Text Recognition with Neural Network

Aesam Alam[1], Gaurav Mitra[2], Syed Ziauddin[3], Souvik Datta[4]

[1]aa2309@cse.jgec.ac.in, [2]gk2359@cse.jgec.ac.in, [3]sz2386@cse.jgec.ac.in,

[4]souvik.datta@cse.jgec.ac.in

[1,2,3,4]Jalpaiguri Government Engineering College

Abstract

Text recognition in natural images is a challenging problem that has received much attention recently. Traditional systems in this area have relied on elaborate models incorporating carefully hand engineered features or large amounts of prior knowledge. In this paper, we take a different route and combine the representational power of large, multilayer neural networks together with recent developments in unsupervised feature learning, which allows us to use a common framework to train highly-accurate text detector and character recognizer modules. Then, using only simple off-the-shelf methods, we integrate these two modules into a full end-to-end, lexicon-driven, scene text recognition system that achieves state-of-the-art performance on standard benchmarks, namely Street View Text and ICDAR.

Keywords: Text recognition, Neural network, Convolutional Technique, Python, OpenCV.

1 Introduction

Extracting textual information from natural images is a challenging problem with many practical applications. Unlike character recognition for scanned documents, recognizing text in unconstrained images is complicated by a wide range of variations in backgrounds, textures, fonts, and lighting conditions. As a result, many text detection and recognition systems rely on cleverly hand-engineered features [5, 4, 14] to represent the underlying data. Sophisticated models such as conditional random fields [11, 19] or pictorial structures [18] are also often required to combine the raw detection/recognition outputs into a complete system. In this paper, we attack the problem from a different angle. For low-level data representation, we use an unsupervised feature learning algorithm that can automatically extract features from the given data. Such algorithms have enjoyed numerous successes in many related fields such as visual recognition [3] and action recognition [7]. In the case of text recognition, the system in [2] achieves competitive results in both text detection and character recognition using a simple and scalable feature learning architecture incorporating very little hand-engineering and prior knowledge.

We integrate these learned features into a large, discriminatively-trained convolutional neural network (CNN). CNNs have enjoyed many successes in similar problems such as handwriting recognition [8], visual object recognition [1], and character recognition [16]. By leveraging the representational power of these networks, we are able to train highly accurate text detection and character recognition modules. Using these modules, we can build an end-to-end system with only simple post-processing techniques like non-maximal suppression (NMS) [13] and beam search [15]. Despite its simplicity, our system achieves state-of-the-art performance on standard test sets.

2 Learning Architecture

In this section, we describe our text detector and character recognizer modules, which are the essential building blocks of our full end-to-end system. Given a 32-by-32pixel window, the detector decides whether the window contains a centered character. Similarly, the recognizer decides which of 62 characters (26 uppercase, 26 lowercase letters, and 10 digits) is in the window. As described at length in Section 3, we slid the detector across a full scene image to identify candidate lines of text, on which we perform word-level segmentation and recognition to obtain the end-to-end results.

For both detection and recognition, we use a multilayer, convolutional neural network (CNN) similar to [8, 16]. Our networks have two convolutional layers with n1 and n2 filters respectively. The network we use for detection with n1 = 96 and n2 = 256 is shown in Figure 1, while a larger, but structurally identical one (n1 = 115 and n2 = 720) is used for recognition.

We train the first layer of the network with an un supervised learning algorithm similar to [2, 3]. In particular , given a set of 32-by-32 grayscale training images1 as illustrated in Figure 2, we randomly extract m 8-by-8 patches, which are contrast normalized and ZCA whitened [6] to form input vectors x (i) ∈ R 64, i ∈ {1, ..., m}. We then use the variant of K-means described in [2] to learn a set of low-level filters D ∈ R 64×n . For a single normalized and whitened 8-by-8 patch x, we compute its first layer responses z by performing inner product with the filter bank followed by a scalar activation function: z = max{0, |DTx| − α}, where α = 0.5 is a hyperparameter.

Given a 32-by-32 input image, we compute z for every 8-by-8 sub-window to obtain a 25-by-25-by-n1 first layer response map. As is common in CNNs, we average pool over the first layer response map to bring its dimensions to 5-by-5-by-n1. We stack another convolution and average pooling layer on top of the first layer to obtain a 2-by-2-by-n2 second layer response map. These outputs are fully connected to the classification layer. We discriminatively train the network by backpropagating the L2-SVM classification error,2 but we fix the filters in the first convolution layer (learned from K-means). Given the size of the networks, fine-tuning is performed using multiple GPUs.

3 End-to-End Pipeline Integration

Our full end-to-end system combines a lexicon with our detection/recognition modules using post processing techniques including NMS and beam search. Here we assume that we are given a lexicon (a list of tens to hundreds of candidate words) for a particular image. As argued in [18], this is often a valid assumption as we can use prior knowledge to constrain the search to just certain words in many applications. The pipeline mainly involves the following two stages: (i) We run sliding window detection over high resolution input images to obtain a set of candidate lines of text. Using these detector responses, we also estimate locations for the spaces in the line. (ii) We integrate the character responses with the candidate spacings using beam search [15] to obtain full end-to-end results. First, given an input image, we identify horizontal lines of text using multiscale, sliding window detection. At each scale s, we evaluate the detector response Rs[x, y] at each point (x, y) in the scaled image. As shown in Figure 3, windows centered on single char acters at the right scale produce positive Rs[x, y]. We apply NMS [13] to Rs[x, r] in each individual row r to estimate the character locations on a horizontal line. In particular, we define the NMS response

4 Experimental Results

In this section we present a detailed evaluation of our text recognition pipeline. We measure cropped character and word recognition accuracies, as well as end-to end text recognition performance of our system on the ICDAR 2003 [10] and the Street View Text (SVT) [18] datasets. Apart from that, we also perform additional analysis to evaluate the importance of model size on different stages of the pipeline. First we evaluate our character recognizer module on the ICDAR 2003 dataset. Our 62-way character classifier achieves state-of-the-art accuracy of 83.9% on cropped characters from the ICDAR 2003 test set. The best known previous result on the same benchmark is 81.7% reported by [2] Our word recognition sub-system is evaluated on images of perfectly cropped words from the ICDAR 2003 and SVT datasets. We use the exact same test setup as [18]. More concretely, we measure word-level accuracy with a lexicon containing all the words from the ICDAR test set (called I-WD), and with lexicons consisting of the ground truth words for that image plus 50 random "distractor" words added from the test set (called I-WD-50). For the SVT dataset, we used the provided lexicons to evaluate the accuracy (called SVT WD). Table 1 compares our results with [18] and the very recent work of [11]. We evaluate our final end-to-end system on both the ICDAR 2003 and SVT datasets, where we locate and recognize words in full scene images given a lexicon. For the SVT dataset, we use the provided lexicons; for the ICDAR 2003 dataset, we used lexicons of 5, 20 and 50 distractor words provided by the authors of [18], as well as the "FULL" lexicon consisting of all words in the test set. We call these benchmarks I-5, I-20, I-50 and I-FULL respectively. Like [18], we only consider alphanumeric words with at least 3 characters. Figure 5 shows some sample outputs of our system. We follow the standard evaluation criterion described in [10] to compute the precision and recall. Figure 4 shows precision and recall plots for the different benchmarks on the ICDAR 2003 dataset

As a standard way of summarizing results, we also report the highest F-scores over the PR curves and com pare with [18] in Table 2. Our system achieves higher F-scores in every case. Moreover, the margin of improvement is much higher on the harder benchmarks (0.16 for I-FULL and 0.08 for SVT), suggesting that our system is robust in more general settings.

5 Conclusion

In this paper, we have considered a novel approach for end-to-end text recognition. By leveraging large, multi-layer CNNs, we train powerful and robust text detection and recognition modules. Because of this increase in representational power, we are able to use simple non-maximal suppression and beam search techniques to construct a complete system. This represents a departure from previous systems which have generally relied on intricate graphical models or elaborately hand-engineered systems. As evidence of the power of this approach, we have demonstrated state-of-the art results in character recognition as well as lexicon driven cropped word recognition and end-to-end recognition. Even more, we can easily extend our model to the general-purpose setting by leveraging conventional open-source spell checkers and in doing so, achieve performance comparable to state-of-the-art.

References

[1] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. High performance neural networks for visual object classification. Technical Report IDSIA-01-11, Dalle Molle Institute for Artificial Intelligence, 2011

[2] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In ICDAR, 2011

[3] A. Coates, H. Lee, and A. Y. Ng. An analysis of singlelayer networks in unsupervised feature learning. In AISTATS, 2011.

[4] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In VISAPP, 2009.

[5] B. Epshtein, E. Oyek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In CVPR, 2010. [6] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. Neural networks, 13(4-5):411–430, 2000

[6] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. Neural networks, 13(4-5):411–430, 2000.

[7] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In CVPR, 2011.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1:541–551, 1989.

[9] C.-L. Liu, M. Koga, and H. Fujisawa. Lexicon-driven segmentation and recognition of handwritten character strings for japanese address reading. IEEE Trans. Pat tern Anal. Mach. Intell., 24(11):1425–1437, Nov. 2002.

[10] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. ICDAR, 2003. [11] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In CVPR, 2012. [12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

[13] A. Neubeck and L. Gool. Efficient non-maximum suppression. In ICPR, 2006.

[14] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In ACCV, 2010.

[15] S. J. Russell, P. Norvig, J. F. Candy, J. M. Malik, and D. D. Edwards. Artificial intelligence: a modern approach. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[16] Z. Saidane and C. Garcia. Automatic scene text recognition using a convolutional neural network. In Workshop on Camera-Based Document Analysis and Recognition, 2007.

[17] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In NIPS, pages 1185–1192, 2004.

[18] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In ICCV, 2011. [19] J. J. Weinman, E. Learned-Miller, and A. R. Hanson. A discriminative semi-markov model for robust scene text recognition. In ICPR, Dec. 2008