 4, and 6. A fully connected layer is used to regress the X, Y, θ outputs with initial bias parameters for X initialized to 1 and biases for Y and θ initialized to 0. This initialization is a prior that lines are straight and read left-to-right.

Handwriting Recognition After the LF network produces a normalized line image, it is fed to a CNN-LSTM network to produce a transcription. The CNN part of the HWR network learns high level features that are vertically collapsed to create a horizontal 1D sequence that is fed to a Bidirection LSTM model. In the BLSTM, learned context features propagate forward and backwards along the sequence before a character classifier is applied to each output time step.

The output sequence of character predictions is much longer than the GT transcriptions, but includes a blank character for use in the CTC decoding step [10]. Decoding is performed by first collapsing non-blank repeating characters and then removing the blanks, e.g. the output `--hh--e-lll-l----oo--` is decoded as `hello`. While the CTC loss does not explicitly enforce alignment between predicted characters and the input image, in practice, we are able to exploit this alignment to refine SOL predictions (see Sec. 3.3).

The architecture of our HWR network is on a CNN-LSTM HWR network [33] and is similar to our LF network. The input size is $W \times 60$, where W, can dynamically vary. There are 6 convolutional layers with 3x3 filters with 64, 128, 256, 256, 512, and 512 feature maps respectively. BN is applied after layers 4 and 5, and 2x2 MP (stride 2) is applied after layers 1, 2. To collapse features vertically we use 2x2 MP with a vertical stride of 2 and a horizontal stride of 1 after layers 4 and 6. Features are concatenated vertically to form a sequence of 1024-dimensional feature vectors that are fed to a 2-layer BLSTM with 512 hidden nodes and 0.5 probability of node dropout. A fully connected layer is applied at each time step to produce character classifications.

The HWR also serves an additional function. LF always runs to the edge of the page and in many cases intersects other columns or SOL positions. The HWR implicitly learns during training when to stop reading (similar to [19]) and as a result we do not need additional post processing to determine when the line ends.

3.2 Post Processing

We introduce a novel non-maximal suppression method for the SOL and LF networks. Given any two LF path prediction we consider the first N steps (we used $N = 6$). We form a polygon by joining start and end points of the center lines. If the area of the resulting polygon is below a threshold proportional to its length, we suppress the line with the lowest SOL probability.

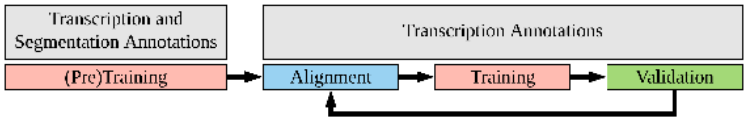


Fig. 5: Our network is first pre-trained on a small training set with segmentation and transcription annotations. The three phase training process is performed over a much larger training set that has only transcription annotations.

To correct recognitions errors we employ an HMM-based 10-gram character-level language model (LM) that has been trained on the training set transcriptions using the Kaldi toolkit [21]. Character-level LMs typically correct out-of-vocabulary words better than word-level LMs [16].

3.3 Training

Fig. 5 summarizes the full training process: (1) Networks are pretrained using a small number of images with GT SOL, segmentations, and line-level transcriptions (Sec. 3.3); (2) Alignment (Sec. 3.3) on a large number of training images with only GT transcriptions produces bootstrapped targets for the SOL and LF networks; (3) Individual networks are trained using SOL and LF targets from alignment and GT transcriptions for the HWR network; (4) Validation is performed over the entire validation set using the best individual weights of each network. Steps 2-4 are repeated until convergence.

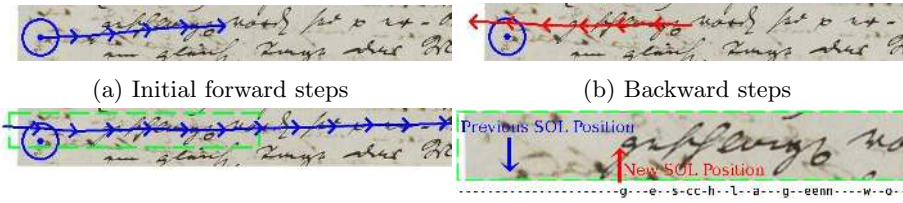
Start-of-Line Network We create the training set for our SOL network by resizing images to be 512 pixels wide and sampling 256x256 patches, with half the patches containing SOLs. Patches are allowed to extend outside the image by padding with each edge's average color. We use the loss function proposed for the multibox object detection model [8], which performs an alignment between the highest probability predicted SOL positions and the target positions.

$$L(l,p;t) = \sum_{n=0}^{N}\sum_{m=0}^{M} X_{nm}(\alpha\|l_n - t_m\|_2^2 - log(p_n)) - (1 - X_{nm})log(1 - p_n) \quad (4)$$

where t_m is a target position, p_n is the probability of SOL occurrence, and l_n is a transformation of the directly predicted $(x_n, y_n, s_n, \theta_n)$:

$$l_n = (-\sin(\theta_n)s_n + x_n,\ -\cos(\theta_n)s_n + y_n,\ \sin(\theta_n)s_n + x_n,\ \cos(\theta_n)s_n + y_n), \quad (5)$$

X_{nm} is a binary alignment matrix between the N predictions and M target positions, while α weights the relative importance of the positional loss and the confidence loss. In our experiments, $\alpha = 0.01$ and we compute the X_{nm} that minimizes L given (l, p, t) using bipartite graph matching as in [8].



(a) Initial forward steps (b) Backward steps

(c) Complete forward steps (d) Refined SOL prediction.

Fig. 6: SOL refinement process. In (b), the LF does not backtrack to the initial (incorrect) SOL. The LF passes through the correct SOL in (c), which is identified using the alignment (d) induced by CTC decoding in the HWR network.

Line Follower While the LF outputs a normalized text line image, the defining image transformation is piece-wise affine and is parameterized by a sequence of upper and lower coordinate points. Thus, for supervision we construct pairs of target coordinate points that induce the desired piece-wise affine transformation and train the LF using a Mean-Square Error (MSE) loss.

$$loss = \sum_{i=0} \|p_{u,i} - t_{u,i}\|_2^2 + \|p_{\ell,i} - t_{\ell,i}\|_2^2 \tag{6}$$

The LF starts at the first target points, $t_{u,0}$ and $t_{\ell,0}$, and every 4th step resets to the corresponding target points. This way, if the LF deviates from the handwriting it can recover without introducing large and uninformative errors into the training procedure. To help the LF be robust to incorrect previous predictions, after resetting to a target position we randomly perturb the LF position by a translation of $\Delta x, \Delta y \in [-2, 2]$ pixels and a rotation of $\Delta\theta \in [-0.1, 0.1]$ radians.

Handwriting Recognition We train the HWR network on line images with the aligned GT transcription using CTC loss [10]. For data augmentation, we apply Random Warp Grid Distortions (RWGD) [33] to model variations in handwriting shape, contrast augmentation [30] to learn invariance to text/background contrast, and global hue perturbation to handle different colors of paper and ink.

Pre-training Before joint training can be effective, each network needs to achieve a reasonable level of accuracy. Individual networks are pre-trained on a small number of images that have SOL, segmentation, and line-level transcription annotations. This follows the same procedure as described in the previous three subsections, but the actual GT is used for targets.

Alignment After the networks are pre-trained, we perform an alignment between SFR predicted line transcriptions with GT line transcriptions for images with only transcription annotations, i.e. no corresponding spatial GT information. The main purpose of this alignment is to create bootstrapped training

targets for the SOL and LF networks because the images lack GT for detection and segmentation. For each GT text line, we keep track of the best predicted SOL and segmentation points, where best is defined by the accuracy of the corresponding predicted line transcription produced by the HWR network.

Alignment and training are alternated (see Fig. 5) as better alignment improves network training and vice versa. To perform the alignment, we first run the SOL finder on the whole image and obtain dense SOL predictions. On predicted SOLs with probability above a threshold, we then apply the LF and HWR networks to obtain a predicted segmentation and transcription. For each GT line, we find the predicted transcription that minimizes the Character Error Rate (CER), which is equivalent to string edit distance. If the CER is lower than the best previous prediction for that GT line, we update that line's target SOL and segmentation points to be those predicted by the SOL and LF networks.

The final step in alignment is to refine the SOL position using spatial information extracted from the LF and HWR networks. To refine a SOL target, we run the LF forward $s = 5$ steps from the current best SOL (Fig. 6a), and then backwards for $s + 1$ steps (Fig. 6b). We then move the current best SOL up or down to align with the backwards path. This works because even if the LF does not start on the text line, it quickly finds the text line in the forward steps and then can follow it back to its start using backwards steps. Next, we run the LF and HWR from this new SOL and find the first non-blank predicted character before CTC decoding (Fig. 6d). We then shift the SOL left and right to align with the image location of this character.

To find the end of the handwriting line, we find the last non-blank character during CTC decoding. Once we have identified line ends, we no longer run the LF past the end of lines, which helps speed training.

End-to-end Training Though our SFR model is end-to-end differentiable in that the CTC loss can backpropagate through the HWR and LF networks to the SOL network, in practice we observed no increase in performance when using end-to-end training on the dataset used in this work. End-to-end training is much slower, and the three networks take significantly different amounts of time to train, with the HWR network taking the most time by far. We have concluded that the majority of errors made by our SFR model are not likely to be fixed by end-to-end error backpropagation because (1) the transcription CTC loss cannot fix very bad segmentations and (2) our joint training provides adequate supervision when predicted SOL and segmentations are reasonably good.

4 Results

We evaluate our SFR model on the 2017 ICDAR HWR full page competition dataset [25] of 1800s German handwriting, which has two training sets. The first set has 50 fully annotated images with line-level segmentations and transcriptions. The second set of 10,000 images has only transcriptions (containing line breaks). This dataset, to our knowledge, is the largest and most challenging

Table 1: ICDAR 2017 HWR Competition results [25] compared to our method.

Method	BLEU with ROIs	BLEU without ROIs
Start, Follow, Read (ours)	**73.0**	**72.3**
BYU	71.5	57.3
ParisTech	48.3	-
LITIS	37.2	-

Table 2: Line-level dataset results. * indicates non-standard train/test split.

Method	Page-level	RIMES		IAM	
		CER	WER	CER	WER
Start, Follow, Read (ours)	X	**2.1**	**9.3**	6.4	23.2
Bluche[3]	X	2.9	12.6	7.9	24.6
Puigcerver [34]		2.3	9.6	**5.8***	**18.4***

public HWR benchmark with 206,161 handwriting lines and 1,769,195 words. The test data is not public, so we use the BLEU score metric reported by the public evaluation server[3]. The competition test data provides multiple regions of interest (ROIs) per image to facilitate text line segmentation, and the evaluation server protocol requires that all predicted text lines be assigned to a ROI. We also evaluate on the IAM and Rimes line-level datasets.

4.1 Quantitative Results

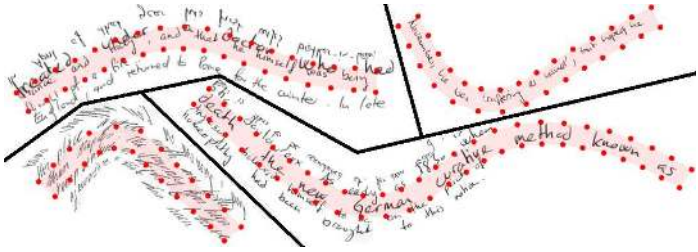
The fully annotated 50 images are used to pre-train the network (see Fig. 5). We then jointly train on 9,000 images (1,000 for validation) by alternating alignment, training, and validation steps. We then submitted two sets of predictions to the evaluation server: one set exploiting the ROI information and one set without. To exploit ROI information, we mask out all other parts of the image using the median image color before running SFR.

Though we also evaluate without ROIs, the evaluation server still requires each line to be assigned to a ROI. After running SFR on full pages (no masking), we simply assign each line prediction to the region in which it has the most overlap. Predictions mostly outside any ROI are discarded, though sometimes these are real unannotated text lines that are completely outside the given ROIs.

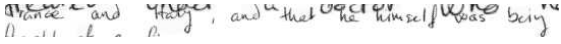
The competition systems made predictions over each ROI by first cropping to the ROI bounding box [25]. The BYU system was evaluated without ROIs using the same process as SFR except lines are only discarded if they intersect no ROI. This difference was necessary because their segmentations span the entire image and too many good text lines would have been discarded.

Table 1 compares SFR with the competition results. Our SFR model achieves the highest BLEU score at 73.0 using ROI annotations, but performance only

[3] https://scriptnet.iit.demokritos.gr/competitions/~icdar2017htr/



(a) LF on warped IAM lines with upper and lower lines as distraction. SOL positions were provided.



(b) Deskewed line. Upper left

(c) Deskewed line. Lower left

Fig. 7: Results from warped IAM dataset.

degrades slightly to 72.3 without ROIs. This shows that the SOL and LF networks perform well and do not benefit much from a priori knowledge of text line location. In contrast, the winning competition system scores 71.5 using the ROIs, but its performance drops significantly to 57.3 without the ROIs.

Table 2 shows results for the IAM (English) and RIMES (French) line-level datasets. Like [3], we evaluated our page-level method on line-level datasets where we do not use the provided line segmentation annotations during training or evaluation, except for 10 pretraining images. We achieved state-of-the-art results on RIMES, outperforming [22] which uses the segmentation annotations for training and evaluation. On IAM, we outperformed the best previously proposed page-level model [3], and we note that [22] used a non-standard data split, so their results are not directly comparable. Results shown in Table 2 are without LM decoding, so that the raw recognition models can be fairly compared.

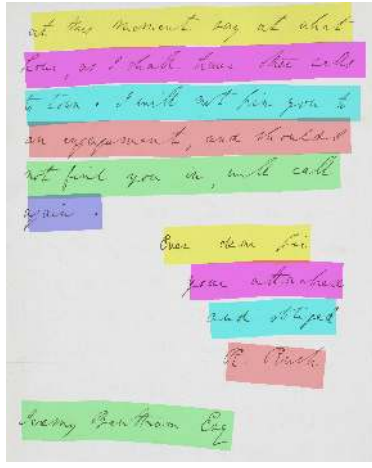
4.2 Qualitative Results

We produced a synthetic dataset to test the robustness of the LF on very curved lines. To generate the data we randomly warped real handwriting lines from the IAM dataset [18] and added distracting lines above and below. We provided the SOL position and did not employ the HWR. Fig. 7 shows results from the validation set. Even when text lines are somewhat overlapping (Fig 7b), the LF is able to stay on the correct line. Though the synthetic warping is exaggerated, this suggests the LF can learn to follow less extreme real-world curvature.

Fig. 9 shows some results on our ICDAR2017 HWR dataset validation set. On clean images, SFR often produces a perfect transcription (Fig. 9a), and only minor errors on noisy handwriting (Fig. 9b). The LF performs well on complicated layouts, such as horizontally adjacent lines (Fig. 9c). However, some noisy lines cause the LF to jump between lines. (Fig. 9d).

We also applied the trained SFR model to other image datasets and found that the SOL and LF networks generalize even to documents in different lan-

(a) Document written in the 1400s from the 2016 ICFHR HWR competition [29]



(b) English document from the ICDAR competition on baseline detection[7]

Fig. 8: Images from other collections applied to our trained model

guages. Fig. 8a shows that SFR correctly segments a document written in Early Modern German and we see similar results on a English document (Fig. 8b). Of course, the HWR network would need to be retrained to handle other languages, though due to the modularity of SFR, the HWR network can be retrained while preserving the previous SOL and LF networks. Additional images can be viewed in the supplementary material.

5 Conclusion

We have introduced a novel Start, Follow, Read model for full-page HWR and demonstrated state-of-the-art performance on a challenging dataset of historical handwriting, even when not exploiting given ROI information. We improved upon a previous SOL method and introduced a novel LF network that learns to segment and normalize handwriting lines for input to a HWR network. After initial pre-training, our novel training framework is able to jointly train the networks on documents using only line-level transcriptions. This is significant because when human annotators transcribe documents, they often do not annotate any segmentation or spatial information.

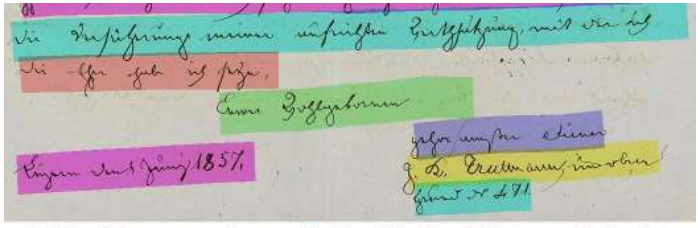
We believe that further improvements can be made by predicting the end-of-line (EOL), in addition of SOL, and applying the LF backwards. Then, the SOL and EOL results can mutually constrain each other and lead to improved segmentation. Also, we did not extensively explore network architectures, so performance could increase with improved architectures such as Residual Networks.

dispositions que vous avez données à fin
que nos Ingenieurs pouissent avoir les
pièces necessaires pour accomplir leur

(a) No errors

eindringlichsten Zureden für jezige angesprochene
Hülfe, auch deutlich auf die Verpflichtung dessin's
hinzuweisen, die der Gotthardkomite schuldig.
Anerkennung für die so volle Aufnahme & Wohl

(b) Noisy lines, few transcription errors



die Versicherung meiner aufrichten Werthschätzung, mit der ich
die Ehre habe ich setze,
Huerrn Wohlgebornen
gehorsamster Sinner
Luzern den 1 Juny 1857.
J. K. Truttmann, in oben
Grund No 471.

(c) Complex layout, few transcription errors

der mehrerwähnten Stelle Ihres Buches zu Haun-
de liegen, zukommen lassen zu wollen &
Wenn ich Sie im fernern ersuchen muß, die-
ser gefälligst mit möglichster Beförderung

(d) Noisy lines, LF error. HWR stopped reading after the error.

Fig. 9: Results from the ICDAR 2017 competition dataset. Colored lines represent different detected lines. Green, red, and purple characters represent insertion, substitution, and omission errors respectively.

References

1. Antonacopoulos, A., Karatzas, D.: Document image analysis for World War II personal records. In: Workshop on Document Image Analysis for Libraries. pp. 336–341. IEEE (2004)
2. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. In: ACM SIGGRAPH 2007 Papers. SIGGRAPH '07, ACM (2007). https://doi.org/10.1145/1275808.1276390
3. Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In: Advances in Neural Information Processing Systems (NIPS). pp. 838–846 (2016)
4. Bluche, T., Louradour, J., Messina, R.: Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention (04 2016)
5. Boiangiu, C.A., Tanase, M., Ioanitescu, R.: Handwritten documents text line segmentation based on information energy. International Journal of Computers, Communications and Control (IJCCC) **9**, 8–15 (12 2014)
6. Bunke, H., Bengio, S., Vinciarelli, A.: Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **26**(6), 709–720 (2004)
7. Diem, M., Kleber, F., Fiel, S., Grüning, T., Gatos, B.: cBAD: ICDAR2017 competition on baseline detection. In: 14th International Conference on Document Analysis and Recognition (ICDAR). pp. 1355–1360. IEEE (2017)
8. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. CoRR **abs/1312.2249** (2013), http://arxiv.org/abs/1312.2249
9. Frinken, V., Fischer, A., Martínez-Hinarejos, C.D.: Handwriting recognition in historical documents using very large vocabularies. In: 2nd International Workshop on Historical Document Imaging and Processing (HIP). pp. 67–72. ACM (2013)
10. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: 23rd International Conference on Machine Learning. pp. 369–376. ACM (2006)
11. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **31**(5), 855–868 (2009)
12. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 545–552 (2009)
13. Ha, J., Haralick, R.M., Phillips, I.T.: Document page decomposition by the bounding-box project. In: 3rd International Conference on Document Analysis and Recognition (ICDAR). vol. 2, pp. 1119–1122. IEEE (1995). https://doi.org/10.1109/ICDAR.1995.602115
14. He, J., Downton, A.C.: User-assisted archive document image analysis for digital library construction. In: International Conference on Document Analysis and Recognition. pp. 498–502. IEEE (2003)
15. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 2017–2025 (2015)

16. Kozielski, M., Rybach, D., Hahn, S., Schlter, R., Ney, H.: Open vocabulary handwriting recognition using combined word-level and character-level language models. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8257–8261 (May 2013). https://doi.org/10.1109/ICASSP.2013.6639275
17. Lorigo, L.M., Govindaraju, V.: Offline arabic handwriting recognition: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **28**(5), 712–724 (2006)
18. Marti, U.V., Bunke, H.: The IAM-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
19. Moysset, B., Kermorvant, C., Wolf, C.: Full-page text recognition: Learning where to start and when to stop. In: 14th International Conference on Document Analysis and Recognition (ICDAR). pp. 871–876. IEEE (2017). https://doi.org/10.1109/ICDAR.2017.147
20. Plötz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. International Journal on Document Analysis and Recognition (IJDAR) **12**(4), 269 (2009)
21. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (Dec 2011), iEEE Catalog No.: CFP11SRW-USB
22. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 14th International Conference on Document Analysis and Recognition (ICDAR). pp. 67–72. IEEE (Nov 2017). https://doi.org/10.1109/ICDAR.2017.20
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (June 2017)
24. Saabni, R., El-Sana, J.: Language-independent text lines extraction using seam carving. In: 11th International Conference on Document Analysis and Recognition (ICDAR). pp. 563–568. IEEE (2011)
25. Sanchez, J.A., Romero, V., Toselli, A.H., Villegas, M., Vidal, E.: ICDAR2017 competition on handwritten text recognition on the READ dataset. In: 14th International Conference on Document Analysis and Recognition (ICDAR). pp. 1383–1388. IEEE (Nov 2017). https://doi.org/10.1109/ICDAR.2017.226, doi.ieeecomputersociety.org/10.1109/ICDAR.2017.226
26. Shapiro, V., Gluhchev, G., Sgurev, V.: Handwritten document image segmentation and analysis. Pattern Recognition Letters **14**(1), 71–78 (1993)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014), http://arxiv.org/abs/1409.1556
28. Smith, R.: Tutorial: Tesseract blends old and new OCR technology (2016)
29. Snchez, J.A., Romero, V., Toselli, A.H., Vidal, E.: ICFHR2016 competition on handwritten text recognition on the READ dataset. In: 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 630–635. IEEE (Oct 2016). https://doi.org/10.1109/ICFHR.2016.0120
30. Tensmeyer, C., Saunders, D., Martinez, T.: Convolutional neural networks for font classification. In: 14th International Conference on Document Analysis and Recognition (ICDAR). pp. 985–990. IEEE (Nov 2018).

https://doi.org/10.1109/ICDAR.2017.164, doi.ieeecomputersociety.org/10.1109/ICDAR.2017.164

31. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. CoRR **abs/1609.03605** (2016), http://arxiv.org/abs/1609.03605
32. Vinciarelli, A.: A survey on off-line cursive word recognition. Pattern recognition **35**(7), 1433–1446 (2002)
33. Wigington, C., Stewart, S., Davis, B., Barrett, W., Price, B., Cohen, S.: Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network. 14th International Conference on Document Analysis and Recognition (ICDAR) pp. 639–645 (2017)
34. Zamora-Martinez, F., Frinken, V., España-Boquera, S., Castro-Bleda, M.J., Fischer, A., Bunke, H.: Neural network language models for off-line handwriting recognition. Pattern Recognition **47**(4), 1642–1652 (2014)