Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks

Xiao Yang[‡], Ersin Yumer[†], Paul Asente[†], Mike Kraley[†], Daniel Kifer[‡], C. Lee Giles[‡]
[‡]The Pennsylvania State University [†]Adobe Research
xuy111@psu.edu {yumer, asente, mkraley}@adobe.com dkifer@cse.psu.edu giles@ist.psu.edu

Abstract

We present an end-to-end, multimodal, fully convolutional network for extracting semantic structures from document images. We consider document semantic structure extraction as a pixel-wise segmentation task, and propose a unified model that classifies pixels based not only on their visual appearance, as in the traditional page segmentation task, but also on the content of underlying text. Moreover, we propose an efficient synthetic document generation process that we use to generate pretraining data for our network. Once the network is trained on a large set of synthetic documents, we fine-tune the network on unlabeled real documents using a semi-supervised approach. We systematically study the optimum network architecture and show that both our multimodal approach and the synthetic data pretraining significantly boost the performance.

1. Introduction

Document semantic structure extraction (DSSE) is an actively-researched area dedicated to understanding images of documents. The goal is to split a document image into regions of interest and to recognize the role of each region. It is usually done in two steps: the first step, often referred to as *page segmentation*, is appearance-based and attempts to distinguish text regions from regions like figures, tables and line segments. The second step, often referred to as *logical structure analysis*, is semantics-based and categorizes each region into semantically-relevant classes like paragraph and caption.

In this work, we propose a unified multimodal fully convolutional network (MFCN) that simultaneously identifies both *appearance-based* and *semantics-based* classes. It is a generalized page segmentation model that additionally performs fine-grained recognition on text regions: text regions are assigned specific labels based on their semantic functionality in the document. Our approach simplifies DSSE and better supports document image understanding.

We consider DSSE as a pixel-wise segmentation problem: each pixel is labeled as background, figure, table, paragraph, section heading, list, caption, etc. We show that our MFCN model trained in an end-to-end, pixels-to-pixels manner on document images exceeds the state-of-the-art significantly. It eliminates the need to design complex heuristic rules and extract hand-crafted features [30, 22, 21, 46, 4].

Figure 1: (a) Examples that are difficult to identify if only based on text. The same name can be a title, an author or a figure caption. (b) Examples that are difficult to identify if only based on visual appearance. Text in the large font might be mislabeled as a section heading. Text with dashes might be mislabeled as a list.

In many cases, regions like section headings or captions can be visually identified. In Fig. 1 (a), one can easily recognize the different roles of the same name. However, a robust DSSE system needs the semantic information of the text to disambiguate possible false identifications. For example, in Fig. 1 (b), the text in the large font might look like section heading, but it does not function that way; the lines beginning with dashes might be mislabeled as a list.

To this end, our multimodal fully convolutional network is designed to leverage the textual information in the document as well. To incorporate textual information in a CNN-based architecture, we build a text embedding map and feed it to our MFCN. More specifically, we embed each sentence and map the embedding to the corresponding pixels where the sentence is represented in the document. Fig. 2 summarizes the architecture of the proposed MFCN model. Our

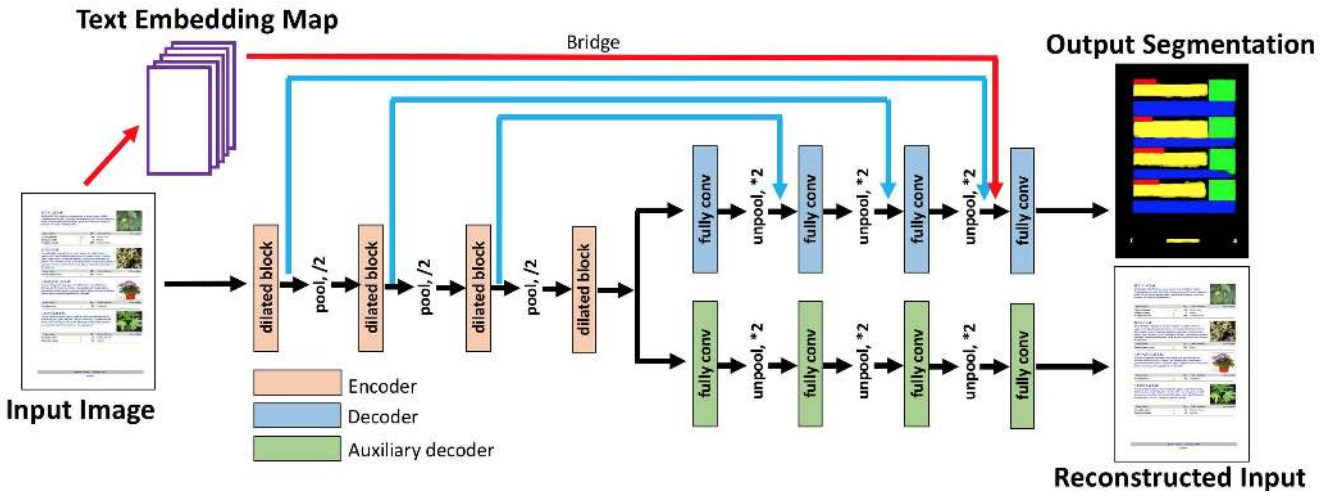

Figure 2: The architecture of the proposed multimodal fully convolutional neural network. It consists of four parts: an encoder that learns a hierarchy of feature representations, a decoder that outputs segmentation masks, an auxiliary decoder for unsupervised reconstruction, and a bridge that merges visual representations and textual representations. The auxiliary decoder only exists during training.

model consists of four parts: an encoder that learns a hierarchy of feature representations, a decoder that outputs segmentation masks, an auxiliary decoder for reconstruction during training, and a bridge that merges visual representations and textual representations. We assume that the document text has been pre-extracted. For document images this can be done with modern OCR engines [47, 1, 2].

One of the bottlenecks in training fully convolutional networks is the need for pixel-wise ground truth data. Previous document understanding datasets [31, 44, 50, 6] are limited by both their small size and the lack of fine-grained semantic labels such as section headings, lists, or figure and table captions. To address these issues, we propose an efficient synthetic document generation process and use it to generate large-scale pretraining data for our network. Furthermore, we propose two unsupervised tasks for better generalization to real documents: reconstruction and consistency tasks. The former enables better representation learning by reconstructing the input image, whereas the latter encourages pixels belonging to the same regions have similar representation.

Our main contributions are summarized as follows:

- We propose an end-to-end, unified network to address document semantic structure extraction. Unlike previous two-step processes, we simultaneously identify both *appearance-based* and *semantics-based* classes.
- Our network supports both supervised training on image and text of documents, as well as unsupervised auxiliary training for better representation learning.
- We propose a synthetic data generation process and use it to synthesize a large-scale dataset for training the supervised part of our deep MFCN model.

2. Background

Page Segmentation. Most earlier works on page segmentation [30, 22, 21, 46, 4, 45] fall into two categories: bottom-up and top-down approaches. Bottom-up approaches [30, 46, 4] first detect words based on local features (white/black pixels or connected components), then sequentially group words into text lines and paragraphs. However, such approaches suffer from the identification and grouping of connected components being time-consuming. Top-down approaches [22, 21] iteratively split a page into columns, blocks, text lines and words. With both of these approaches it is difficult to correctly segment documents with complex layout, for example a document with non-rectangular figures [38].

With recent advances in deep convolutional neural networks, several neural-based models have been proposed. Chen et al. [12] applied a convolutional auto-encoder to learn features from cropped document image patches, then use these features to train a SVM [15] classifier. Vo et al. [52] proposed using FCN to detect lines in handwritten document images. However, these methods are strictly restricted to visual cues, and thus are not able to discover the semantic meaning of the underlying text.

Logical Structure Analysis. Logical structure is defined as a hierarchy of logical components in documents, such as section headings, paragraphs and lists [38]. Early work in logical structure discovery [18, 29, 24, 14] focused on using a set of heuristic rules based on the location, font and text of each sentence. Shilman et al. [45] modeled document layout as a grammar and used machine learning to minimize the cost of a invalid parsing. Luong et al. [35] proposed using a conditional random fields model to jointly

label each sentence based on several hand-crafted features. However, the performance of these methods is limited by their reliance on hand-crafted features, which cannot capture the highly semantic context.

Semantic Segmentation. Large-scale annotations [32] and the development of deep neural network approaches such as the fully convolutional network (FCN) [33] have led to rapid improvement of the accuracy of semantic segmentation [13, 42, 41, 54]. However, the originally proposed FCN model has several limitations, such as ignoring small objects and mislabeling large objects due to the fixed receptive field size. To address this issue, Noh et al. [41] proposed using unpooling, a technique that reuses the pooled "location" at the up-sampling stage. Pinheiro et al. [43] attempted to use skip connections to refine segmentation boundaries. Our model addresses this issue by using a dilated block, inspired by dilated convolutions [54] and recent work [49, 23] that groups several layers together . We further investigate the effectiveness of different approaches to optimize our network architecture.

Collecting pixel-wise annotations for thousands or millions of images requires massive labor and cost. To this end, several methods [42, 56, 34] have been proposed to harness weak annotations (bounding-box level or image level annotations) in neural network training. Our consistency loss relies on similar intuition but does not require a "class label" for each bounding box.

Unsupervised Learning. Several methods have been proposed to use unsupervised learning to improve supervised learning tasks. Mairal et al. [36] proposed a sparse coding method that learns sparse local features by sparsity-constrained reconstruction loss functions. Zhao et al. [58] proposed a Stacked What-Where Auto-Encoder that uses unpooling during reconstruction. By injecting noise into the input and the middle features, a denoising auto-encoder [51] can learn robust filters that recover uncorrupted input. The main focus in unsupervised learning has been image-level classification and generative approaches, whereas in this paper we explore the potential of such methods for pixel-wise semantic segmentation.

Wen et al. [53] recently proposed a center loss that encourages data samples with the same label to have a similar visual representation. Similarly, we introduce an intra-class consistency constraint. However, the "center" for each class in their loss is determined by data samples across the whole dataset, while in our case the "center" is locally determined by pixels within the same region in each image.

Language and Vision. Several joint learning tasks such as image captioning [16, 28], visual question answering [5, 20, 37], and one-shot learning [19, 48, 11] have demonstrated the significant impact of using textual and visual representations in a joint framework. Our work is unique in that we use textual embedding *directly* for a segmentation task for the first time, and we show that our approach improves the results of traditional segmentation approaches that only use visual cues.

3. Method

Our method does supervised training for pixel-wise segmentation with a specialized multimodal fully convolutional network that uses a text embedding map jointly with the visual cues. Moreover, our MFCN architecture also supports two unsupervised learning tasks to improve the learned document representation: a reconstruction task based on an auxiliary decoder and a consistency task evaluated in the main decoder branch along with the per-pixel segmentation loss.

3.1. Multimodal Fully Convolutional Network

As shown in Fig. 2, our MFCN model has four parts: an encoder, two decoders and a bridge. The encoder and decoder parts roughly follow the architecture guidelines set forth by Noh et al. [41]. However, several changes have been made to better address document segmentation.

First, we observe that several semantic-based classes such as section heading and caption usually occupy relatively small areas. Moreover, correctly identifying certain regions often relies on small visual cues, like lists being identified by small bullets or numbers in front of each item. This suggests that low-level features need to be used. However, because max-pooling naturally loses information during downsampling, FCN often performs poorly for small objects. Long et al. [33] attempt to avoid this problem using skip connections. However, simply averaging independent predictions based on features at different scales does not provide a satisfying solution. Low-level representations, limited by the local receptive field, are not aware of object-level semantic information; on the other hand, high-level features are not necessarily aligned consistently with object boundaries because CNN models are invariant to translation. We propose an alternative skip connection implementation, illustrated by the blue arrows in Fig. 2, similar to that used in the independent work *SharpMask* [43]. However, they use bilinear upsampling after skip connection while we use unpooling to preserve more spatial information.

We also notice that broader context information is needed to identify certain objects. For an instance, it is often difficult to tell the difference between a list and several paragraphs by only looking at parts of them. In Fig. 3, to correctly segment the right part of the list, the receptive fields must be large enough to capture the bullets on the left. Inspired by the Inception architecture [49] and dilated convolution [54], we propose a dilated convolution block, which is illustrated in Fig. 4 (left). Each dilated convolution block consists of 5 dilated convolutions with a 3×3 kernel size and a dilation $d = 1, 2, 4, 8, 16$.

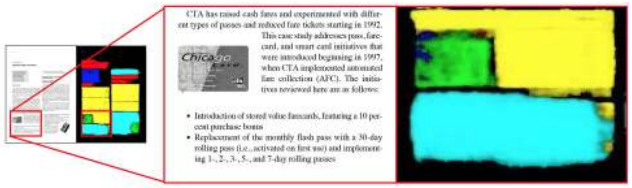

Figure 3: A cropped document image and its segmentation mask generated by our model. Note that the top-right corner of the list is yellow instead of cyan, indicating that it has been mislabeled as a paragraph.

3.2. Text Embedding Map

Traditional image semantic segmentation models learn the semantic meanings of objects from a visual perspective. Our task, however, also requires understanding the text in images from a linguistic perspective. Therefore, we build a text embedding map and feed it to our multimodal model to make use of both visual and textual representations.

We treat a sentence as the minimum unit that conveys certain semantic meanings, and represent it using a low-dimensional vector. Our sentence embedding is built by averaging embeddings for individual words. This is a simple yet effective method that has been shown to be useful in many applications, including sentiment analysis [26] and text classification [27]. Using such embeddings, we create a text embedding map as follows: for each pixel inside the area of a sentence, we use the corresponding sentence embedding as the input. Pixels that belong to the same sentence thus share the same embedding. Pixels that do not belong to any sentences will be filled with zero vectors. For a document image of size $H \times W$, this process results in an embedding map of size $N \times H \times W$ if the learned sentence embeddings are N-dimensional vectors. The embedding map is later concatenated with a feature response along the number-of-channel dimensions (see Fig. 2).

Specifically, our word embedding is learned using the skip-gram model [39, 40]. Fig. 4 (right) shows the basic diagram. Let V be the number of words in a vocabulary and w be a V-dimensional one-hot vector representing a word. The training objective is to find a N-dimensional ($N \ll V$) vector representation for each word that is useful for predicting the neighboring words. More formally, given a sequence of words $[w_1, w_2, \cdots, w_T]$, we maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-C\leq j\leq C, j\neq 0} \log P(w_{t+j}|w_t) \qquad (1)$$

where T is the length of the sequence and C is the size of the context window. The probability of outputting a word

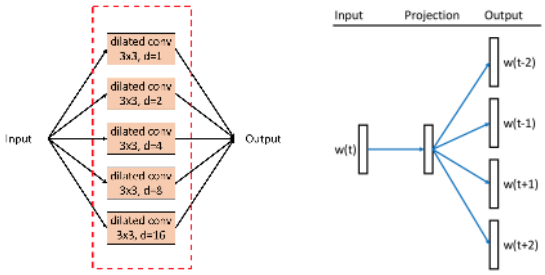

Figure 4: Left: A dilated block that contains 5 dilated convolutional layers with different dilation d. Batch-Normalization and non-linearity are not shown for brevity. Right: The skip-gram model for word embeddings.

w_o given an input word w_i is defined using softmax:

$$P(w_o|w_i) = \frac{\exp(v'^{\top}_{w_o} v_{w_i})}{\sum_{w=1}^{V} \exp(v'^{\top}_{w} v_{w_i})} \qquad (2)$$

where v_w and v'_w are the "input" and "output" N-dimensional vector representations of w.

3.3. Unsupervised Tasks

Although our synthetic documents (Sec. 4) provide a large amount of labeled data for training, they are limited in the variations of their layouts. To this end, we define two unsupervised loss functions to make use of real documents and to encourage better representation learning.

Reconstruction Task. It has been shown that reconstruction can help learning better representations and therefore improves performance for supervised tasks [58, 57]. We thus introduce a second decoder pathway (Fig. 2 - axillary decoder), denoted as D_{rec}, and define a reconstruction loss at intermediate features. This auxiliary decoder only exists during the training phase.

Let $a_l, l = 1, 2, \cdots L$ be the activations of the l^{th} layer of the encoder, and a_0 be the input image. For a feed-forward convolutional network, a_l is a feature map of size $C_l \times H_l \times W_l$. Our auxiliary decoder D_{rec} attempts to reconstruct a hierarchy of feature maps $\{\tilde{a}_l\}$. Reconstruction loss $L_{rec}^{(l)}$ for a specific l is therefore defined as

$$L_{rec}^{(l)} = \frac{1}{C_l H_l W_l} \|a_l - \tilde{a}_l\|_2^2, \quad l = 0, 1, 2, \cdots L \qquad (3)$$

Consistency Task. Pixel-wise annotations are labor-intensive to obtain, however it is relatively easy to get a set of bounding boxes for detected objects in a document. For documents in PDF format, one can find bounding boxes by analyzing the rendering commands in the PDF files (See our supplementary document for typical examples). Even if their labels remain unknown, these bounding boxes are still beneficial: they provide knowledge of which parts of a document belongs to the same objects and thus should not be segmented into different fragments.

By building on the intuition that regions belonging to same objects should have similar feature representations, we define the consistency task loss L_{cons} as follows. Let $p_{(i,j)}$ $(i = 1, 2, \cdots H, j = 1, 2, \cdots W)$ be activations at location (i, j) in a feature map of size $C \times H \times W$, and b be the rectangular area in a bounding box. Let each rectangular area b is of size $H_b \times W_b$. Then, for each $b \in B$, L_{cons} will be given by

$$L_{cons} = \frac{1}{H_b W_b} \sum_{(i,j) \in b} \left\| p_{(i,j)} - p^{(b)} \right\|_2^2 \tag{4}$$

$$p^{(b)} = \frac{1}{H_b W_b} \sum_{(i,j) \in b} p_{(i,j)} \tag{5}$$

Minimizing consistency loss L_{cons} encourages intra-region consistency.

The consistency loss L_{cons} is differentiable and can be optimized using stochastic gradient descent. The gradient of L_{cons} with respect to $p_{(i,j)}$ is

$$\begin{aligned} \frac{\partial L_{cons}}{\partial p_{(i,j)}} =& \frac{2}{H_b^2 W_b^2} (p_{(i,j)} - p^{(b)})(H_b W_b - 1) + \\ & \frac{2}{H_b^2 W_b^2} \sum_{\substack{(u,v) \in b \\ (u,v) \neq (i,j)}} (p^{(b)} - p_{(u,v)}) \end{aligned} \tag{6}$$

since $H_b W_b \gg 1$, for efficiency it can be approximated by:

$$\frac{\partial L_{cons}}{\partial p_{(i,j)}} \approx \frac{2}{H_b W_b} \left(p_{(i,j)} - p^{(b)} \right). \tag{7}$$

We use the unsupervised consistency loss, L_{cons}, as a loss layer, that is evaluated at the main decoder branch (blue branch in Fig. 2) along with supervised segmentation loss.

4. Synthetic Document Data

Since our MFCN aims to generate a segmentation mask of the whole document image, pixel-wise annotations are required for the supervised task. While there are several publicly available datasets for page segmentation [44, 50, 6], there are only a few hundred to a few thousand pages in each. Furthermore, the types of labels are limited, for example to text, figure and table, however our goal is to perform a much more granular segmentation.

To address these issues, we created a synthetic data engine, capable of generating large-scale, pixel-wise annotated documents.

Our synthetic document engine uses two methods to generate documents. The first produces completely automated and random layout of partial data scraped from the web. More specifically, we generate LaTeX source files in which paragraphs, figures, tables, captions, section headings and lists are randomly arranged to make up single, double, or triple-column PDFs. Candidate figures include academic-style figures and graphic drawings downloaded using web image search, and natural images from MS COCO [32], which associates each image with several captions. Candidate tables are downloaded using web image search. Various queries are used to increase the diversity of downloaded tables. Since our MFCN model relies on the semantic meaning of text to make prediction, the content of text regions (paragraph, section heading, list, caption) must be carefully selected:

- For paragraphs, we randomly sample sentences from a 2016 English Wikipedia dump [3].
- For section headings, we only sample sentences and phrases that are section or subsection headings in the "Contents" block in a Wikipedia page.
- For lists, we ensure that all items in a list come from the same Wikipedia page.
- For captions, we either use the associated caption (for images from MS COCO) or the title of the image in web image search, which can be found in the span with class name "irc_pt".

To further increase the complexity of the generated document layouts, we collected and labeled 271 documents with varied, complicated layouts. We then randomly replaced each element with a standalone paragraph, figure, table, caption, section heading or list generated as stated above.

In total, our synthetic dataset contains 135,000 document images. Examples of our synthetic documents are shown in Fig. 5. Please refer to our supplementary document for more examples of synthetic documents and individual elements used in the generation process.

5. Implementation Details

Fig. 2 summarizes the architecture of our model. The auxiliary decoder only exists in the training phase. All convolutional layers have a 3×3 kernel size and a stride of 1. The pooling (in the encoders) and unpooling (in the decoders) have a kernel size of 2×2. We adopt batch normalization [25] immediately after each convolution and before all non-linear functions.

We perform per-channel mean subtraction and resize each input image so that its longer side is less than 384 pixels. No other pre-processing is applied. We use Adadelta [55] with a mini-batch size of 2. During semi-supervised training, mini-batches of synthetic and real documents are used alternatively. For synthetic documents, both per-pixel classification loss and the unsupervised losses are active at back-propagation, while for real documents, only the unsupervised losses are active. Since the labels are unbalanced (e.g. the area of paragraphs is

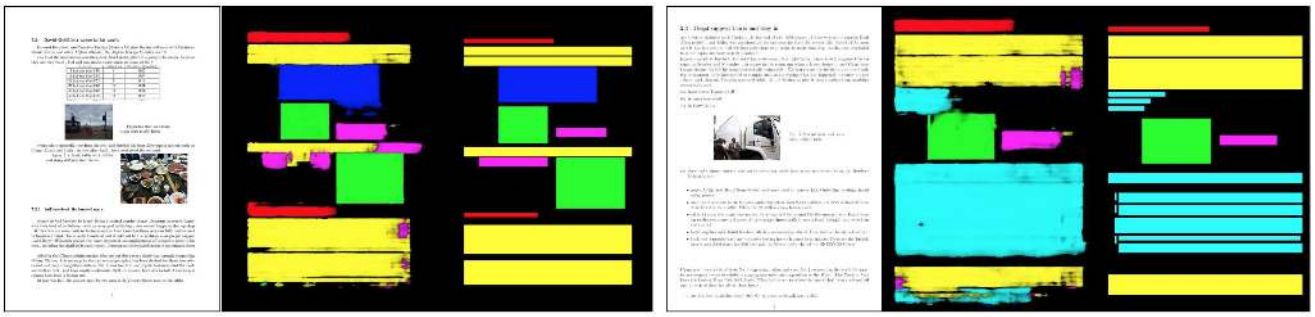

Figure 5: Example synthetic documents, raw segmentations and results after optional post-processing (Sec. 5). Segmentation label colors are: **figure**, **table**, **section heading**, **caption**, **list** and **paragraph**.

much larger than that of caption), class weights for the per-pixel classification loss are set differently according to the total number of pixels in each class in the training set.

For text embedding, we represent each word as a 128-dimensional vector and train a skip-gram model on the 2016 English Wikipedia dump [3]. Embeddings for out-of-dictionary words are obtained following Bojanowski et al. [9]. We use Tesseract [47] as our OCR engine.

Post-processing. We apply an optional post-processing step as a cleanup strategy for segment masks. For documents *in PDF format*, we obtain a set of candidate bounding boxes by analyzing the PDF format to find element boxes. We then refine the segmentation masks by first calculating the average class probability for pixels belonging to the same box, followed by assigning the most likely label to these pixels.

6. Experiments

We used three datasets for evaluations: ICDAR2015 [6], SectLabel [35] and our new dataset named DSSE-200. ICDAR2015 [6] is a dataset used in the biennial ICDAR page segmentation competitions [7] focusing more on appearance-based regions. The evaluation set of ICDAR2015 consists of 70 sampled pages from contemporary magazines and technical articles. SectLabel [35] consists of 40 academic papers with 347 pages in the field of computer science. Each text line in these papers is manually assigned a semantics-based label such as text, section heading or list item. In addition to these two datasets, we introduce DSSE-200[1], which provides both appearance-based and semantics-based labels. DSSE-200 contains 200 pages from magazines and academic papers. Regions in a page are assigned labels from the following dictionary: figure, table, section, caption, list and paragraph. Note that DSSE-200 has a more granular segmentation than previously released benchmark datasets.

The performance is measured in terms of pixel-wise intersection-over-union (IoU), which is standard in semantic segmentation tasks. We optimize the architecture of our MFCN model based on the DSSE-200 dataset since it contains both appearance-based and semantics-based labels. Sec. 6.4 compares our results to state-of-the-art methods on the ICDAR2015 and SectLabel datasets.

6.1. Ablation Experiment on Model Architecture

We first systematically evaluate the effectiveness of different network architectures. Results are shown in Table 1. Note that these results do not incorporate textual information or unsupervised learning tasks. The purpose of this experiment is to find the best "base" architecture to be used in the following experiments. All models are trained from scratch and evaluated on the DSSE-200 dataset.

As a simple baseline (Table 1 Model1), we train a plain encoder-decoder style model for document segmentation. It consists of a feed-forward convolutional network as an encoder, and a decoder implemented by a fully convolutional network. Upsampling is done by bilinear interpolation. This model achieves a mean IoU of 61.4%.

Next, we add skip connections to the model, resulting in Model2. Note that this model is similar to the *SharpMask* model. We observe a mean IoU of 65.4%, 4% better than the base model. The improvements are even more significant for small objects like captions.

We further evaluate the effectiveness of replacing bilinear upsampling with unpooling, giving Model3. All upsampling layers in Model2 are replaced by unpooling while other parts are kept unchanged. Doing so results in a significant improvement for mean IoU (65.4% vs. 71.2%). This suggests that the pooled index should not be discarded during decoding. These indexes are helpful to disambiguate the location information when constructing the segmentation mask in the decoder.

Finally, we investigate the use of dilated convolutions. Model3 is equivalent to using dilated convolution when $d = 1$. Model4 sets $d = 8$ while Model5 uses the dilated block illustrated in Fig. 4 (left). The number of output channels are adjusted such that the total number of parame-

[1] http://personal.psu.edu/xuy111/projects/cvpr2017_doc.html.

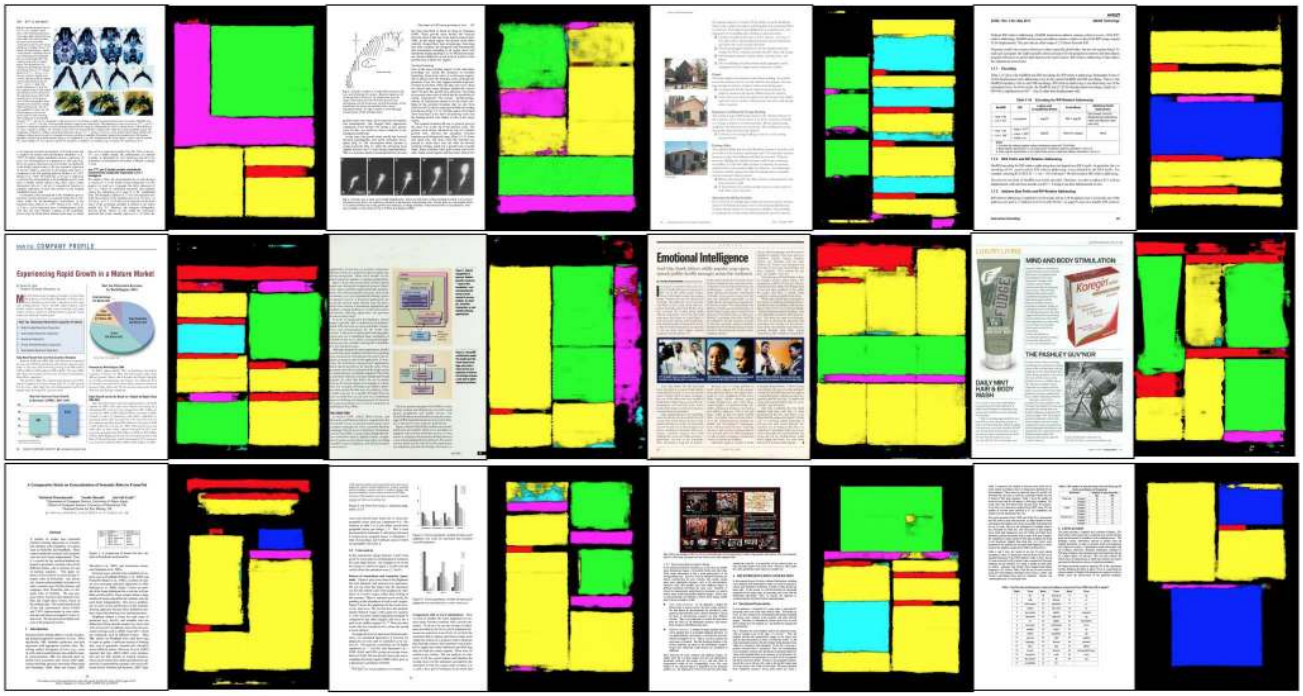

Figure 6: Example real documents and their corresponding segmentation. Top: DSSE-200. Middle: ICDAR2015. Bottom: SectLabel. Since these documents are not in PDF format, the simple post-processing in Sec. 5 can not be applied. One may consider exploiting a CRF [13] to refine the segmentation, but that is beyond the main focus of this paper. Segmentation label colors are: **figure**, **table**, **section heading**, **caption**, **list** and **paragraph**.

Model#	dilation	upsampling	skip	bkg	figure	table	section	caption	list	paragraph	mean
1	1	bilinear	no	80.3	75.4	62.7	50.0	33.8	57.3	70.4	61.4
2	1	bilinear	yes	82.1	76.7	74.4	51.8	42.4	58.7	74.4	65.4
3	1	unpooling	yes	84.1	81.2	77.6	54.6	60.3	65.9	74.8	71.2
4	8	unpooling	yes	83.9	74.9	69.7	57.2	60.2	64.6	76.1	69.5
5	block	unpooling	yes	**84.6**	**83.3**	**79.4**	**58.3**	**61.0**	**66.7**	**77.1**	**73.0**

Table 1: Ablation experiments on DSSE-200 dataset. The architecture of each model is characterized by the dilation in convolution layers, the way of upsampling and the use of skip connection. IoU scores (%) are reported.

ters are similar. Comparing the results for these three models, we can see that the IoU of Model4 for each class is on par with or worse than Model3, while Model5 is better than both Model3 and Model4 for all classes.

6.2. Adding Textual Information

We now investigate the importance of textual information in our multimodal model. We take the best architecture, Model5, as our vision-only model, and incorporate a text embedding map via a bridge module depicted in Fig. 2. This combined model is fine-tuned on our synthetic documents. As shown in Table 2, using text as well improves the performance for *textual* classes. The accuracy for section heading, caption, list and paragraph is boosted by 1.1%, 0.1%, 1.7% and 2.2%, respectively.

We rely on existing OCR engines [47] to extract text, but they are not always reliable for scanned documents of low quality. To quantitatively analyze the effects of using extracted text, we compare the performance of using extracted text versus real text. The comparison is conducted on a subset of our synthetic dataset (200 images), since ground-truth text is naturally available. As shown in Table 2, using real text leads to a remarkable improvement (6.4%) for mean IoU, suggesting the effectiveness of incorporating textual information. Using OCR extracted text is not as effective, but still results in 2.6% improvement. It is better than the 0.3% improvement on DSSE-200 dataset; we attribute this to our synthetic data not being as complicated as DSSE-200, so extracting text becomes easier.

base	dataset	text	bkg	figure	table	section	caption	list	para.	mean
Model5	D	none	**84.6**	83.3	**79.4**	58.3	61.0	66.7	77.1	73.0
Model5	D	extract	83.9	**83.7**	79.7	**59.4**	**61.1**	**68.4**	**79.3**	**73.3**
Model5	S	none	87.7	83.1	84.3	70.8	70.9	82.3	83.1	79.6
Model5	S	extract	88.8	85.4	86.6	73.1	71.2	83.6	87.2	82.2
Model5	S	real	**91.2**	**90.3**	**89.0**	**78.4**	**75.3**	**87.5**	**89.6**	**86.0**

Table 2: IoU scores (%) on the DSSE-200 (D) and synthetic dataset (S) using text embedding map. On synthetic dataset, we further investigate the effects of using extracted text versus real text when building the text embedding map.

	L_{cls}	L_{rec}	L_{cons}	$L_{rec+con}$
mean	73.3	73.9	75.4	75.9

Table 3: IoU scores (%) when using different training objectives on DSSE-200 dataset. *cls*: pixel-wise classification task, *rec*: reconstruction task and *cons*: consistency task.

Methods	non-text	text
Leptonica [8]	84.7	86.8
Bukhari et al. [10]	90.6	90.3
Ours (binary)	**94.5**	**91.0**
Methods	**figure**	**text**
Fernandez et al. [17]	70.1	85.8
Ours (binary)	**77.1**	**91.0**

Table 4: IoU scores (%) for page segmentation on the ICDAR2015 dataset. For comparison purpose, only IoU scores for non-text, text and figure are shown. However our model can make fine-grained predictions as well.

Methods	section	caption	list	para.
Luong et al. [35]	0.916	0.781	0.712	**0.969**
Ours	**0.919**	**0.893**	**0.793**	**0.969**

Table 5: F1 scores on the SectLabel dataset. Note that our model can also identify non-text classes such as figures and tables.

6.3. Unsupervised Learning Tasks

Here, we examine how the proposed two unsupervised learning tasks — reconstruction and consistency tasks — can complement the pixel-wise classification during training. We take the best model in Sec. 6.2, and only change the training objectives. Our model is then fine-tuned in a semi-supervised manner as described in Sec. 5. The results are shown in Table 3. Adding the reconstruction task slightly improves the mean IoU by 0.6%, while adding the consistency task leads to a boost of 1.9%. These results justify our hypothesis that harnessing region information is beneficial. Combining both tasks results in a mean IoU of 75.9%.

6.4. Comparisons with Prior Art

Table 4 and 5 present comparisons with several methods that have previously reported performance on the ICDAR2015 and SectLabel datasets. It is worth emphasizing that our MFCN model simultaneously predicts both appearance-based and semantics-based classes while other methods can not.

Comparisons on ICDAR2015 dataset (Table 4). Previous pixel-wise page segmentation models usually solve a binary segmentation problem and do not make predictions for fine-grained classes. For fair comparison, we change the number of output channels of the last layer to 3 (background, figure and text) and fine-tune this last layer. Our binary MFCN model achieves 94.5%, 91.0% and 77.1% IoU scores for non-text (background and figure), text and figure regions, outperforming other models.

Comparisons on SectLabel dataset (Table 5). Luong et at. [35] first use Omnipage [2] to localize and recognize text lines, then predict the semantics-based label for each line. The F1 score for each class was reported. For fair comparison, we use the same set of text line bounding boxes, and use the averaged pixel-wise prediction as the label for each text line. Our model achieves better F1 scores for section heading (0.919 VS 0.916), caption (0.893 VS 0.781) and list (0.793 VS 0.712), while being capable of identifying figures and tables.

7. Conclusion

We proposed a multimodal fully convolutional network (MFCN) for document semantic structure extraction. The proposed model uses both visual and textual information. Moreover, we propose an efficient synthetic data generation method that yields per-pixel ground-truth. Our unsupervised auxiliary tasks help boost performance tapping into unlabeled real documents, facilitating better representation learning. We showed that both the multimodal approach and unsupervised tasks can help improve performance. Our results indicate that we have improved the state of the art on previously established benchmarks. In addition, we are publicly providing the large synthetic dataset (135,000 pages) as well as a new benchmark dataset: DSSE-200.

Acknowledgment

This work started during Xiao Yang's internship at Adobe Research. This work was supported by NSF grant CCF 1317560 and Adobe Systems Inc.

References

[1] Abbyy. https://www.abbyy.com/. 2

[2] Omnipage. https://goo.gl/nDQEpC. 2, 8

[3] Wikipedia. https://dumps.wikimedia.org/. 5, 6

[4] A. Amin and R. Shiu. Page segmentation and classification utilizing bottom-up approach. *International Journal of Image and Graphics*, 1(02):345–361, 2001. 1, 2

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 3

[6] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher. A realistic dataset for performance evaluation of document layout analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 296–300. IEEE, 2009. 2, 5, 6

[7] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher. Icdar2015 competition on recognition of documents with complex layouts-rdcl2015. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1151–1155. IEEE, 2015. 6

[8] D. S. Bloomberg and L. Vincent. Document image applications. *Morphologie Mathmatique*, 2007. 8

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016. 6

[10] S. S. Bukhari, F. Shafait, and T. M. Breuel. Improved document image segmentation algorithm using multiresolution morphology. In *IS&T/SPIE Electronic Imaging*, pages 78740D–78740D. International Society for Optics and Photonics, 2011. 8

[11] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *arXiv preprint arXiv:1603.00550*, 2016. 3

[12] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold. Page segmentation of historical document images with convolutional autoencoders. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1011–1015. IEEE, 2015. 2

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 3, 7

[14] A. Conway. Page grammars and page parsing. a syntactic approach to document layout recognition. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 761–764. IEEE, 1993. 2

[15] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[16] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 3

[17] F. C. Fernández and O. R. Terrades. Document segmentation using relative location features. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1562–1565. IEEE, 2012. 8

[18] J. L. Fisher. Logical structure descriptions of segmented document images. *Proceedings of International Con ference on Document Analysis and Recognition*, pages 302–310, 1991. 2

[19] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 3

[20] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in Neural Information Processing Systems*, pages 2296–2304, 2015. 3

[21] J. Ha, R. M. Haralick, and I. T. Phillips. Document page decomposition by the bounding-box project. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 2, pages 1119–1122. IEEE, 1995. 1, 2

[22] J. Ha, R. M. Haralick, and I. T. Phillips. Recursive xy cut using bounding boxes of connected components. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 2, pages 952–955. IEEE, 1995. 1, 2

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[24] R. Ingold and D. Armangil. A top-down document analysis method for logical structure recognition. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 41–49, 1991. 2

[25] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[26] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*, 2015. 4

[27] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016. 4

[28] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 3

[29] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):737–747, 1993. 2

[30] F. Lebourgeois, Z. Bublinski, and H. Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pages 272–276. IEEE, 1992. 1, 2

[31] J. Liang, R. Rogers, R. M. Haralick, and I. T. Phillips. Uwisl document image analysis toolbox: An experimental environment. In *Document Analysis and Recognition, 1997.,*

Proceedings of the Fourth International Conference on, volume 2, pages 984–988. IEEE, 1997. 2

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3, 5

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 3

[34] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao. Learning from weak and noisy labels for semantic segmentation. 2016. 3

[35] M.-T. Luong, T. D. Nguyen, and M.-Y. Kan. Logical structure recovery in scholarly articles with rich document features. *Multimedia Storage and Retrieval Innovations for Digital Library Systems*, 270, 2012. 2, 6, 8

[36] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009. 3

[37] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–9, 2015. 3

[38] S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms: a literature survey. In *Electronic Imaging 2003*, pages 197–207. International Society for Optics and Photonics, 2003. 2

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 4

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 4

[41] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 3

[42] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 3

[43] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3

[44] J. Sauvola and H. Kauniskangas. Mediateam document database ii. *A CD-ROM collection of document images, University of Oulu Finland*, 1999. 2, 5

[45] M. Shilman, P. Liang, and P. Viola. Learning nongenerative grammatical models for document analysis. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 962–969. IEEE, 2005. 2

[46] A. Simon, J.-C. Pret, and A. P. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277, 1997. 1, 2

[47] R. Smith. An overview of the tesseract ocr engine. 2007. 2, 6, 7

[48] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 3

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 3

[50] L. Todoran, M. Worring, and A. W. Smeulders. The uva color document dataset. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(4):228–240, 2005. 2, 5

[51] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 3

[52] Q. N. Vo and G. Lee. Dense prediction for text line segmentation in handwritten document images. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3264–3268. IEEE, 2016. 2

[53] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 3

[54] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[55] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 5

[56] W. Zhang, S. Zeng, D. Wang, and X. Xue. Weakly supervised semantic segmentation for social images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2718–2726, 2015. 3

[57] Y. Zhang, E. K. Lee, E. H. Lee, and U. EDU. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. 4

[58] J. Zhao, M. Mathieu, R. Goroshin, and Y. Lecun. Stacked what-where auto-encoders. *arXiv preprint arXiv:1506.02351*, 2015. 3, 4