To Appear in CVPR 2009

Robust Unsupervised Segmentation of Degraded Document Images with Topic Models

Timothy J. Burns and Jason J. Corso
SUNY at Buffalo
Computer Science and Engineering
201 Bell Hall, Buffalo, NY 14260
tjburns@cse.buffalo.edu, jcorso@cse.buffalo.edu

Abstract

Segmentation of document images remains a challenging vision problem. Although document images have a structured layout, capturing enough of it for segmentation can be difficult. Most current methods combine text extraction and heuristics for segmentation, but text extraction is prone to failure and measuring accuracy remains a difficult challenge. Furthermore, when presented with significant degradation many common heuristic methods fall apart. In this paper, we propose a Bayesian generative model for document images which seeks to overcome some of these drawbacks. Our model automatically discovers different regions present in a document image in a completely unsupervised fashion. We attempt no text extraction, but rather use discrete patch-based codebook learning to make our probabilistic representation feasible. Each latent region topic is a distribution over these patch indices. We capture rough document layout with an MRF Potts model. We take an analysis-by-synthesis approach to examine the model, and provide quantitative segmentation results on a manually-labeled document image data set. We illustrate our model's robustness by providing results on a highly degraded version of our test set.

1. Introduction

We examine the problem of segmenting document images into text, whitespace, images, and figures through unsupervised learning methods. Many methods currently exist for performing text extraction and segmentation for OCR [7, 10]. The main drawback of these methods is that they largely rely on heuristics to separate the different regions from a given document image. Common heuristic methods are generally not descriptive enough to capture the significant variance often present between different document images. Furthermore, most document segmentation methods require high quality scans of the pages. Hence, the presence of significant noise or degradation found in low-resolution video or cell phone images, causes many common methods to fall apart. Unsupervised methods are desirable in this context because they do not involve a great deal of domain specific training knowledge, and ground truth is not necessary. For example, the same methods we propose could work on technical articles (as we discuss here), newspapers, magazines, etc.

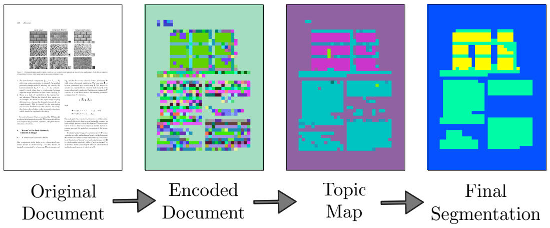

Figure 1. We propose an unsupervised model in which we learn a set of latent topics over image regions to perform document segmentation. We perform three steps as illustrated in the pipeline above: codebook learning, latent topic inference, and MRF smoothing to better estimate the layout. When all this is done we have a final segmentation of a given document image.

We propose an unsupervised model that, rather than seeking to perform text extraction, breaks each document down into reasonably sized image patches to perform discrete patch-based image coding (refer to figure 1 to see these steps graphically). Using the codebook obtained from this process we can represent each image as a bag of these codewords and then learn latent topic distributions over these codewords for each document in the corpus. We expect in this latent discovery of topics that the model distributes them in a way that is semantically meaningful. In other words, each topic will correspond to regions of document images that are visually similar. Finally, we model the layout of the documents using a Potts MRF energy model with high-order energy terms to encourage the kind of layout we expect. We note that our model is unsupervised in every aspect except the final translation from topic-mapped images to final label images which requires a small amount of manually labelled images to reference. After discussing the model we then take an analysis-by-synthesis approach judge its ability to capture layout and demonstrate its main application to segmentation. We conclude by testing robustness by applying the same model to highly degraded document images. Due to its unsupervised nature, comparing to existing heuristic methods is difficult, but we provide quantitative results by testing our model on a test set of 100 manually-labeled document images, and show robustness in the presence of severe degradation.

2. Background

One early method for document segmentation into more than two region types is [8]. The authors use different thresholding techniques and histogram features as input to a neural network in order to classify the different regions. Similarly, [12] use modified heuristic edge following methods to efficiently segment document images for explicit OCR. This second approach is not able to identify more regions than just text and whitespace, as we have proposed.

Some approaches to document segmentation focus on specific features of the image or identifying word blocks in order to perform classification and obtain segmentation results. One such approach [13] uses matched wavelet filters and Fisher classifiers to estimate a three class image labeling problem: text, background, and picture. Another approach [21] uses a set of filter and histogram features in conjunction with Fisher classifiers to identify and distinguish text regions for handwritten and machine printed text. Both of these methods then use an MRF post-processing step to smooth out as many misclassifications as possible. Although these approaches are learning-based, they still rely on large amounts of pre-processing and a fully annotated training set to perform the supervised learning step.

Another way to view the task of document segmentation is as a texture modeling problem. In this context we seek a descriptive/generative model that can learn patterns of textures present in the document image. In one of the earlier works to examine texture in document images [9, 19], the authors examine the texture of different document images through the use of multichannel filtering based on Gabor filters in order to identify textual regions. Although many recent and sophisticated models exist to model textures in images, such as [22], they fail to represent the kind of inhomogeneous texture which we find present across the different regions of a document image and remain too computationally expensive to be of any practical value. Some approaches like [5] seek to get around this by using nonparametric sampling in the synthesis process based on image windows rather than local pixel relations, but the methods are not yet sufficiently advanced.

Topic Models have been left largely unused in document image analysis—to the best of our knowledge, we are the first paper to propose using unsupervised topic models in the document imaging domain. In their original paper, Blei et al. [3] discuss a few applications of their model (which they call LDA) to various discrete corpora: document modeling (textual), document classification, and collaborative filtering. Given the current trend to use some form of codebook over image data sets, it is surprising that very few authors have applied this model in the imaging domain. One exception is Fei-Fei et al. [6], who transfer the model into the image domain and add one more hidden variable to perform scene categorization. Another application can be found in [15] in which the authors improve upon the topic model's inability to capture spatial localization. This lack of localization is the biggest problem in applying these models successfully to images which, unlike text corpora, contain considerable spatial information. Their early success of applying topic models in imaging has laid the ground-work for our work. They provide a good example of how the model could work. However, their approach is not directly applicable since we are interested in unsupervised segmentation.

3. The Model

We now describe each part of our model's "pipeline" as given in figures 1 and 2. Starting in section 3.1 we present our codebook learning approach. In section 3.2 we illustrate how topic models are applied to our problem. Finally, in section 3.3 we impose some layout through a high-order Potts MRF model.

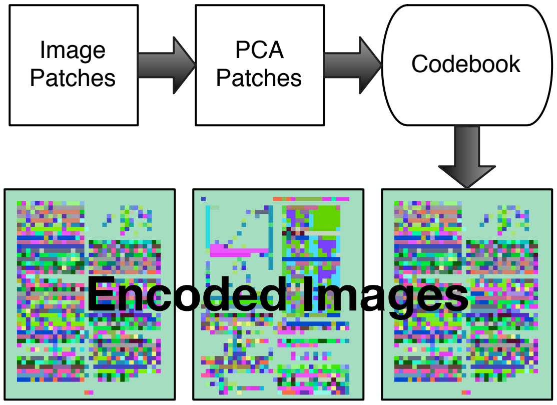

Figure 2. Codebook Learning Process: We take raw image patches, perform PCA over a large set, and run k-Means to get a set of k codewords which can encode our images.

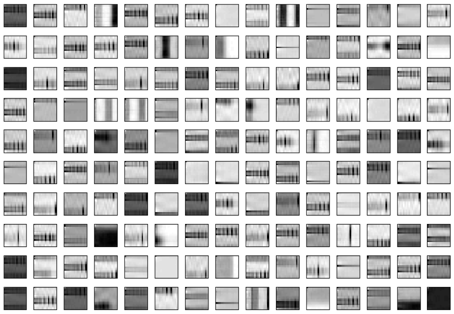

Figure 3. A codebook consisting of 150 codewords (centroids produced by the K-Means algorithm). The gray effect is due to the averaging effect the PCA process has on the patches when subtracting the empirical mean.

3.1. Codebook Learning

Codebook learning is closely related to vector quantization [14] and is used to obtain a lower dimensional embedding of an image. In the imaging domain typically these kinds of codebooks consist of image descriptors or patches. A number of papers propose using codebooks for features in a larger model in areas like object recognition and categorization [4, 11, 17, 20]. An approach based on texture given in [16] makes use of filter-based "textons" to transform an image into a texture map. In [6], the authors learn a codebook using patches on a sliding grid in random scales. Almost all of these methods use some clustering algorithm like K-Means in order to learn the basis from which to form their codebook.

We adopt such a learning scheme for our codebook. Since we are interested in modeling document images which are generally very linearly organized, we choose our features to be small 16x16 image patches arranged on a regular, rectangular grid over the entire document image. Since all of our document images are roughly 800x600 pixels, this patch size is large enough to capture the structure of individual words and other features of a document while not being so large that we lose overall descriptiveness.[1] These image patches p_i are transformed into feature vectors and are combined into a single data set $X = \{p_1, p_2, ..., p_n\}$. The principle components of this data set are then determined via a PCA transformation $Y = PCA\{X\}$ which also then reduces the dimension of the patches. Once we have this, we apply K-Means for unsupervised clustering. K-Means produces a set of vectors which represent the k centroids of the model. This set of centroids $C = \{c_1, ..., c_k\}$ represents our codebook where each i is a "codeword" which indexes some c_i.

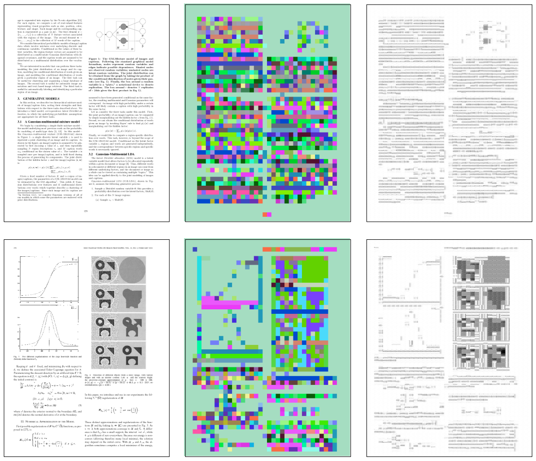

Figure 4. Encoded representations of two different document images (one per row). The first column are the images themselves, the second are the encoded images (with colors randomly assigned to each codeword), and the third are the reconstructed images consisting of the representative patches of each codeword on the image lattice.

After learning a codebook we can then define a document image over a lattice of codewords where each codeword represents an individual image patch. For an average document size of roughly 800x600 pixels this represents roughly a reduction of about 300 times. In addition, we expect to have captured much of the spatial interactions among individual pixels in a similar manner even to that obtained by filter banks. When representing a given patch p_i as a codeword from our model we choose the closest codeword in a Euclidean sense:

$$w_{p_i} = \underset{k}{\operatorname{argmin}} \left\{ \sqrt{(p_i - c_k)^2} \right\} \tag{1}$$

From figure 3 we see that the codeword patches resemble the kinds of responses one might observe from filter application to text. Figure 4 illustrates two document images and their representative encodings under the codebook in figure 3. Much of the perceptual qualities of these images are preserved under the embedding.

3.2. Topic Models for Document Images

Latent Dirichlet Allocation [3] (LDA) or *Topic Models*, is a method used for modeling collections of discrete data such as text corpora. We observe that our encoded document images can essentially be represented as discrete collections of codewords. We hence adopt the notation from the LDA model in what follows.

The following are the terms we use in our document image topic model:

[1] In the degraded, low-resolution experiments we reduce the patch-sizes comparably (see section 5)

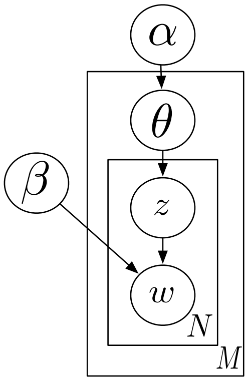

Figure 5. Graphical model for LDA (Topic model). Outer plate represents M documents, while the inner represents repeated N topic/word choice per document.

- A *codeword* $w_{p_i} \in \{1, ..., k\}$ is our fundamental unit of discrete data.
- A *document* (image) is a sequence of N codewords denoted by $\mathbf{w} = \{w_{p_1}, w_{p_2}, ..., w_{p_N}\}$.
- A *corpus* is a collection of M documents (images) $\mathbf{D} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$.

Figure 5 illustrates the following generative process graphically:

1. Choose $\theta \sim Dir(\alpha)$
2. For each of the N codewords w_{p_i}:

   a) Choose a "topic" $z_i \sim Multinomial(\theta)$

   b) Choose a codeword $w_{p_i} \sim p(w_{p_i}|z_i, \beta)$

A t-dimensional Dirichlet random variable has the following probability density over the $(t-1)$ simplex [3]:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{t}\alpha_i)}{\prod_{i=1}^{t}\Gamma(\alpha_i)}\theta_1^{\alpha_1 - 1}\cdots\theta_t^{\alpha_t - 1} \tag{2}$$

The Dirichlet distribution is conjugate to the Multinomial distribution.

Given the hyper-parameters α and β, we can express the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N codewords $\mathbf{w}$ which they generate as [3]:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha)\prod_{n=1}^{N}p(z_n|\theta)p(w_n|z_n, \beta) \tag{3}$$

The probability of a corpus is then derived to be:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn}, \beta)\right) d\theta_d \tag{4}$$

Inference is made difficult due to the coupling of certain hidden variables so variational inference is employed. Full derivations are available in the original paper [3].

In the context of our model, the parameters β and α are the mixing parameters of the Dirichlet distribution. θ is the distribution over image regions like text and whitespace for each document image, and z is the distribution of image codewords within each distinct image region (topic).

We expect that by applying this model it will be able to cluster data based on these latent "topics." In this case topics are meant to represent the distribution of codewords typically found in distinct document image regions. By limiting the number of topics to those typically found such as whitespace, text, images, and figures, the topic model will then discover these regions by grouping codewords that have similar appearance.

3.3. Incorporating Layout through MRF

We introduce a Potts-like MRF model to constrain some of the expected layout of the topics (document regions). Although such a local MRF-based model cannot capture the full gamut of global layout, we design a set of high-order potential functions that encourage that expected local structure. In a document image, regions of text, background, or figures are often found in homogenous blocks. Therefore, we want to define a local energy function which seeks to promote these kinds of structures within a sample image.

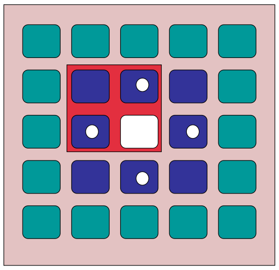

Figure 6. Local neighborhood Potts energy model for topics. White dots are the local first order neighbors (eq. 5). Dark red is the corner penalty (eq. 6) to encourage sharp edges. The large light red area is the second-order neighborhood (eq. 7) penalty to encourage larger regions.

We define the local energy in a Potts-like model with some extra higher-order interaction penalties. The Gibbs energy has the following form:

$$\begin{aligned} H(x) = \quad & \gamma_1 \sum_{s \sim t \in \partial x} \delta(x_s, x_t) && (5) \\ + \quad & \gamma_2 \sum_{s \sim t \in C} \delta(x_s, x_t) && (6) \\ + \quad & \gamma_3 \sum_{s \sim t \in \partial^2 x} \delta(x_s, x_t) && (7) \\ - \quad & \gamma_4 \quad \log\left(P(x_s|s)\right) && (8) \end{aligned}$$

Note that in the above equations ∂x are the first-order neighborhood, C represents each surrounding corner, and $\partial^2 x$ are the second order neighbors. Each term contains a δ function which activates a penalty if the neighbor or neighbors in question are not equal to x_s. This is the same as the δ function in a standard Potts model. Equation 4 is a standard first-order penalty just as in a Potts model, while equations 5 and 6 are designed to encourage sharp corners (to make regions more square as they would be in typical machine printed document) and encourage larger regions. Obviously this model would have to be adjusted for handwritten text and assumes that ground truth is viewed in this regular fashion. We feel, however, that this assumption is reasonable given the nature of document images. The final term is simply the likelihood of finding a particular topic at a specific lattice location s.

4. Dataset and Inference

Essentially, what we have proposed is a three step modeling framework through which we will obtain our results:

1. Learn the codebook;
2. Estimate the Topic Model;
3. Apply the MRF layout Model

Our data set consists of 1780 total document images from which we label 100 to use as a test set for quantitative segmentation results (keep in mind that since our model is unsupervised we did not need to do this for training). We cut each of our 1680 total training document images up into 16x16 patches and select a random subset of all those patches. We end up with approximately 175,000 image patches on which we perform PCA and input to the K-Means algorithm. The output of this algorithm is a set of codewords which we will use to encode each document image. Since we are interested in segmenting whitespace, text, figures, and images, we use a four topic model. After learning the topic model we infer the best topic distributions for the 100 test document images. Once we have the learned topic model we compute a simple maximum likelihood solution $P(t_i|w_j, \mathbf{w})$ representing the probability of topic t_i being responsible for observing codeword w_j in document $\mathbf{w}$. This gives us a preliminary segmentation of each image in the test set. In order to improve upon this and impose the MRF layout proposed, we then run each image through simulated annealing using a cooling schedule of $T = T_0 \frac{T_N}{T_0}^{i/N}$. Each instance of the Gibbs sampler is run for five iterations which proves to converge in a short amount of time. Simulated annealing is practical here due to the small lattice size after we encode the images. Once this converges we compute segmentation results by expanding each location of the topic lattice to a 16x16 grid and calculating accuracy at the pixel level. To analyze the model's robustness on grossly degraded images, we down sample our test images to 80x60 resolution and perform the same process. Since some of the documents in our data set are already of lower quality already, this degradation is amplified. We do this since these low-quality, low-resolution images are what one would find by performing recognition or segmentation of documents from video in a larger office scene, for example.

5. Results

We present our results on the original and degraded document images by i) an analysis-by-synthesis investigation and ii) a quantitative comparison to manual annotation.

5.1. Synthesis

In figure 7 we show the results of our synthesis procedure with different numbers of topics. Values for γ_1, γ_2, γ_3, and γ_4 were set empirically by manually determining what settings produced the best quantifiable segmentation results on the training set. The balance between the gamma values is a crucial aspect of the MRF sampling process. If the emphasis on local evidence via γ_4 is too large then the border tends to invade too far into the center of each document. If layout and smoothness is preferred via γ_1, γ_2, and γ_3 then synthesized documents tend to look more random. With a good balance we find regions like that shown in column 1 of figure 7. These regions are very close to the kind we expect.

Between different topic models it is clear that the effect of the MRF model is different. The better looking model in terms of the layout produced by the MRF seems to be the four topic model, however, the final sampled image appears more random. The five topic model seems to show a better balance in terms of both layout and final results.

In terms of the learning for the topic model itself, the initial parameters seem to have little effect of the final results of the synthesis procedure. We learned all the models represented here with an initial setting for α of 0.1.

The layout that is sampled here is promising, but clearly none of the sampled documents in column three contain enough structure. This is expected, however, since our main goal here is segmentation and not layout analysis. The vari-

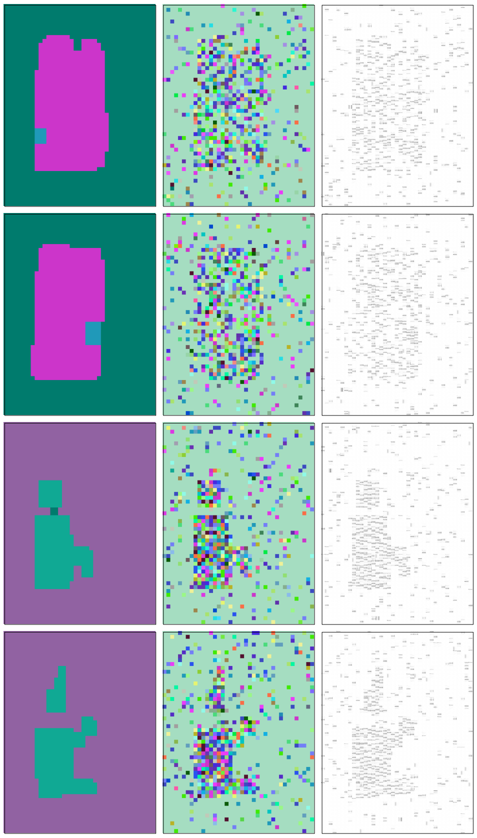

Figure 7. Two synthesized documents from a 4 topic model (top two rows) and two from a 5 topic model (bottom two rows). Column one is the topic map sampled from the MRF model, column two are the sampled images from these topics shown in encoded form, and the third column shows the representative documents

ability in layout produced by slight modification of the MRF parameters indicates that the model is too sensitive to small local interactions to accurately model the global layout.

5.2. Segmentation

We now turn our attention to the main focus of our paper. We use the maximum likelihood distributions $p(t_i|w_j)$ inferred from the topic model as evidence. Since the algorithm is unsupervised we then manually determine the mapping from topics to ground truth and compute segmentation accuracy on a pixel-by-pixel basis (note that this is the only "supervised" portion of the model). We perform segmentation for codebook sizes ranging from 30 codewords to 150 codewords. Furthermore, for each codebook we compute a different segmentation for both a four and five topic model to see if there is any significant difference. Finally, we demonstrate the robustness of our model on the same set of images degraded by down-sampling to 80x60 resolution.

Figure 8 is a graph of pixel accuracy of the two models as the codebook size increases. The solid and dashed lines represent the segmentation with applied layout via simulated annealing, while the other two represent a simple max-likelihood result for each codebook. All of the models do very well given their completely unsupervised nature, but the MRF layout provides both more consistency across all codebook sizes as well as general performance improvements. This graph also surprisingly seems to indicate that there is no real advantage of one codebook size over another (at least with annealing). This might mean that regardless of any possible redundancy in the codebook, the topic model is not affected at least up to the size we explored. The four topic model also appears to very slightly outperform the five topic model for most codebook sizes, but the difference is not substantial which seems to indicate that using the method we've proposed there is a limit to how much the topic model can discover.

Table 1 provides an example confusion matrix for the entire 100 document test set. The figure and image classes are combined because upon observing the results we noticed that the figure class is rarely distinguished under the topic model. This most likely has to do with an inability of the topic model to distinguish these regions due to their sparse nature since they are comprised primarily of graphs and drawings which contain thin lines and considerable whitespace. Despite these drawbacks the model does well enough with all the other classes to still achieve almost 90% total pixel accuracy.

Figure 9 contains ten examples of automatically segmented document images. At a high-level, our segmentations are accurate as the sums in Table 1 indicate. But, there are a few places where the MRF model has over-smoothed and we can see that the figure class is generally left undiscovered. The seventh document in the figure is most likely the worst since the original image was degraded enough to where it was not obvious if the three image regions were even images or just figures. Examples like this are very few and, in fact, even in the presence of significant degradation we see still good results. If we turn our attention to the second part of figure 9 we see how well our model does on the same set of images degraded to 80x60 resolution. The model performs at 80% total pixel accuracy even at this low

	WS	Text	Fig/Im
WS	65.19	34.74	0.07
Text	10.89	89.10	0.009
Fig/Im	39.76	32.54	27.70

Table 1. Confusion matrix for the four topic model over 70 codewords. The table shows the percentage of total pixels from the test set and how they are classified. The figure and image class are shown together here because the figure class is rarely classified (either correctly or incorrectly).

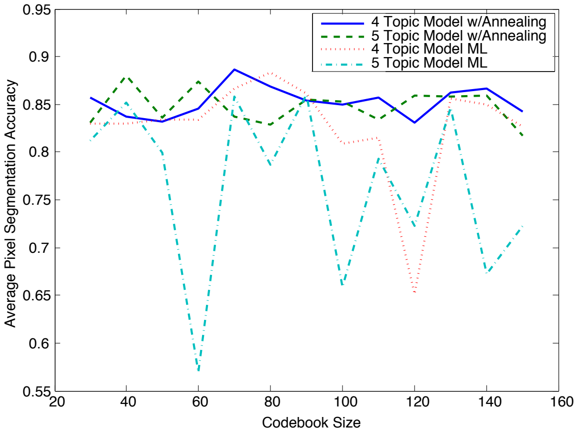

Figure 8. A graph of segmentation accuracy as codebook size increases.

level of resolution which illustrates our model's robustness and flexibility. Finally, we provide an example segmentation on all eight pages of this document (see figure 10) to illustrate how well the model does on data completely outside of the initial training and test set.

Although we do not have direct access to the data sets used, we can roughly make note of our method in relation to existing ones. In [1] most of the methods discussed obtain no more than 60% accuracy in general on the different classes of data under their defined EDM (Entity Detection Metric). We use a more raw approach; it seems promising that with about 90% accuracy we would do at least comparably.

6. Conclusions and Future Work

We have proposed a framework for segmentation of document images and presented results through synthesis and quantitative segmentation on a test set of example document images. Our model shows considerable promise, particularly due to the fact that it is largely unsupervised in its approach and in how well it holds up under significant degradation. Some details, such as the lack of detection of figure regions, demonstrate the need for a more advanced topic model which takes spatial information into account. Possibly providing more context and background information might help to enhance the model as well, although if this was relied upon too much we could lose the advantage of the unsupervised approach. We will consider training the model on specific document types and formats (eg. journal and conference formatting, etc) and possibly breaking it down into page type (eg. front page, reference page, etc) and learn separate models for each.

There are a number of extensions to the topic model itself that could be explored. Correlated topic models [2] might introduce interesting dependencies between topics (perhaps in an effort to learn more structural information), and author-topic models [18] could be explored in an effort to perform document recognition. Although clearly there is much room for extension, the most important contribution of this paper is the demonstration of accurate unsupervised segmentation of normal and degraded document images without heuristics.

References

[1] A. Antonacopoulos, B. Gatos, and D. Bridson. ICDAR2005 page segmentation competition. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 75–79.

[2] D. M. Blei and J. D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing*, volume 18, Cambridge, MA, 2006. MIT Press.

[3] D. M. Blei, A. Y. NG, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, (3), 2003.

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[5] A. A. Effros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, 1999.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[7] J. Fisher, S. Hinds, and D. D'Amato. A rule-based system for document image segmentation. *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, i:567–572 vol.1, Jun 1990.

[8] S. Imade, S. Tatsuda, and T. Wada. Segmentation and classification for mixed text/image documents using neural network. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 930–934, 1993.

[9] A. Jain and S. Bhattacharjee. On texture in document images. In *In Proceedings of Computer Vision and Pattern Recognition*, number 31, pages 677–680, 1992.

[10] A. Jain and B. Yu. Document representation and its application to page decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):294–308, Mar 1998.

[11] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005.

[12] B. Kruatrachue and P. Suthaphan. A fast and efficient method for document segmentation for ocr. In *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology*, volume 1, pages 381–383, 2001.

[13] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi. Text extraction and document image segmentation using matched wavelets and mrf model. *IEEE Transactions on Image Processing*, 16(8), August 2007.

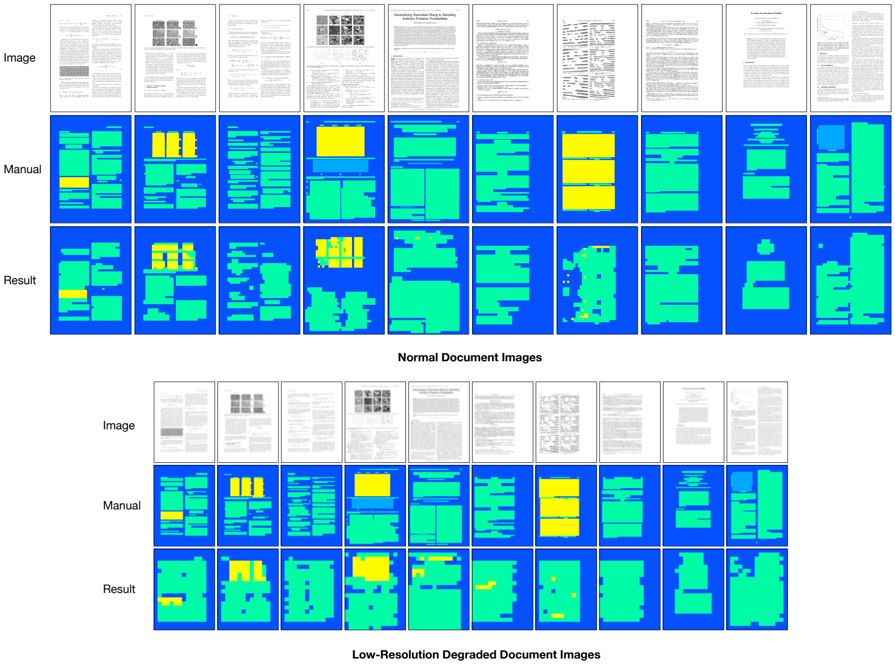

Figure 9. An Illustration of segmentation on ten example images from our test set with a 70 word codebook and a four topic model. First row images are the documents, second are the ground truth labeled images, third are the single-level results and fourth are the multi-level MRF results. Part one shows the segmentation at normal 800x600 resolution. Part two shows low-resolution 80x60 segmentation results. (Blue are whitespace, green are text, yellow are image, and light blue are figure regions)

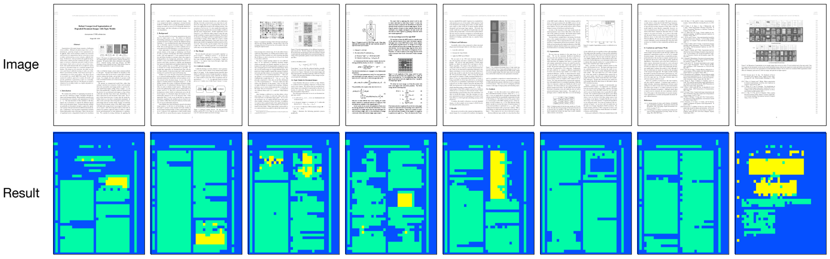

Figure 10. Example segmentation on this document. We can see that the model does very well on images of this document which was not even part of the initial training set. In this case we have no manual-labels to get quantitative results, but to the eye only figures pose a significant problem as in the prior results. These results illustrate the adaptability of the model.

[14] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1), 1980.

[15] D. Liu and T. Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. In *ICCV*, 2007.

[16] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 2001.

[17] F. Perronnin, C. R. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *ECCV*, 2006.

[18] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press Arlington, Virginia, United States, 2004.

[19] M. Tuceryan and A. K. Jain. *The Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis, pages 207–248. World Scientific Publishing Co., 1998.

[20] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned visual dictionary. In *In Proc. Intl. Conference on Computer Vision (ICCV)*, 2005.

[21] Y. Zheng, H. Li, and D. Doermann. Text identification in noisy document images using markov random field. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003.

[22] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Filters, random field and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):1–20, 1998.