<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *ISVC 2015, December 14–16, Las Vegas, NV*.

Citation for the original published paper:

Wilkinson, T., Brun, A. (2015)

A novel word segmentation method based on object detection and deep learning.

In: *Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part I* (pp. 231-240). Springer

Lecture Notes in Computer Science

http://dx.doi.org/10.1007/978-3-319-27857-5_21

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-272181>

A Novel Word Segmentation Method Based on Object Detection and Deep Learning

Tomas Wilkinson and Anders Brun

Department of Information Technology
Uppsala University
{tomas.wilkinson, anders.brun}@it.uu.se

Abstract. The segmentation of individual words is a crucial step in several data mining methods for historical handwritten documents. Examples of applications include visual searching for query words (word spotting) and character-by-character text recognition. In this paper, we present a novel method for word segmentation that is adapted from recent advances in computer vision, deep learning and generic object detection. Our method has unique capabilities and it has found practical use in our current research project. It can easily be trained for different kinds of historical documents, uses full gray scale information, does not require binarization as pre-processing or prior segmentation of individual text lines. We evaluate its performance using established error metrics, previously used in competitions for word segmentation, and demonstrate its usefulness for a 15th century handwritten document.

1 Introduction

Segmentation of individual words in documents, in particular historical handwritten documents, is a challenging task that is often crucial for further processing and data mining. Applications include for instance word spotting (image retrieval), clustering, and full text recognition where the document images are converted to a string of characters. Unconstrained handwriting can vary significantly, over time and between different writers. Challenges include words touching each other, both vertically and horizontally; words with no apparent space between them; Non-uniform letter sizes; words with excessive amounts of skewing or shearing. This is particularly challenging in historical documents, where damages and unconventional writing habits add to the variability. For these reasons, we have based our approach in this paper on learning. Given a relatively small amount of training data, we can adopt and optimize our method to the material at hand.

1.1 Prior Work

The majority of the approaches for word segmentation operate on the text line level [1–4]. That is, the text line segmentation is a precursor to word

segmentation. Most methods also require binary images [1–3], though there are exceptions [4]. The proposed method in this paper has neither of these constraints and instead works on images of entire document pages in gray scale. In fact, our approach is more similar to "text spotting in the wild" approaches like [5] that use a two-step paradigm to detect words based on proposal sampling and subsequent classification. However, text spotting in the wild focus on different kinds of data, detecting and recognizing text in natural images, where the difficulty lies in finding text in complex scenes rather than separating several lines of text into distinct words.

2 Background

To understand the method that we propose, we here briefly describe the concepts of convolutional neural networks and the bounding box overlap metric.

2.1 Convolutional Neural Networks

Because the method we present consists of two steps, where the second step is a classification of image region proposals (word candidates), we have a need to classify images. Recent advances in computer vision and object detection has shown that Convolutional Neural Networks (CNN) is a powerful tool for this. In particular, methods using CNNs have won several competitions lately and is widely considered to be current state of the art [6–8].

A (CNN) is a special kind of network specifically designed for image data. A typical CNN is a multi-layered model consisting of an input layer, into which data is fed, several hidden layers, and an output layer. What characterizes a CNN is that one or more of the hidden layers are so-called convolutional layers. A convolutional layer consists of a filter bank, where each filter is applied to the input using convolution, generating a set of output feature maps. A non-linear function is then applied to produce the output. When training a CNN, the weights being learned are the filter coefficients. A convolutional layer is typically followed by a max-pooling layer. This layer subsamples an image using the max operation, that is by keeping the max value over a small region (e.g., 2×2 pixels) for each region position in an image. The hidden layers of a CNN usually has several convolutional layers, each of which followed by a max-pooling layer, then one to two fully connected layers. CNNs are easier to train than regular neural networks due to convolutional layers having significantly fewer parameters, which is a partial explanation behind their recent success in a variety of different tasks. For more, see for example [9].

2.2 Bounding Box Overlap

When we evaluate the result of our method, there is a need to measure how close the resulting segmentation resembles ground truth, i.e. bounding boxes of words in a document that have been annotated by an expert. For this we need a metric

to compare bounding boxes. Intersection over union (IoU) [10, 5] is a common way to calculate how much two rectangular boxes overlap as a percentage. It is defined as the area of intersection between two boxes divided by the area of their union. It is commonly used in the detection of objects or text in images to determine how precise the algorithm in question can localize the objects you are interested in. This is typically calculated using only the box representations, e.g., a point and a width and height. Though if you have binary data, it is possible to calculate the IoU using only the foreground information. That is, you count how many foreground pixels are present in the intersection of the two boxes divided by how many foreground pixels there are in the union of the two boxes.

3 Method

The method we present in this paper is adapted from the R-CNN framework [10], Regions with Convolutional Neural Network Features. It is a method used in computer vision for generic object detection. The R-CNN framework has proven to work well in object detection, where it achieved state-of-the-art results when it was introduced and it has continually been improved upon since then. For example, removing a lot of unnecessary computation by making use of the Spatial Pyramid Pooling layer is introduced in [11]. Word segmentation can be seen as a special case of object detection that will work better if we adapt the framework for some specific properties of regions containing handwritten words in historical documents.

The method consists of two parts or modules. The first module produces bounding box region proposals. It is completely separate from the second part, which classifies the proposals into two classes, whether or not a bounding box is centered on a word. The algorithm used in [7] failed to generate word-like region proposals on the document images from the datasets used in this paper. Consequently, we have re-designed a region proposal algorithm specifically for word segmentation in historical document images.

3.1 Generating Region Proposals for Handwritten Words

The algorithm to generate tentative proposals for handwritten words takes its point of departure in the uncertainty of gray levels and the exact scale and shape of the morphology operators necessary to separate words.

1. Given a gray scale image, generate a unique set of l binary images, with the foreground being white, by globally thresholding the image at different multiples of the image's mean intensity. Call this set of thresholds L .
2. Generate a set of k rectangular structuring elements by forming all combinations of heights and widths (in pixels) from two sets H and W . Typically H and W are ranges of natural numbers, sampled at even intervals.

Fig. 1: The CNN architecture used for classifying whether or not a bounding box is centered on a word.

3. For all the l binary images resulting from step 1, apply each of the k structuring elements from step 2 using a morphological closing operation. This results in a new set of $k \cdot l$ binary images.
4. For each of the $k \cdot l$ binary images, label its foreground objects and extract a bounding box for each labeled object.
5. Remove all duplicate boxes.

The choice of L , H and W is discussed in the experiments section and given for each dataset that we have tested our algorithm on.

3.2 Classification

The second module consists of a deep convolutional neural network (CNN) trained to classify whether bounding boxes are centered on words or not. The network architecture is based on the one used in [5] to recognize text, though it has been simplified slightly. The network consists of four convolutional layers followed by two fully connected layers, where the last layer is an SVM output layer [12]. See Figure 1 for an image of the architecture used. Zero-padding is used to keep the sizes intact when performing convolutions. After the first three convolutional layers, subsampling by max pooling is done in a 2×2 region, which corresponds to downsampling by a factor of 2 in each dimension. A dropout layer [8] was used after the first fully connected layer with dropout ratio of 0.5. The network is trained using the L1-hinge loss. Rectified linear units were used for all activation functions.

The region proposals come in a large variety of sizes and since the CNN requires a predefined fixed size input, the region proposals are resized to 40×120 before being inputted into the network. Before resizing, we include a surrounding context of 16 pixels vertically and 48 pixels horizontally for each bounding box.

We create training data for the network by labeling each of the region proposals extracted from the images belonging to the training set. For a given region proposal, we calculate the IoU overlap between the it and all ground truth boxes. If the one of the overlaps exceeds a predefined threshold t_u , it is assigned a positive label, otherwise it is assigned a negative label. The training details are described in the experiments section.

For testing, all of the region proposals are scored by the CNN, where the score is the probability of the box being centered on a word. Then non-max

philosophy, he is an enigmatic figure known only through the classical accounts of his students. Plato's dialogues are the most comprehensive accounts of Socrates to survive from antiquity. Forming an accurate picture of the historical Socrates and his philosophical viewpoints is problematic at best. This issue is known as the Socratic problem. The knowledge of the man, his life, and his philosophy is based on writings by the students and contemporaries. Foremost among them is Plato, however, works by Xenophanes, Aristotle, and Aristophanes also provide important insight. The difficulty of finding

(a) English

অমর্যাদা প্রসঙ্গে বলে মুক্ত ভিক্ষুণী

কটমার্জ, ১৩ নোবেম্বর : হিন্দুদের একটি ভবিষ্যৎ বুদ্ধদের ও বৌদ্ধদের
অমর্যাদা করে হয়েছে বলে প্রত্যাশা জ্ঞানালীন নিগালয় বৌদ্ধ ভিক্ষুণী
‘হিন্দু বুদ্ধ’ নামের ওই গিনমা’ সেখানে দেখানো হয়েছে একটি
বুদ্ধমূর্তি ‘মায়’ বলে আছে। ‘ভবিষ্যৎ পরিচালক’ বুদ্ধদের জন্ম
মূর্তির কাছে ‘বাসনাময়’ শব্দের বৌদ্ধ ভিক্ষুণী ‘আবিলিষ্ট’ এই ভবিষ্যৎ
প্রত্যাশার কীর্তি দর্শনে বিভ্রান্ত জ্ঞানালীন। তাঁদের মতে এভাবে বুদ্ধ
বুদ্ধদেরকে নয়, বৌদ্ধ ধর্ম ও এই শব্দের অর্থাদিকেই বুদ্ধদেব হিসেবে
দেখা সঠিক। অমর্যাদা’ অমর্যাদা’ অমর্যাদা’ অমর্যাদা’ অমর্যাদা’
‘মিষ্ট’ বিভ্রান্ত দেখানো বলে ‘সিদ্ধান্ত’ প্রমাণ করে বসে বসে।

‘সিদ্ধান্ত’ প্রমাণ।

(b) Bangla

Fig. 2: Two sample images from the ICDAR2013 dataset in two languages.

suppression is applied as a post-processing step in which boxes that have an overlap with a higher scoring box over a threshold value is removed. Finally, boxes with a score lower than threshold are discarded (typically noise from the binarization and small parts of characters).

4 Data

Here we describe the datasets we have used to evaluate the proposed method.

4.1 ICDAR2013

The first dataset is the ICDAR 2013 handwriting segmentation contest ¹ [13], the latest installment in a well known series of handwriting segmentation contests. Along with the contest, a dataset (hereafter referred to as ICDAR2013) was released for both text line and word segmentation. It consists of 200 training pages and 150 testing pages and contains several writers, writing in English, Greek and Bangla languages. We use 10%, or 20 pages, of the training data as a validation set. The images of the dataset are binary and does not contain any non text elements. Two sample images can be seen in Figure 2

4.2 C61

The second dataset consist of five spreads or ten pages from the C61 collection, written in the 15th century. It is written in Swedish and although the entire C61 collection is written by either four or five writers, the subset we use is written by a single writer. The text is tightly written (as is often the case with historical manuscripts, as parchment was expensive) with little space between text lines and words, making word segmentation for this data quite

¹ <http://users.iit.demokritos.gr/~nstam/ICDAR2013HandSegmCont/>

challenging. A sample of an image from C61 can be seen in Figure 3. Contrary to the ICDAR2013 dataset, C61 is available in colour. The annotations for this dataset were manually done for the five images. We split the data into a training set of three images, a validation set of one image and a testing image to show the results.

Fig. 3: A part of a page from the C61 dataset.

5 Evaluation

In this section we describe how we evaluate the proposed approach.

5.1 The choice of overlap threshold

The choice of overlap threshold is critical when evaluating a word segmentation algorithm. In this paper, the data can be divided into categories; Overlap calculated using all the pixels in the bounding boxes and overlap calculated using only the foreground pixels. The two categories need different thresholds to achieve a similar overlap requirement in practice. This is evident in Figure 4, where the two boxes have an overlap of 96% calculated using only foreground pixels and 66% using all pixels. The problem is particularly evident with words that contain ascenders and descenders (The parts of letters that go below and above the baseline, e.g., g and h).

5.2 Competition Metrics

To be comparable to the previous results we adopt the evaluation metrics used in the ICDAR 2013 handwriting segmentation competition [13]. According to

Fig. 4: IoU overlap using all the pixels between 4a and 4b is 66% while using only foreground pixels is 96%.

the competition rules, a word is correctly segmented if the IoU of the foreground pixels of the ground truth and a proposal is higher than 90%. Let N be the number of ground truth elements, M be the number of resulting proposals from a segmentation algorithm and $o2o$ be the number of one-to-one matches between the results and the ground truth. Then the detection rate (dr) and recognition accuracy (ra) are defined as

$$dr = \frac{o2o}{N}, ra = \frac{o2o}{M} \quad (1)$$

and the final metric is the harmonic mean between dr and ra .

$$fm = \frac{2 \cdot dr \cdot ra}{dr + ra} \quad (2)$$

The three metrics vary between 0-100%, where 100% is a perfect score.

6 Experiments

In this section, we experimentally evaluate our word segmentation approach using two datasets. The experiments were performed using the Cuda/C++ library Caffe [14] and Python. The region proposer is implemented in pure Python and processing an image takes about 15 to 30 minutes, depending on the sizes of L, H, W and the size of the image

6.1 ICDAR2013

Since the images in this dataset are already binary, the first step in 3.1 is simplified. Let the triple $(start, stop, step)$ represent a range of numbers starting at $start$, ending at $stop$ with step length $step$. We found that $H = (1, 28, 3)$ and $W = (2, 42, 4)$, giving $k = 110$ gives a good trade-off between running time and recall. Using these settings we get an average test set recall of 93.28%. The upper IoU threshold, t_u , of 90% is already defined by the competition rules. A non-max suppression overlap of 0.08 and score threshold of 0.3 is determined by doing a coarse grid search on a validation set.

The weights of the network are initialized from a zero mean Gaussian distribution with a standard deviation of 0.01. Training was done using an initial learning rate of 0.001 and multiplying it after 22000 iterations with 0.1. Training is stopped at 42000 iterations. A momentum of 0.9 was used as well as an l_2 weight decay of 0.0005. We use three times as many negative examples as positive in the training set.

As can be seen in Table 1, the proposed method achieves the second to highest score on the ICDAR2013 dataset, coming in below the state-of-the-art by 0.7%.

Table 1: The results of our method compared to other methods on the ICDAR2013 dataset

Method	dr (%)	ra (%)	fm (%)
GOLESTAN-a from [13]	89.66	90.44	90.04
method from [1]	90.50	91.55	91.03
proposed method	87.09	93.82	90.33

6.2 C61

Since this dataset is in color, we first convert it to gray scale by averaging the R, G and B color channels. Then five binary images were generated from the gray-scale image with $L = 0.6, 0.7, \dots, 1.0$. We found that $H = (3, 28, 5)$ and $W = (3, 28, 5)$, giving $k = 36$ structuring element sizes worked sufficiently well. These parameters give a recall of 89.6% on the test set. Since we work in gray scale with this dataset, IoU overlap is counted using all pixels, requiring a lower u_t . We found $u_t = 70\%$ reasonable. As above, a non-max suppression overlap of 10% and a score threshold of 0.07 is determined by doing a coarse grid search on the validation set.

We initialize weights of the network using the weights from the network trained on the ICDAR2013 dataset, this is called pre-training and is a common practice in deep learning. We use an initial learning rate of 0.001 and multiply it every 3000 iterations with 0.1 until training is stopped at 9000 iterations. Note that though the images are binarized when extracting region proposals, the image classification network is trained using the gray scale images.

Once again using the metrics from above, we get $dr = 77.85\%$, $ra = 82.41\%$, and $fm = 80.07\%$. The results are visualized in Figure 5.

7 Discussion

The choice of structuring elements to use is a trade-off between recall and running time. For a document image with a width or height between words of at most

Fig. 5: Result from the proposed methods for the test set with its corresponding ground truth from the C61 dataset.

100, setting both H and W to $(1, 50, 1)$ would yield highest recall. This would give $k = 50 \cdot 50 = 2500$, which is probably unnecessarily high. In practice, you can significantly reduce k with a marginal loss in recall by increasing *step* and reducing *stop*. We have found that we consistently achieve a recall of over 90% using a k in the range of 30-100. Another important choice is the selection of L . We opted for a simple approach, inspired by [15]. Exchanging this method to state-of-the-art binarization approaches would probably increase performance, but at the cost of making the algorithm more complex.

8 Conclusions

In this paper, we have introduced a new method adopted from R-CNN, a generic object detection framework, to the new task of handwritten word segmentation in document images. The proposed method achieves results competitive to the state-of-the-art methods, as evaluated on the ICDAR2013 dataset, while at the same time removing the requirement of pre-segmented text lines and binary images. A possible downside of the method is a reliance on training data and, for the current implementation, a GPU for fast training. However, we also show that the amount of training data needed can be quite low, thanks to pre-training on other datasets. Only three spreads (six pages), were needed as training data to get a reasonable result for a challenging image from this particular 15th century document, consisting of over 1100 pages in total.

Acknowledgment

This project is a part of q2b, From quill to bytes, a framework program sponsored by the Swedish Research Council (Dnr 2012-5743) and Uppsala university. The

work is done in part as a collaboration with the Swedish Museum of Natural History (Naturhistoriska riksmuseet)

References

1. Ryu, J., Koo, H.I., Cho, N.I.: Word segmentation method for handwritten documents based on structured learning. *Signal Processing Letters, IEEE* **22** (2015) 1161–1165
2. Stafylakis, T., Papavassiliou, V., Katsouros, V., Carayannis, G.: Robust text-line and word segmentation for handwritten documents images. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* (2008) 3393–3396
3. Varga, T., Bunke, H.: Tree structure for word extraction from handwritten text lines. In: *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on.* (2005) 352–356 Vol. 1
4. Manmatha, R., Rothfeder, J.L.: A scale space approach for automatically segmenting words from historical handwritten documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27** (2005) 1212–1225
5. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* (2014) 1–20
6. Gidaris, S., Komodakis, N.: Object detection via a multi-region & semantic segmentation-aware cnn model. *arXiv preprint arXiv:1505.01749* (2015)
7. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. *arXiv preprint arXiv:1505.03540* (2015)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems.* (2012) 1097–1105
9. Bengio, Y., Goodfellow, I.J., Courville, A.: *Deep learning.* Book in preparation for MIT Press (2015)
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE* (2014) 580–587
11. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *Computer Vision–ECCV 2014.* Springer (2014) 346–361
12. Tang, Y.: Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239* (2013)
13. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: Icdar 2013 handwriting segmentation contest. In: *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, IEEE* (2013) 1402–1406
14. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: *Proceedings of the ACM International Conference on Multimedia, ACM* (2014) 675–678
15. Kovalchuk, A., Wolf, L., Dershowitz, N.: A simple and fast word spotting method. In: *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on.* (2014) 3–8