Document Conversion to LaTeX with OCR

Logan Short, Christopher Wong, and David Zeng

***Abstract*—agg**

1 Introduction

OPTICAL character recognition (OCR) is the process of converting images of typed or handwritten text into digital characters. In particular, we are considering the problem of using OCR techniques to turn an image of a typesetted document into LaTeX, a markup language commonly used to typeset scientific literature. The motivation for our problem comes from the fact that many old books and academic papers exist online only as scanned images, and to be able to convert these images into formattable LaTeX documents would allow these documents to be much more easily maintained by websites and readers. These documents could also be indexed and searched by search engines, and perhaps in the future we will be able to query academic papers directly with LaTeX markup.

2 Related Work in OCR

agg

2.1 *Summary of Previous Work*

microsfot sliding window paper

In [2], Frey and Slate

In [3], Gupta et al. discuss methods for identifying document layout structure from images of documents. The key intuition behind the algorithm proposed in [3] is that the various components of documents, in this case technical papers, can be identified using the features of the bounding boxes encapsulating each section as well as the realtive positioning and ordering of said bounding boxes. Gupta et al. first preprocess their document images by thresholding at 80% of the image intensity. Characters in the document are then located by identifying contours in the thresholded image and bounding boxes are formed around these characters. Bounding boxes that are close together are then combined horizontally to yield line level bounding boxes and combined vertically to obtain paragraph level bounding boxes. Following the construction of the paragraph level bounding boxes, features such as aspect ratio and average line height are used to differentiate between textual components and graphical components. At this point the algorithm switches over to a layout component classification

In a 2004 paper, Tapia and Rojas describes an algorithm for learning the appropriate LaTeX layout for a handwritten mathematical expression. The paper assumes that for a given expression, the individual symbols in the expressions have already been discovered and labeled, and a bounding box has been drawn around each. Given such a setup, Tapia and Rojas define a concept they call dominance. Each symbol dominates a region around it; for example, a summation symbol dominates the region above and below it, where subexpressions are usually located that define the range for the summation. To construct the appropriate LaTeX layout for the expression, Tapia and Rojas notice that LaTeX markup essentially forms a tree structure, where arguments, subscripts, and superscripts are essentially children of some given parent symbol. They devise a method for learning this tree structure. Each symbol is treated as a node and edges between two symbols are weighted based on the distance between the centroids of the bounding boxes of the two symbols. First, the concept of dominance is used to find a dominant baseline of symbols in an expression. The remaining symbols are attached to this baseline by finding the minimum spanning tree that includes the baseline. In terms of the LaTeX structure, the dominant baseline provides the sequence of top level symbols, whereas the branches of the minimum spanning tree that connect to the baseline are the arguments, superscripts, and subscripts of the baseline symbols. If these branches off of the baseline are complex enough, this process is recusively applied to find baselines for each branch, from which smaller minimum spanning trees are then built of the baseline for each of the branches.

2.2 *Main Contributions of Our Work*

The goal of our project is to create an end-to-end system that takes in as input an image of a typesetted document and outputs the matching LaTeX for the document. Our main contribution resides not so much in improving upon the OCR techniques described in the previous work, but rather in designing a top-down approach for converting a document image into LaTeX by building upon smaller modules that utilize the OCR techniques in previous literature. Many of the OCR techniques we have cited are targeted at isolated problems such as recognition of handwritten characters or math equations. Since we are instead concerned with full images of typesetted documents, the technical challenges at each level of our system are related to but slightly different from those discussed in the corresponding prior works. Details of the techniques we use in our implementation will be discussed in Section 3.2.

3 Implementation

We begin by giving an overview of our system in Section 3.1, and in Section 3.2, we give specific implementation details of each part of our algorithm.

3.1 *Summary of Implementation*

The key tasks in constructing a LaTeX representation from an image of a document are extracting the document contents and information from the image and converting this information into LaTeX form.

OCR techniques for extracting information from a document achieve the highest accuracy when the document is oriented in a left to right and top to bottom manner. Scans or photos of documents are not always of optimal quality and thus document images are not guaranteed to be oriented exactly in this ideal fashion. It is therefore advantageous for our algorithm to first rotate the document image to the proper orientation before any other techniques for extracting document information are utilized.

Given a document image whose orientation is properly aligned in a left to right and top to bottom manner, the next step in extracting document contents is to identify the sections of the image that contain textual information. Focusing on these sections exclusively allows for finer tuned character recognition and information extraction. In general, for academic papers document information is primarily contained in plain text sections and display mode equation blocks. The next phase of our implementation thus deciphers which parts of the document image correspond to these information containing sections. The structure of information differs between the two types of information blocks, thus during this step it is important to record the type of a document section as well as its location in the image.

Analyzing blocks of text is performed using a hierarchical approach. The algorithm begins by splitting text blocks into lines of text. Each of these lines of text is then split into its component words and the text of each word is then determined using a text classifier. The deciphered word texts are then concatenated horizontally to form textual representations of each line and these lines are then concatenated vertically to obtain the contents of the text block.

Identifying the contents of display mode equation sections uses a similar approach where the section is broken down into its individual equations. Each equation is then analyzed to obtain a LaTeX representation. The LaTeX layout for each equation is learned using the method in the paper by Tapia and Rojas described in section 2.1. The obtained LaTeX representations are then encapsulated into a display mode LaTeX block which is returned as the contents of the equation section.

After the contents of each document section are extracted from the image, a LaTeX representation of the document is constructed by concatenating the contents of each individual section in the top to bottom order in which they appear. Layout is kept simple as the main focus of the implementation is to maximize the accuracy of the data contained in the document.

3.2 *Implementation Details*

agg

3.2.1 Rotation Fixing: agg

3.2.2 Text and Equation Sectioning: agg

3.2.3 Text Analyzing: agg

3.2.4 Equation Analyzing: agg

3.2.5 Latex Construction: agg

4 Experimentation Results

agg

5 Conclusion

agg

References

[1] C. Jacobs, P. Simard, P. Viola, J. Rinker. Text Recognition of Low-resolution Document Images. Microsoft, 2005.

[2] P. Frey, D. Slate. Letter Recognition Using Holland-Style Adaptive Classifiers. Machine Learning, 1991.

[3] G. Gupta, S. Niranjan, A. Shrivastava. Document Layout Analysis & Classification and Its Application in OCR. EDOCW, 2006.

[4] Agg