Smart Library: Identifying Books on Library Shelves using Supervised Deep Learning for Scene Text Reading

Xiao Yang, Dafang He, Wenyi Huang, Alexander Ororbia
Zihan Zhou, Daniel Kifer, C. Lee Giles
The Pennsylvania State University, University Park, PA 16802, USA
{xuy111, duh188}@psu.edu, harrywy@gmail.com, {ago109,zzhou}@ist.psu.edu,
dkifer@cse.psu.edu, giles@ist.psu.edu

ABSTRACT

Physical library collections are valuable and long standing resources for knowledge and learning. However, managing and finding books or other volumes on a large collection of bookshelves often leads to tedious manual work, especially for large collections where books or others might be missing or misplaced. Recently, deep neural-based models have been successful in detecting and recognizing text in images taken from natural scenes. Based on this, we investigate deep learning for facilitating book management. This task introduces further challenges including image distortion and varied lighting conditions. We present a library inventory building and retrieval system based on scene text reading. We specifically design our text recognition model using rich supervision to accelerate training and achieve state-of-the-art performance on several benchmark datasets. Our proposed system has the potential to greatly reduce the amount of manual labor required for managing book inventories.

1 INTRODUCTION

Despite the increasing availability of digital books, many still favor reading physical books and not all books or volumes have been digitized. The large libraries that house them require great amounts of time and labor to manage inventories that number in the millions. Manually searching bookshelves is time-consuming and can be unfruitful depending on how vague the search is. To solve this problem, we propose a deep neural network-based system to automatically localize, recognize and index text on bookshelves images.

We first process bookshelves images to localize and recognize book spine text so as to build a digital book inventory. Then, we utilize this digital inventory to help users quickly locate a book or volume they are looking for. Our pipeline is summarized in Figure 1.

Our contributions are as follows: 1) we build a scene text reading system specifically designed for book spine reading and library inventory management. We demonstrate that the book spine text information extracted by our system alone can achieve good retrieval performance. This is essential, since other types of data, like digital images of all book covers in a collection, are not necessarily available to all users. 2) For text recognition, we adopt a deep sequential labeling model based on convolutional neural nets (CNN) and recurrent neural nets (RNN). We propose using a per-timestep classification loss in tandem with a weighted Connectionist Temporal Classification (CTC) loss function [3] in order to accelerate training and improve performance.

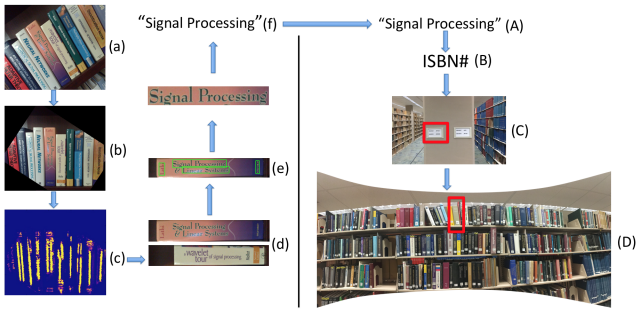

Figure 1: System pipeline. (a)-(f) correspond to building a book inventory while (A)-(D) correspond to locating a book in a library. (a) Original image. (b) Rotated image based on estimation of dominant orientation. (c) Saliency image. (d) Segmented book spines. (e) Detected lines of words. (f) Detected words. (A) Query keywords. (B) Corresponding ISBN number. (C) Location of the stack the book belongs to. (D) Location of the book in the shelf.

2 RELATED WORK

Previous work on book inventory management has typically focused on book spine detection and retrieval such as the framework in [13] using high-frequency filtering and thresholding. A Hough Transform based book boundary detector [2] was designed to extract features to retrieve books in an inventory. Nevetha et. al. [12] used a line segment detector with several heuristic rules to extract book spines with Optical Character Recognition (OCR) then applied to read text. A quantized color histogram of a book spine image [11] was used as features to search bookshelves. A hybrid method [17] combined a text-reading method with an image-matching one.

It is important to note that the performance of most existing approaches is limited by book spine segmentation and off-the-shelf OCR systems. Handcrafted features based book spine segmentation suffers from image distortion and low contrast

JCDL'17, Toronto, Ontario, Canada

DOI:

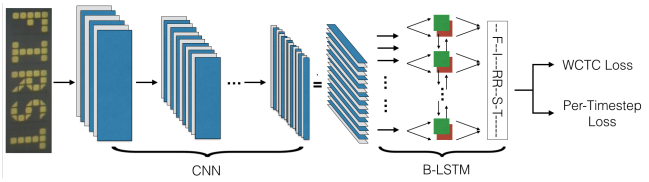

Figure 2: Architecture of the proposed framework where the CNN architecture is similar to VGG16 [15].

between books. Off-the-shelf OCR systems, such as Tesseract [16], perform poorly on images taken in natural scenes. Recently, scene text reading has become popular in computer vision [6, 9, 14]. Here, we present a deep neural network based system that reads scene text and show that scene text reading can be effectively utilized for the purpose of book inventories management and book retrieval. Combined with other image processing techniques such as Hough Transform, our system achieves robust performance on book retrieval task.

3 TEXT LOCALIZATION

We describe our method for detecting text in images of library bookshelves. We first segment each book spine image and then localize text on these images.

Book spine segmentation is a critical component of our system since every book is expected to be indexed and queried independently. Most existing methods rely exclusively on low-level methods, such as the Hough Transform, for segmentation. In contract, we only use the Hough Transform as a pre-processing step for extraction of the dominant direction of book spines, which is later used to rotate the entire image (Figure 1(b)).

After rotating the image, we apply a text/non-text CNN model to the input image in a sliding window manner to generate saliency maps (Figure 1(c)). The saliency maps can be further used to: 1) detect book title locations, and 2) segment each book. We adopt a non-max suppression method to find the segmenting point for each book along horizontal direction. As such, we circumvent the use of a Hough Transform or other low-level routines, which are sensitive to lighting conditions and low contrast.

A scene text localization algorithm based on a CNN is subsequently applied to each book spine image. This step further detects individual words on book spines. We refer the reader to[5] for details.

4 TEXT RECOGNITION

In our system, book spine images are identified based on the recognized text, which is then used for indexing and searching from a book database.

4.1 Text Recognition via Sequence Labeling

For text recognition, a conventional approach is to first segment and recognize each character, then predict a word based on a language model or a combination of heuristic rules. However, these approaches are highly sensitive to various distortions in images causing character-level segmentation imperfections. To bypass the character segmentation step, we cast text recognition as a sequential labeling task, recognizing a sequence of characters simultaneously.

Similar to [6, 14], our model consists of a CNN and a RNN as shown in Figure 2. We first learn a sequence of deep CNN features $F = \{f_1, f_2, \cdots, f_T\}$ from an image I. To further exploit the interdependence among features, a bidirectional Long Short-Term Memory (B-LSTM) [7] is applied on top of the learned sequential CNN features, yielding another sequence $X = \{x_1, x_2, \cdots, x_T\}$ as final outputs. Each x_i is normalized through a softmax function and can be interpreted as the emission of a character or a blank label at a specific time-step. On the other hand, the target word Y can also be viewed as a sequence (of characters): $Y = \{y_1, y_2, \cdots, y_L\}$. Since sequences X and Y have different lengths, we adopt CTC loss [3] to allow an RNN to be trained. Stochastic gradient descent (SGD) method is used for optimization. The gradient of the CTC loss can be efficiently computed using a forward-backward dynamic programming method [4]. Decoding (finding the most likely Y from the output sequence X) can be done by beam search.

4.2 CTC Training with Per-Timestep Supervision

During the CTC training process, blank labels typically dominant the output sequence. Non-blank labels only appear as isolated peaks (see Figure 3). This is a consequence of the forward-backward algorithm [3]. Since we add a blank label between each character, there are more possible paths going through a blank label at a given timestep in the CTC forward-backward graph. In the early stage of CTC training where model weights are randomly initialized, all paths have similar probabilities. As a result, the probability of a given timestep being a blank label is much higher than any other kinds of labels when summing up all valid paths in CTC graph.

Owing to the blank label issue described above, it generally takes many iterations before a non-blank label appears in the output sequence during training. To accelerate training, we introduce per-timestep supervision. If character-level bounding boxes are available, we can decide the label of x_i at each timestep i, based on its receptive field. In our experiments, x_i is assigned a label $z_i = y_j$ if its receptive field overlaps with more than half of the area of character y_j, otherwise z_i =blank label. The objective function becomes:

$$L_1(X) = CTC(X) + \lambda L_{pt}(X) \tag{1}$$

$$L_{pt}(X) = \frac{1}{T}\sum_{i=1}^{T} -\log P(z_i|x_i) \tag{2}$$

where λ is a hyper-parameter meant to balance the two terms. Since our per-timestep supervision only provides one possible kind of alignment, we decrease λ throughout training. At the start λ is set such that the gradients yielded by the two kinds of losses have the same magnitude.

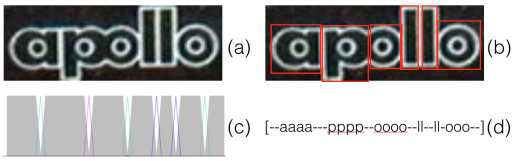

Figure 3: (a) An example word image. (b) Character-level bounding boxes. (c) A typical output sequence from a CTC-trained model where blank labels (gray area) dominant. (d) A per-timestep groundtruth generated based on (b).

4.3 CTC Training with a Decoding Penalty

Another issue of CTC training is the gap between the objective function and the evaluation criterion. CTC loss will try to maximizes the log probability of outputting *completely correct* labels $Y = \{y_1, y_2, \cdots, y_L\}$. In another words, incorrect predictions are treated as equally bad. However, this is not always the way model performance is assessed. For example, for text recognition edit distance is often reported. Graves et. al. [4] proposed a sample-based methods to calculate the expected loss. However, this sampling step significantly slowed down training.

As such, we propose a simpler solution to penalize bad predictions. A weighted CTC is introduced:

$$WCTC(x) = -\log P(Y|X) \cdot L_e(Y, Y_D) \quad (3)$$

$$L_2(x) = WCTC(x) + \lambda L_{pt}(X) \quad (4)$$

where $L_e(\cdot, \cdot)$ is a real-value loss function (e.g. edit distance between two strings) and Y_D is the decoded prediction using beam search.

5 EXPERIMENTS

5.1 Text Recognition

To assess the performance of our text recognition, we report results on three widely-used benchmark datasets: IC03, SVT and III5K using a standard evaluation protocol [18]. Each image is associated with a lexicon containing 50 or 1,000 candidate words for the purpose of refining model predictions.

We refer to our base model, trained using the CTC loss, as Deep Sequential Labeling (DSL-base), where the models trained using $L_1(x)$ and $L_2(x)$ losses will be referred to respectively as DSL-L_1 and DSL-L_2.

Figure 4 shows the loss curves during training. As we can see, adding per-timestep classification loss would significantly speedup training at early stage. At later stage, as λ becomes smaller and smaller, the difference between with and without L_{pt} becomes marginal. However, from test set we can still observe performance gain when using L_{pt}.

Table 1 shows text recognition results. We can see that the OCR engine Tesseract performs poorly on all datasets. Our recognition models outperform methods with handcrafted-features [10, 20] and several deep neural-based methods focusing individual characters [9, 19], indicating the benefits of learning sequential information. [8] achieves the best results on IC03-50. However, since they treat text recognition as a multi-class (number of classes equals number of words) classification task, it is impossible for their model to adapt to

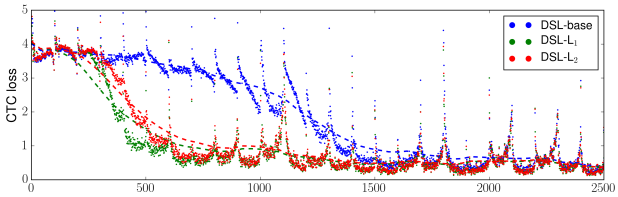

Figure 4: CTC loss during training using different objective functions. Dotted curves are the training loss while solid curves are the validation loss. Best viewed in color.

Table 1: Cropped word recognition accuracy across several benchmark datasets.

Method	Recognition Accuracy(%)			
	IC03-50	SVT-50	III5K-50	III5K-1k
Tesseract [16]	60.1	65.9	-	-
Lee 2014 [10]	88.0	80.0	-	-
Yao 2014 [20]	88.5	75.9	80.2	69.3
Wang 2012 [19]	90.0	0.0	-	-
Jaderberg 2014 [9]	96.2	86.1	-	-
Jaderberg 2014b [8]	**98.7**	95.4	97.1	92.7
Shi 2016 [14]	**98.7**	**96.4**	97.6	94.4
Our DSL-base	96.1	94.5	97.7	95.1
Our DSL-L_1	96.6	94.5	98.3	95.9
Our DSL-L_2	98.2	94.6	**98.5**	**96.0**

out-of-dictionary text. [14] achieves better results on IC03-50 and SVT-50 than our DSL-base, despite that they share similar model architecture and training procedure. We attribute this to the fact that we use a much smaller training set. Both [8] and [14] use 8 million synthetic images for training while we only use 0.7 million. Yet we achieve the best results on III5K dataset, which contains more testing images (3003) than IC03 (865) and SVT (647).

Both DSL-L_1 and DSL-L_2 perform better than DSL-base. We hypothesize that adding per-timestep loss reduce peak prediction phenomenon, which would confuse the model about where to yield non-blank predictions. DSL-L_2 ties or slightly outperforms DSL-L_1 on all datasets, suggesting that our revised loss $WCTC(X)$ is effective.

5.2 Book Spine Retrieval

To assess the retrieval performance, we adopt the retrieval-based evaluation method similar to that of [2] and [17]. However, since we only have access to the 454 book spine images they used for querying instead of the entire database to search from which contains 2,300 books, it is necessary to build our own collection. Therefore we crawled and sampled 9,100 books that are from the same library and in similar areas to those 454 books. For each book, we crawled and indexed its title and meta-data such as the author and publisher. We expect that the our collection is a superset of theirs, which would mean that higher precision and recall from our results suggest superior performance.

We first obtain book spine images using the approach described above. For each book spine image, text is detected and

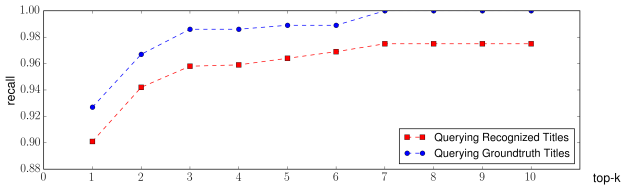

Figure 5: Recall at top-k during retrieval by querying recognized titles and groundtruth titles, respectively.

recognized. The outputs are further refined by matching to a dictionary from our database using nearest neighbor matching. Finally, we use these outputs as keywords to search from our database. During search, *tf-idf* (term frequency-inverse document frequency) weights are used to rank returned results. We built our search engine with Apache/Solr [1] which means it can easily scale to a large collection of books and volumes.

As in [2] and [17], we report the precision and recall when querying the 454 spine images. We further report recall at top-k, which measures the number of correctly identified books to appear in top-k results of a search. All these measures are widely used by the information retrieval community.

Table 2 shows our results compared with [17]. Numbers are extracted from their paper's precision/recall curve that yield best F-score. Using only textual information, we achieve the best recall and a much higher (0.91 VS 0.72) F-score. Given that our database is much larger than theirs (9,100 VS 2,300), the results show the better performance of our proposed method. [17] achieved 0.97, 0.86 and 0.91 for precision, recall and F-score respectively in their hybrid model. However, their hybrid model uses both text and image as queries, which requires much more processing time. Moreover, for all libraries one cannot always assume that book images are already available.

Figure 5 shows recall at different top-k rank. Our model achieves 96.4% recall within the top 5 search results. Further investigation of the failure cases found that a large portion of wrong predictions were due to the fact that multiple books may have similar or even identical titles. As such even using groundtruth titles as keywords in search cannot guarantee 100% recall at top-1 rank position. The results can be further improved by detecting and recognizing additional meta-information on the book spine such as publisher or author. Although image-based search might address this issue, it comes with the cost of storing and matching images. The rest of the failure cases are largely due to incorrect text localization.

6 CONCLUSION

We propose a scene text detection and recognition system for identifying books in a bookshelf library and building a digital library inventory. We achieve state-of-the-art performance for scene text recognition and at the same time reduce training time. Information retrieval experiments were conducted on a large physical library database. Performance on the whole system demonstrates that text-based retrieval is competitive with image-matching retrieval, and that text-based retrieval reduces the need for storing or matching book spine images. Finally,

Table 2: Precision and recall compared with another method, using only text as queries.

	Precision	Recall	F-score
Tsai2011 (Text)	**0.92**	0.60	0.72
Ours	**0.92**	**0.90**	**0.91**

we speculate that this method could help to find the location of books on the bookshelves of scholars and researchers.

ACKNOWLEDGMENTS

We gratefully acknowledge partial support from NSF grant CCF 1317560 and a hardware grant from NVIDIA.

REFERENCES

[1] Apache Solr. (????). http://lucene.apache.org/solr/.
[2] David M Chen, Sam S Tsai, Bernd Girod, Cheng-Hsin Hsu, Kyu-Han Kim, and Jatinder Pal Singh. 2010. Building book inventories using smartphones. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM.
[3] Alex Graves, Santiago Fernández, Faustino J Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23th International Conference on Machine Learning (ICML-06)*. 369–376.
[4] Alex Graves and Navdeep Jaitly. 2014. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1764–1772.
[5] Dafang He, Xiao Yang, Zihan Zhou, Daneil Kifer, and Lee C Giles. 2016. Aggregating Local Context for Accurate Scene Text Detection. In *Asian Conference on Computer Vision*. Springer, 91–105.
[6] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. 2016. Reading scene text in deep convolutional sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 3501–3508.
[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[8] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227* (2014).
[9] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep features for text spotting. In *European conference on computer vision*. Springer.
[10] Chen-Yu Lee, Anurag Bhardwaj, Wei Di, Vignesh Jagadeesh, and Robinson Piramuthu. 2014. Region-based discriminative feature pooling for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4050–4057.
[11] DJ Lee, Yuchou Chang, JK Archibald, and C Pitzak. Matching book-spine images for library shelf-reading process automation. In *2008 IEEE International Conference on Automation Science and Engineering*.
[12] MP Nevetha and A Baskar. 2015. Automatic book spine extraction and recognition for library inventory management. In *Proceedings of the Third International Symposium on Women in Computing and Informatics*. ACM.
[13] Nguyen-Huu Quoc and Won-Ho Choi. 2009. A framework for recognition books on bookshelves. In *International Conference on Intelligent Computing*. Springer.
[14] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on PAMI* (2016).
[15] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).
[16] Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, Vol. 2. IEEE, 629–633.
[17] Sam S Tsai, David Chen, Huizhong Chen, Cheng-Hsin Hsu, Kyu-Han Kim, Jatinder P Singh, and Bernd Girod. 2011. Combining image and text features: a hybrid approach to mobile book spine recognition. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 1029–1032.
[18] Kai Wang, Boris Babenko, and Serge Belongie. 2011. End-to-end scene text recognition. In *Proceedings of the 2011 International Conference on Computer Vision*. IEEE Computer Society, 1457–1464.
[19] Tao Wang, David J Wu, Andrew Coates, and Andrew Y Ng. 2012. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 3304–3308.
[20] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4042–4049.