CLEAT:
A CLassification, Enhancement and Analysis Toolkit for Heterogeneous Document Image Collections

A Proposal to

Centre for Strategic Infocomm Technologies
Winston Heng Kwong Huai, Singapore

from

Laboratory for Language and Media Processing
University of Maryland, College Park, MD, USA

September 20, 2006

Table of Contents

1 Executive Summary

The challenges related to the analysis of large heterogeneous collections of document images ultimately encompass almost all aspects of the field of document image processing. The written language takes on many forms that differ in presentation and content, yet the trained individual can interpret the visual language rather simply. The goal of document analysis is ultimately to be able to make an informed interpretation about the intended message of the visual language.

In this work, we propose to develop specific modules of interest to the sponsors related to Triage, Enhancement, Segmentation, and Content Labeling. The work will be accomplished by researchers in the Laboratory for Language and Media Processing (LAMP) at the University of Maryland and integrated with an existing infrastructure for document image analysis. The proposal contains an in-depth discussion of the problems and lays a roadmap for addressing them. We anticipate further conversations with the sponsors will focus the directions outlined here.

We assume, at the lowest level, we are given an image that may contain useful document related content. Our goal is to first determine if the image does contain document content, then to enhance and process it to the point of sufficient layout metadata to support down stream content processing such as optical character recognition. To support focused research we will develop the necessary tools, gather ground truth, visualize results, and provide efficient implementations of the algorithms we develop.

2 Scope, Approach, and Methodology

Document analysis research has focused on many different problems during the past five decades. In one continuing issue, however, many problems have been addressed in isolation. For example, OCR is often applied to handwritten or machine printed documents assuming there is a triage component that can distinguish between the different kinds of text. In fact, before making the handwritten/machine printed decision, the system assumes it knows it has a “document” with text, as opposed to a scene image. In order to produce a system that can process a truly heterogeneous stream of documents, all intermediate decisions need to be addressed and managed. A key component where all of our solutions will fit is the underlying architecture.

A second issue involves the limited work done on highly degraded data. Documents with noise and clutter often cause traditional methods to fail. Solutions require either cleaning the document before applying traditional methods, modifying traditional methods to deal with these degradations, or a combination. We will address specific quality estimation issues and provide enhancement modules to facilitate improvements in later processing.

2.1 Background

The work in this project will be built using an existing systematic development approach and shared architecture, DOCLIB, which has been successfully used in collaboration across academia, business, and government environments. DOCLIB has been developed with the intention of providing basic document and image-processing capabilities, as well as a platform for programmers to easily develop their own application on top of DOCLIB. The applications will be accessible through a well-documented, easy to use interface (API or command line). We deemed C++ the most appropriate programming language for this effort, especially for the support of a functional, stable, and robust programming interface, including data structures facilitating the collaborative development of research capabilities. For these reasons, we considered it important that DOCLIB support a plug-and-play architecture that allows straightforward addition of new image types and their conversion. We planned DOCLIB as a mechanism for easily transferring and communicating software-related research ideas. Thus, DOCLIB must to be scalable, flexible, and extendable, allowing additional functionality as needed. We, therefore, conceived an add-on mechanism that allows research groups or organizations

to augment DOCLIB easily with new features that, for various reasons, do not belong to the core of DOCLIB, without modifications to the existing DOCLIB code. The solution implemented also allows research groups/organizations to "plug-in" confidential or proprietary code. The capabilities that this project adds to DOCLIB will be a fundamental test for the architecture.

2.2 Technical Work plan

In this section, we define the core topics and the work plan involved in the research for this project. Each topic will be described briefly, followed an outline of our approach to solving it. These topics will be integrated with existing LAMP software DOCLIB to provide a toolkit that can selectively process a heterogeneous stream of images and perform detailed analysis on those document images. The key areas include data collection, classification of document type, enhancement, layout analysis, and content classification.

2.2.1 Data collection

An essential component of this project involves collecting a large, heterogeneous collection of documents as a test set. The collection will draw on existing datasets as well as targeted collection. After collecting the data the next essential step is the ability to provide ground truth and visualize the results of processing in a way that demonstrates performance.

2.2.1.1 Document Images

Beginning in the first month of the project, we will assemble a collection of document images to train and test the developed algorithms. The collection will consist of approximately 10,000 pages consisting of an approximate distribution as follows:

Type	Number
Class 1: Traditional Document Images	9000
Class 2: Camera captured, Text in Scene, and Color documents	500
Class 3: Non-document Images	500

The 9000 traditional documents will be distributed as follows. Please note, the classes are not necessarily disjoint. Some classes may overlap (for example, we have a handwritten memo that appears in two classes).

Genre	Number
Forms, Drawing, Tables	1000
Business Documents, Memos, Letters	2500
Journal and Conference Papers, Articles	2500
Newsletters, Flyers	1000
Structured Documents – phone books, dictionaries	1000
Handwritten	1000
Foreign Language – handwritten and machine printed	1000
Highly Degraded	500
Mixed Annotation	2000

At the beginning of the project, we will define the attributes used to ground truth each document, and the metadata will be provided in XML form so it can be visualized as described below.

2.2.1.2 Synthetic Degradation Tools

A set of degradation tools is currently being added to the core DocLib functionality. In order to provide ground truth data for documents which overlapping regions, care must be taken in ground truthing. These tools allow us to take documents with known zone level ground truth and degrade them with know parameters. The tools have been proven effective in evaluating algorithms on highly degraded data when sufficient real data is unavailable or cannot easily be ground truthed.

2.2.1.3 Ground Truth and Visualization

GEDI is a ground truth editor that gives users the ability to create, modify, and compare metadata on or about document images. A series of scanned image files can be opened simultaneously in a folder, and each document image has an associated xml text file used to store the metadata. In cases where the metadata already exists, the tool can be used for visualization and editing.

The interface is completely configurable, allowing the user to define a set of attributes for each page and a set of "objects" or zones that can appear on the page. Each zone then has a set of attributes, along with a box to define its location in image coordinates. So, the labeling of graphic components, for example, may occur by defining different zone types (logos and stamps) or by defining a single zone type (graphics) and having an associated attribute to distinguish between logos or stamps. In addition to attributes for each zone, the folder as a whole may have attributes such as a writer ID, quality, language, or generation date. These attributes pertain to all the image files in that collection.

Labeling the data on a document is as simple as drawing a box, and can be simplified by setting shortcut keys to tasks which may set an attribute to a certain value or the desired zone type. Functionalities appear within GEDI that support text ground truthing in any language, provide the ability to view a subset of the defined zones, and manually add attributes and new zone types on the fly.

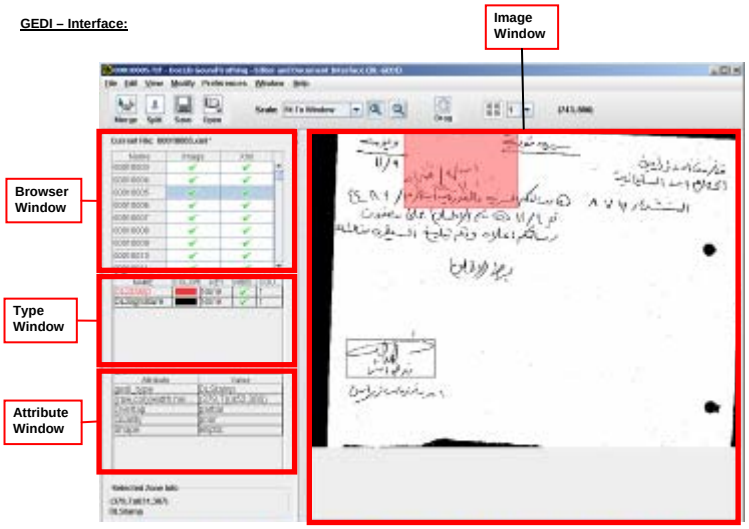


Figure 1: GEDI Interface Markup

A hierarchy between zones is also evident, and visual lines can be seen by toggling the parent/child option. Should the user want to discontinue maintaining a hierarchy, they can toggle a hierarchy option. Hierarchy between two zones can be created or destroyed at any point.

The GEDI tool will ingest an XML provided by standard DOCLIB modules. In this project, we will adhere to the DOCLIB XML representation and provide GEDI as a way to visualize all results. For example, in the evaluation section below, we will describe the output of page segmentation comparison methods that will provide GEDI compatible markup to identify errors in the segmentation results. Figure 1 shows the GEDI interface.

The XML representation is generic and can easily be parsed outside of DOCLIB. Figure 2 shows an example XML File.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
  GEDI is developed at Language and Media Processing Laboratory, University of Maryland.
-->
<GEDI xmlns="http://lamp.cfar.umd.edu/GEDI" version="1.0">
<DL_DOCUMENT src="CYR0540001.tif" NrOfPages="1" docTag="xml">
<DL_PAGE gedi_type="DL_PAGE" src="CYR0540001.tif" pageID="1" width="2496" height="3300"
        Quality="good" Writing="handwritten"
        Gender="female" Primary_Language="cyrillic" Writer_ID="54" Hand="unknown" Source="original"
        Secondary_Language="none" Age="20-50">
 <DL_ZONE gedi_type="DL_TEXTLINEGT" id="<None>" col="296" row="552" width="659" height="415"
        contents="This is a test" offsets="146, 332, 512" segmentation="word" />
<DL_ZONE gedi_type="DLTable" id="<None>" col="1359" row="886" width="352" height="366" />
<DL_ZONE gedi_type="DLStamp" id="<None>" col="464" row="1263" width="215" height="213" />
<DL_ZONE gedi_type="DLPicture" id="<None>" col="1592" row="255" width="378" height="305" />
</DL_PAGE>
</DL_DOCUMENT>
</GEDI>
```

Figure 2: Sample XML Document

2.2.2 Page Classification

Page classification presents a problem not often not fully considered when designing document analysis algorithm, in part, because most research assumes a know group of documents. In cases where one processes large numbers of images of unknown origin and source, however, both text and non-text images will likely comprise the document collection. We will integrate page classification at two levels. The first distinguishes between three classes of documents at the fundamental level: traditional documents, text in images, and images without text. This represents a triage component where different enhancement and processing methods are required for each class. The second provides classification on traditional documents. This will involve building models for various classes of genre, or will have the user provide instances of documents of interest and ranked retrieval.

2.2.2.1 Document Text/Image Text/Non-Text Discrimination

The three classes of images segment the document analysis problem at the highest level. Non-Text images contain no recognizable text. Often images with repetitive horizontal texture fool text detection algorithms. We are not interested in processing these types of images, so this class essentially will be filtered. The second class of image do not conform to the default “pseudo-binary” nature of documents, yet contain text. Typically, these documents contain text embedded scene or graphic imagery, have non-affine distortion such as perspective, or have text written in colorful, display fonts. Specialized techniques are required for image text. The third class contains traditional documents. These are bi-level and have text-like structures (zones, lines, etc), although noise can overwhelm the content.

We propose a promising approach, known as image n-grams, shown to be useful for other classification problems. The approach of using n-grams was initially developed for natural language processing. The basic idea is to view a string as a pattern and represent a document by a series of overlapping n-length sequences. For example, the word DOCUMENT could be represented by the 3-grams, DOC OCU CUM UME MEN ENT. In this way, if we judge similarity between text passages, and errors or inexact matches occur, then considerable overlap still happens. If we match D@CUMENT to DOCUMENT, then only two of the n-grams differ, but four others are identical. More importantly, for a larger text passage, we can develop a statistical representation of the language. These techniques have been applied to retrieval, language ID, and summarization .

The basic concept can easily extend to images with two dimensional n-grams. If we consider representing a document by a series of 3x3 or 5x5 windows, the same principle applies. The goal is to measure the amount of “texture” in the image. In order to save computation, we will explore normalizing the document images.

2.2.2.2 Page Layout Similarity and Genre Classification

This section of the proposal describes extensions to the development of a document ranking system based on layout similarity. Our previous research began to examine three algorithmic solutions for ranking documents. One of the solutions builds upon prior work by one of the current authors. This solution consists of detecting text lines, then considering the quadrilaterals generated by all pairs of lines (as objects describing the page layout). In order to compare page layouts, quadrilaterals from new documents and documents in training sets are compared. Arkin's distance measure was used. This distance was also used for clustering the quadrilaterals, so the subsequent comparisons were applied only to cluster centers in order to increasing computation speed. This approach gives surprisingly good results. By producing N^2 quadrilateral objects from the initial N text line objects, the expanded representation the configuration of every text line is expressed with respect to every other text line by the shape of a quadrilateral without the need for a frame of reference, which may be difficult to find reliably when text scanning is skewed. However, drawbacks include generating more objects than when the process began, and that efficient algorithms for clustering, nearest neighbor, and range search are more difficult to implement with Arkin's distance. Therefore, the goals of the research described in the original paper were to evaluate the performance of Euclidean descriptions of quadrilaterals, and also to discover if more concise layout representations by Euclidean descriptions of single text lines would provide competitive performance in spite of higher sensitivity to document skew and translation. As with our prior work, we quickly discard, without further comparison, potential matches with fonts of very different heights. In one difference, we focus on purely geometric aspects and do not use any text script information in the training and ranking procedures.

The system operates as follows:

1. Find text lines (by grouping connected components, see Figure 3, top) and de-skew the text.
2. Generate quadrilateral objects composed of all pairs of lines (Figure 3, bottom) or lines paired with the top edge of the bounding box.
3. Cluster these objects and find cluster centers.

Then we apply the following steps to rank documents:

1. For each of the documents shown as examples of wanted documents, store objects that are cluster centers into a database with a "wanted" label.
2. For each of the documents shown in a training set of unwanted documents, store objects that are cluster centers into the database with an "unwanted" label.
3. For each document of the set that needs to be ranked, extract its lines and related objects, cluster them, and score each cluster center by looking at its neighbors in the database of wanted objects. Incorporate into the score the presence of neighbors in the database of unwanted objects. Then, obtain a score for the document by combining the scores of each of its objects.
4. Present the documents as a ranked list.

In this project, we will integrate the capabilities to operate on the data collected and provide it as an additional add-on. We will use it to supplement genre classification where users have predefined classes for which they are looking. For evaluation, we will use the collection we gather, use existing documents as queries, and calculate precision and recall.

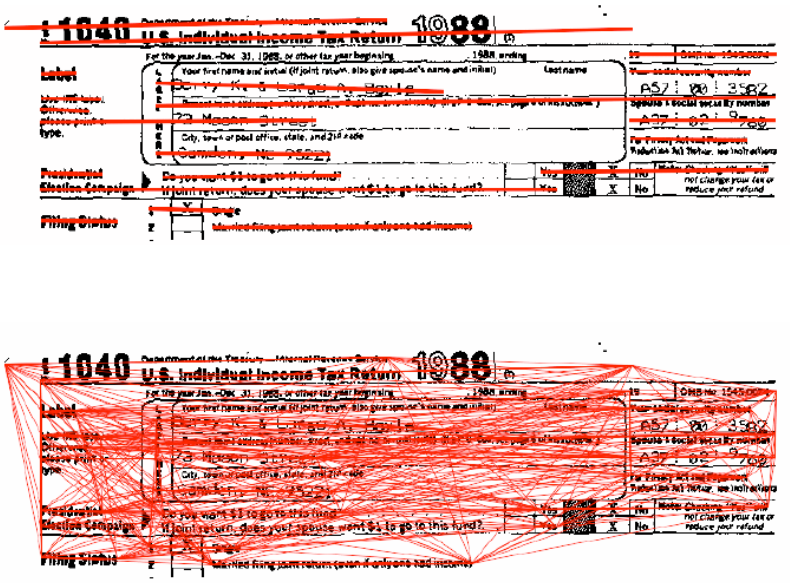


Figure 3: Top: Text lines detected in tax form by grouping connected components of black pixels. Bottom: Set of quadrilaterals formed by considering all pairs of text lines.

2.2.3 Preprocessing and Enhancement

Document images degrade in two ways: 1) physical degradation of the hardcopy documents during creation and/or storage, and 2) degradation introduced by digitization. If severe enough, either can reduce the performance of a document analysis system significantly. Several document degradation models, methods for document quality assessment, and document enhancement algorithms have been presented in previous work. One common enhancement approach is window-based morphological filtering. Morphological filtering performs a look up table procedure to determinate an output of ON (black pixel) or OFF (white pixel) for each entry of the table, based on a windowed observation of its neighbors. These algorithms can be further categorized as manually designed, semi-manually designed, or automatically trained approaches. The kFill algorithm, proposed by O'Gorman, is a manually designed approach and has been used by several other researchers. Experiments show it is effective for removing salt-and-pepper noise. Liang and Haralick proposed a semi-manually designed approach with a 3 x 3 window size. They manually determine some entries to output ON or OFF based on a priori observations. The remaining entries are trained to select the optimal output. It is difficult to design a filter manually with a large window size, and success depends on experience. If both ideal and degraded images are available, optimal filters can be designed by training. After registering the ideal and degraded images at the pixel level, an optimal look-up table, based on observation of the outputs of each specific windowed context, can be designed. However, it is difficult to train, store, and retrieve the look-up table with a large window size. This approach requires both the original and the corresponding degraded images for training. Loce and Dougherty used artificially degraded images generated by models for training, while Kanungo, et al. proposed methods for validation and parameter estimation of degradation models. Though the uniformity and sensitivity of this approach has been tested by other researchers, no degradation model has been declared to pass the validation. Another problem with morphological approaches involves small window sizes. The most commonly used window size appears no larger than 5x5, which is too small to contain enough information for enhancement.

Ideally, image quality should be estimated first so the appropriate enhancement algorithms can be applied automatically. Cannon, et al. proposed a document quality assessment algorithm based on five

factors: small speckle, white speckle, touching characters, broken characters, and font size. They used a linear classifier to select the best four enhancement algorithms and reduced the OCR error rate from 20.27 percent to 12.60 percent on their database. Li and Doermann proposed an approach for quality estimation of color video text that classifies the video text quality into six levels. A majority of the above approaches focus on improving OCR accuracy in noisy documents. As shown in the figure below, degradation will not only deteriorate OCR performance but other document processing tasks, such as page segmentation, as well. Little work has been done in this area. Our approach differs from previous work in that we perform classification to identify noise and exploit contextual information of neighboring blocks as a post processing to refine identification. Experiments show our noise removal algorithm can increase page segmentation accuracy significantly.

Several existing “enhancements” already exist in the toolkit, including skew detection and correction, basic morphology and size normalization. For this project, the primary means of addressing noise will be through trained filters, as described in the layer segmentation process below. In particular, we will focus on scanner and copier noise effecting the edges of the document, on scanner and copier artifacts such as horizontal and vertical dropout or lines, and other clutter in the image.

2.2.4 Layout Analysis

For document layout analysis, salient regions can take the form of text, graphics, or half-tones, and can be nearly any shape or size. However, for the general problem, the decomposition is class-dependent, and unless a specific model is available to guide the analysis, the correct descriptions of the region may not always be obtained at the pixel or component levels. Consider, for example, the problem of table interpretation. A valid decomposition may label a table region appropriately, but, depending on the complexity of the model, a structural analysis may require a more complete description of the column, spacing, and separator components. For this reason, we do not claim the decomposition is complete, but that it divides the document into components that act as a guide to the interpretation process.

A representation is under development that allows the description of document regions according to their physical characteristics (e.g., text, graphics, and half-tones), which can be augmented with appropriate semantic labels.

For general document understanding problems, in which a priori knowledge exists about the contents of the document, the process of decomposition, derivation of document class, and logical component labeling are interdependent. Beginning with a candidate decomposition of the document, as described above, it is possible to establish a hierarchy of abstraction that extends from the physical entities (syntactic components) through the logical entities (semantic labels). In general, this parallels a scene description hierarchy in general computer vision, in which the low-level information is at the pixel level, and the high-level description involves the identification of objects, their components and relationships with other objects. The analysis task derives from a meaningful instantiation of this hierarchy based on information about the document layout and a model space that describes valid structure and logical document organizations.

The structural analysis of documents more specifically involves the derivation of the logical or semantic meaning of a set of salient fields or regions within a document. In general, the problem involves attributes and structural relationships of the document to label document components within the contextual rules dictated by the document class or type (memo, letter, journal article, newspaper, etc.).

Nevertheless, we must start with a basic decomposition. A first approximation to these regions is obtained from a page decomposition module to provide specialized processing for individual components. The goal is to represent the homogeneous regions of the image. Traditionally, this occurred with zones (boxes or polygons) on the page, indicating spatially compact regions such as text – handwritten, machine print, graphics, and image. Each of these zones would then be labeled as to content type, and processed separately.

For the domain on which this project focuses, highly degraded documents no longer satisfy the assumption that white-space or background happens between regions. For such cases, pixel-level representations are most appropriate. We will use a combination of a layered representation (text, graphics, and noise) and subsequent zones within each layer. The final representation will identify machine print, handprint, graphics, and noise layers, with zones around various content blocks in each layer respectively. For the purposes of discussion, we will describe the issues top down. First referring to the problem of page decomposition into layers, followed by zone segmentation. Zone labeling will be described as part of content labeling, in the next section.

2.2.4.1 Page/Layer Decomposition

Some work has been done on handwriting/machine printed text identification. The classification is typically performed at the text line word, or character level. At the line level, machine printed text lines are arranged regularly with a straight baseline, while handwritten text lines are irregular with a varying baseline. Srihari, et al. implemented a text line based approach using this characteristic and achieved a classification accuracy of 95 percent. This approach has the advantage that it can be used in different scripts (Chinese, English, etc.) with little or no modification. Guo and Ma proposed an approach based on the vertical projection profile of the segmented words. They used a Hidden Markov Model (HMM) as the classifier and achieved a classification accuracy of 97.2 percent.

Although, less information is available at the character level, humans can still identify the handwritten and machine printed characters easily, inspiring researchers to pursue classification at the character level. Kuhnke, et al. proposed a neural network based approach with straightness and symmetry as features. Zheng, et al. used run-length histogram features to identify handwritten and printed Chinese characters and achieved promising results. In previous work, we implemented a handwriting identification method based on several categories of features and a trained Fisher classifier. However, the problems introduced by noise were not addressed.

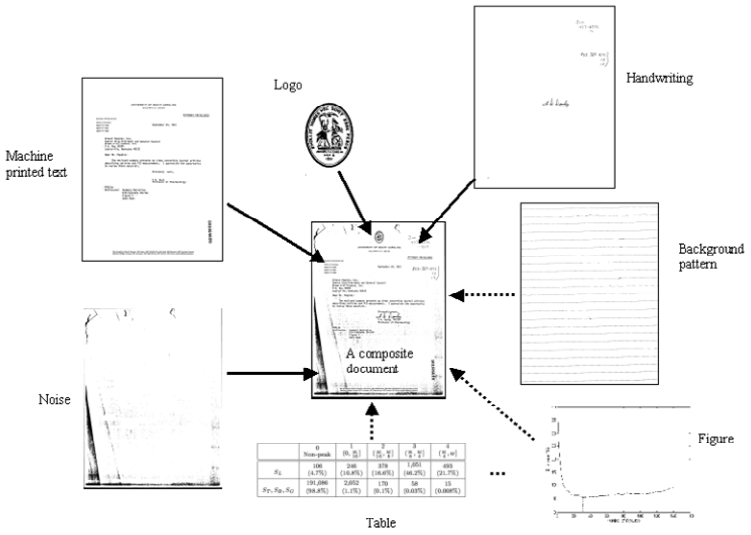


Figure 4: Document Composition

Documents result from a set of physical processes and conditions and consist of layers (letterhead, content, signatures, annotations, noise, etc., in the case of business correspondence). Document analysis reverses these processes to segment a document into layers with different physical and semantic properties. After decades of research, automatic document analysis has advanced to where text segmentation and recognition are a solved problem in clean, well-constrained documents. However, the performance degrades quickly when introduced to a small amount of noise. For example,

a typical bottom-up page segmentation method starts from the extraction of connected components. Based on spatial proximity and size, connected components then merge into text lines and zones. A classification process then identifies zone types (text, tables, images, etc.). These algorithms work well on clean documents where zones with different properties can be easily separated. However, they often fail on noisy documents where noise mixes with and/or is spatially close to content regions. Figures below show segmentation results for an extremely noisy document when we use the Docstrum algorithm and ScanSoft SDK. Text and noise are erroneously segmented into the same zones by both algorithms. In this work, we present a novel approach to identifying text in extremely noisy documents. Instead of simple noise filtering, as used in other work, we treat noise as a distinguished class and model it based on selected features. We further identify handwriting from machine printed text since: 1) handwriting in a document often indcates corrections, additions, or other supplemental information that should be treated differently from the main content, and 2) segmentation and recognition techniques for machine printed text and handwriting differ significantly. Based on these considerations, we treat the problem as a four-class (machine printed text, graphics, handwriting, and noise) identification problem. Previously, work has been done in which graphics remianed in the noise layer. This project seeks to extent previous work to include the processing of graphics.

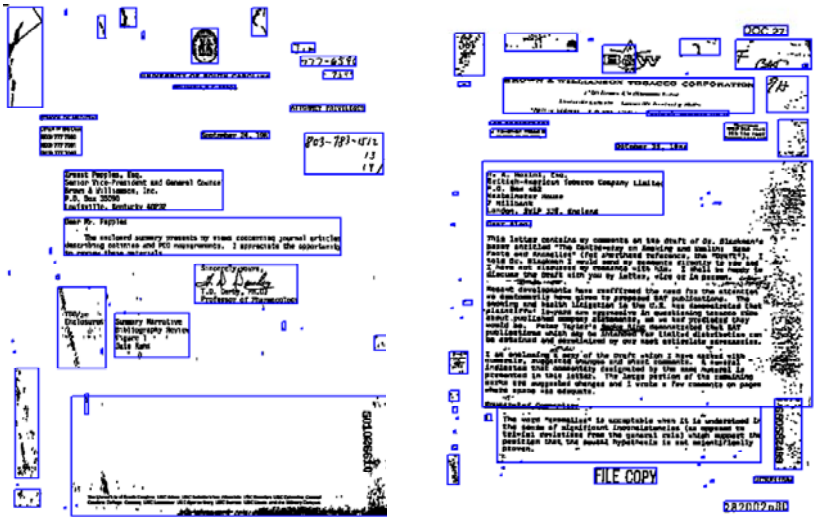


Figure 5: Normal Page Segmentation

In practice, misclassification often happens in an overlapping feature space. (The features we extract are shown below.) This holds especially true for handwriting and noise. To deal with this problem, we exploit contextual information in post-processing and refine the classification. Contextual information is useful for improving classification accuracy. It is widely used in many OCR systems, and its effectiveness has been demonstrated in previous work. The key involves modeling the statistical dependency among neighboring components. The OCR system outputs a text stream that is one-dimensional. Therefore, an N-gram language model, based on an Nth order 1D Markov chain, effectively models the context. With assistance from a dictionary, the N-gram approach can correct most recognition errors. Images, however, are two-dimensional. Generally, 2D signals are not causal, and it is much harder to model the dependency among neighboring components in an image. Among the image models studied so far, Markov Random Fields (MRF) have been widely studied and successfully used in many applications. MRFs suit image analysis because the local statistical dependency of an image can be well-modeled by Markov properties. MRFs can incorporate a priori contextual information or constraints in a quantitative way. The MRF model has been extensively used in various image analysis applications, such as texture synthesis and segmentation, edge detection,

and image restoration. In this work, we use MRFs to model the dependency of segmented neighboring blocks. As postprocessing, MRFs can further improve classification accuracy.

The documents we are processing are extremely noisy, with machine printed text, handwriting, and noise mixed together. We first extract the connected components and merge them at the word level, based on spatial proximity. We then extract several categories of features and use trained Fisher classifiers to classify each word into machine printed text, handwriting, or noise. Finally, contextual information incorporates into MRF models to refine the classification results further.

Feature set	Feature description	# of features	# of features selected
Structural	Region size, connected components	18	9
Gabor filter	Stroke orientation	16	4
Run-length histogram	Stroke length	20	5
Crossing count histogram	Stroke complexity	10	6
Bi-level co-occurrence	Texture	16	2
2×2 gram	Texture	60	5
Total		140	31

Figure 6: Features used and selected for classification

2.2.4.2 Segmentation

We have shown previously that page segmentation on clean layers is often fairly straightforward, and well-known techniques are satisfactory, especially for machine printed documents. The figure below shows two documents, before and after segmentation.

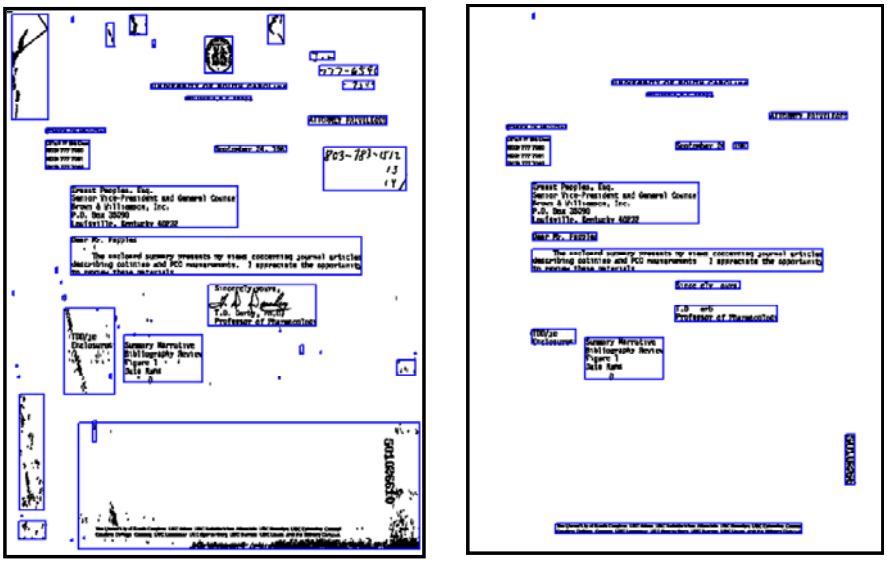


Figure 7: Segmentation results using basic Docstrum, before (left) and after (right) segmentation

It has also been demonstrated that standard commercial off the shelf OCR works better on the clean segmentation. It can be hypothesized that the trained algorithms estimate parameters from the pixels in the zone that they find. When the zones are not homogeneous or they contain non or mixed-text components, the algorithms tend to have trouble, even with clean text.

We will further implement, integrate, and test three segmentation algorithms commonly cited in the literature. Implementations of XY-Cuts, Docstrum, and a Vorinoi based segmentation algorithm will be implemented, integrated, and tested on the machine print, hand print, and graphic layers of the test data set. We view thing as a well-addressed problem, yet providing these tools is a significant components of an end-to-end system.

2.2.4.3 Handprint Line Detection

Although further decomposition of handwritten text content into text lines is not an immediate goal of this project, we will run preliminary experiments to demonstrate the feasibility in anticipation of more in-depth work.

Handwritten text line detection and segmentation remains a significant challenge as a precursor to an off-line handwriting Optical Character Recognition (OCR) system. Instead of improving mature techniques for detecting text lines in machine printed documents, we model text line detection as an image segmentation problem. We first enhance the text line structures by smoothing with a Gaussian window to convert a binary image to gray-scale, then evolve an initial estimate of text line boundaries using the level set method. At the end of boundary selection, a localization module applies to group isolated connected components into nearest text lines to improve accuracy.

We tested our method using multiple scripts, different orientations, and different scales. Experiments show this script-independent method achieves high accuracy (92%) for detecting text lines in both handwritten and machine printed documents.

Our algorithm will investigate a novel perspective for detecting handwritten text lines, and result shows the algorithm as a suitable tool for document image analysis. Some advantages include:

- Our method is more robust compared to a bottom-up connected component based approach.
- The algorithm is script independent.
- The results are non-overlapping regions, which provides a better representation than rectangular bounding box.
- Our method can be used for binary, gray scale, and color document images without major change.

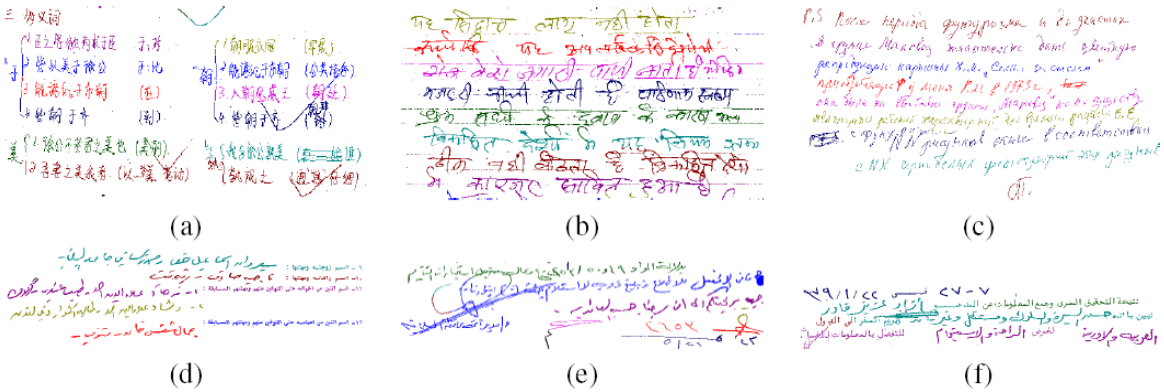


Figure 8: Example Pixel Level results for handwritten text line detection.

2.2.5 Content Labeling

Once a document has been physically segmented into zones, feed forward systems then attempt to classify the zones and assign various semantic labels. In this project, we are concerned primarily with signatures, logos, and stamps for semantic labels, but will also experiment with a general zone labeling technique previously developed in our lab.

2.2.5.1 Zone Labeling

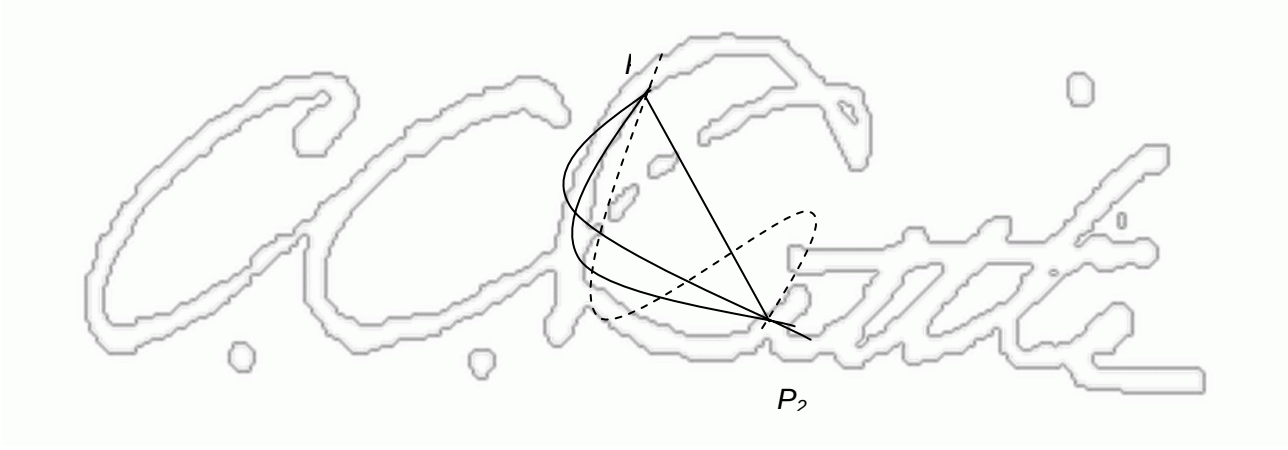
Any generic OCR system performs classification of zones into various syntactic categories such as text, graphics, logo, etc. as an important subtask. Automated techniques for training zone classifiers are crucial because a) the test datasets keep changing, and automated algorithms can be easily adapted to the new datasets by retraining the algorithms, b) the algorithm is not governed by subjective bias of an individual, and c) these generic methods can be employed for any classification problem.

A decision tree based classifier was previously and tested on the University of Washington (UW) dataset. The classifier has a 96% accuracy and approximately 33% fewer misclassification errors than the (UW) algorithm.

- Feature Extraction: Software exists to extract features based on connected components. These features include mean and standard deviation of component height, width, area, and aspect ratio; number of connected components; and percentage of area covered by connected components.
- Classifier: A CART-based decision tree will train on the (UW) dataset.
- Evaluation: The training and testing was done by dividing the dataset into 10 mutually exclusive subsets, training on nine, and testing on one, then rotating the test and training sets.

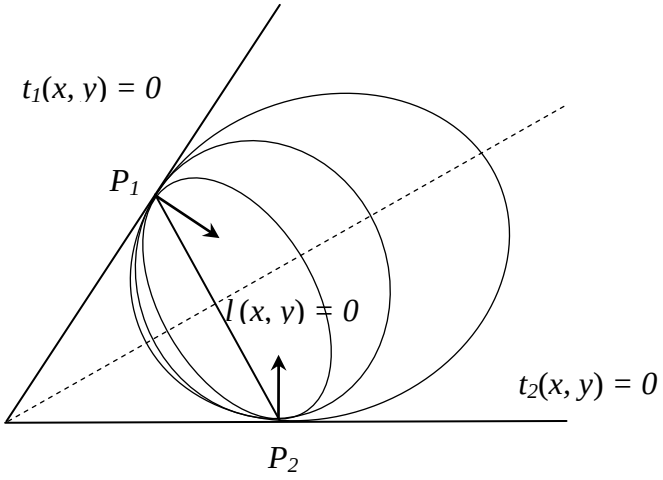
2.2.5.2 Signatures and Annotations

Signature detection and subsequent processing depends highly on cultural styles, document layout, and document type. Although a great deal of work has been done on signature verification, much less work has been done on signature detection. Most work focuses on heuristic rules. We have done some preliminary work on modeling signatures of western writers and using them to detect candidate signature regions. The approach uses Hollerbach's oscillation theory of handwriting. The fundamental approach relies on estimating parameters of this oscillation while maximizing smoothness.



One of the challenges is that given two points *P1* and *P2*, even if we know their gradient directions, infinite possibilities exist for curve fitting. Our idea views a signature as a symbol that possesses

maximally smooth segments, i.e. it should not contain any inflection point inside each segment. We characterize the smoothness of signature curvature using a second-order measure that reflects its physical creation process:



Knowing two points *P1* and *P2* and their gradient directions, we know a family of second-order curves that pass both points

$$f(x,y) \equiv l^2(x,y) - \lambda t_1(x,y) t_2(x,y) = 0$$
$$= ax^2 + 2hxy + by^2 + 2gx + 2fy + c = 0$$

In the Cartesian coordinate system, the graph of a quadratic equation in two variables is always a conic section. For two points on a signature, i.e. for a set of {(*x1*, *y1*), (*x2*, *y2*), (*p1*, *q1*), (*p2*, *q2*)}, the range of λ that corresponds to an ellipse. For the complexity, let *N* be the number of feature points. Running through each pair of points on a connected component of finite bounded length takes O(*NL*).

Since *L* is bounded in practice, our signature detection method runs in linear time O(*N*). Using context further speeds the search.

Figure 9: Top ranked signatures (upper left), bottom ranked signatures (upper right) and false detections (lower)

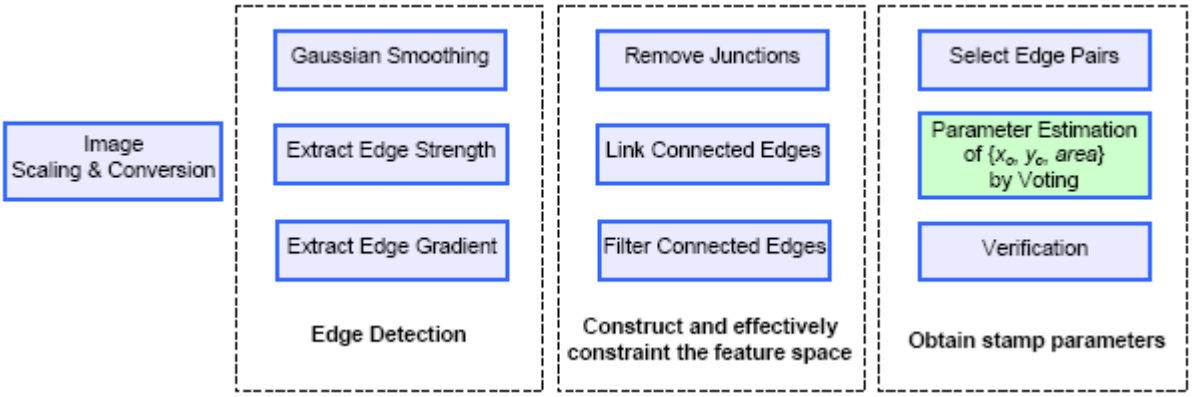
We compare our signature detection results with the two reported offline handwriting recognition systems Guo and Zheng

Detection Approaches	Test Databases	Reported Performance
Guo's HMM based approach	Clean documents scanned at 300-400 DPIs	72.19% recall for fully extracted words, 92.86% recall for partially extracted words, Overall precision of 92.86%.
Zheng's Fisher classifier + MRF approach	Subset of tobacco database	69.9% precision at recall of 93.2% (word level), 83.3% precision at recall of 93.0% (block level).
Our approach	Tobacco database	88.4% precision or 92.7% mean average precision at recall of 81.8%, 90.8% mean average precision at recall of 90.1%.

We will further refine these techniques, test on our new datasets, and integrate with deliverables.

2.2.5.3 Logos and Stamps

Detecting documents with target stamp instances would effectively enable us to retrieve documents associated with a specific source. However, this unique detection problem has essentially remained unaddressed. We will present and integrate a novel logo and stamp detection framework based on detection of analytic shapes through parameter estimation of connected edge patterns. Our approach efficiently exploits orientation information from pairs of connected edge points to determine the center position and area of a stamp region, without computing its entire set of functional parameters. Also, it allows a priori information from available stamp samples to be incorporated effectively. We introduce effective algorithms to address the fact that stamps likely overlay the background content of the document. Experimental results on degraded documents demonstrated the robustness of this retrieval approach on large databases consisting of both printed text and handwritten notes.



2.3 Evaluation

2.3.1 Background

Evaluation is a key component in any research project. It helps to measure the state of the art, identify shortcomings in existing approaches, and measure progress. In this project, we will continue to integrate evaluation capabilities into the toolkit. Although evaluation has been described in various sections throughout this document, it useful to reiterate the general strategy for 1) Segmentation and Layout Analysis, 2) Content Labeling, and 3) Enhancement.

Ground truth will be provided at the zone level of major text and graphics regions in the page, represented in the XML format shown above. Similarly, our modules will produce a similar result set in XML. The evaluation of segmentation will use traditional overlap measures to judge the correspondence between ground truth and result zones. Although the initial layered representation will be presented at the pixel level (visualized with false color), the evaluation will occur at the zone level.

For content detection and labeling, standard detection metrics will be used to identify correct, missed, and false detections in the analysis. These, too, will occur at the zone level.

Finally for enhancement, we will provide synthetically degraded documents, from which the original layers are known a priori. Evaluation will be performed in a purposive manner by measuring the effects of enhancement on downstream processes, such as zone labeling.

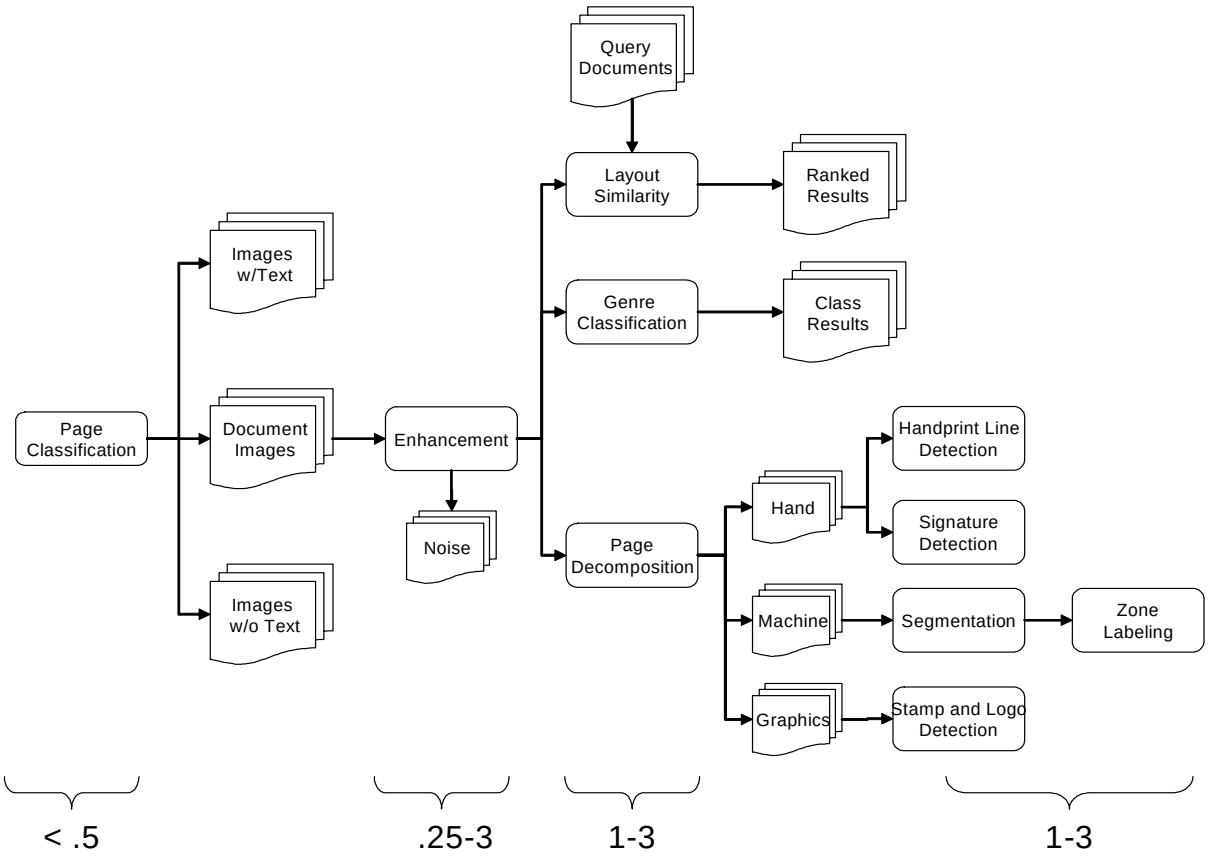
2.3.2 Performance

The projects goal aims to provide an end-to-end toolkit for “CLassification, Enhancement, and Analysis of Heterogeneous Document Image Collections” Since a great deal of this proposal focuses on basic research, it is difficult to estimate performance of various modules. Nevertheless, we will

focus on, but cannot guarantee, to meet or exceed the following performance levels on the CLEAT dataset.

Task	Performance Goal
Page Classification	80% precision across all three classes
Enhancement	10-30% increase in accuracy of downstream processes – segmentation, detection
Layer Separation	90% coverage at the pixel level
Segmentation (Print and Hand)	85% using implementation of existing methods
Logo and Stamp Detection	75% precision at 85% recall
Signature Detection	75% precision at 85% recall

Finally, all our algorithms will solve the specifically defined problems, but care will be taken to ensure they can be efficiently integrated into enterprise workflow solutions. When systems must process tens of thousands of pages per day, systems that take 30, 20, or even 10 seconds per page may be unacceptable. Thus far, most algorithms have been implemented so they can run under 10 seconds per page. Many basic processing, such as segmentation, are near 1 second per page. For each algorithm, we will carefully consider the cost of implementation and strive to minimize the processing time required, with a goal of under 10 seconds per page. Below is a flow chart indicating target processing speeds for various stages of processing.



Target Processing Speed in Seconds

3 Project Management Approach

The project will be conducted out in the Laboratory for Language and Media Processing at the University of Maryland. The laboratory employs undergraduate and graduate students, faculty research assistants, and post-doctoral research assistants, in addition to its director and clerical staff.

The students will work under the direction of Dr. David Doermann. Students will be assigned work on one of the key problems and will research the background and previous work. Individual meetings will occur between the supervisor and the students at least weekly to review progress and define direction. Often, direction occurs on a daily basis. The laboratory meets as a whole on a weekly basis, and students are expected to present research results in the form of an hour-long technical discussion at least once a semester.

4 Statement of Work

4.1 Tasks

Task 1: Data Collection

The project will produce a dataset that represents, as closely as possible, the most significant challenges for the sponsors, including different genre, composition, levels of noise, and content. A total 10000 documents will be collected and provide for use exclusively in the project. The collection will occur in stages, to allow for feedback from the sponsors and adequate time to adapt the collection in an attempt to make it as representative as possible.

Task 2: Ground Truthing

The project seeks to provide ground truth on a subset (approximately 1000 pages) of test data. Handwritten and machine printed regions, graphics regions – including stamps, and logos -- and other content regions, including signatures, will be marked with a defined XML format. All ground truthing will happen at the zone level. Zones will include, for example, homogeneous, continuous regions of hand printed and machine printed text; compact graphics regions such as logos, stamps, or figures; and other document “figures” such as drawings, tables, and graphs. In cases where zones contain mixed text and graphics, they will be treated as figures with a single surrounding box.

Task 3: Evaluation Framework

The researchers will work closely with the sponsors to refine existing metrics for page enhancement, segmentation, classification, and content labeling. Tools will input ground truth and results and output labeled XML data showing correct, missed, and falsely detected regions. In order to provide evaluation of overlaid and degraded documents at the pixel level, degradation models will provide synthetic (but representative) images for which we have pixel accurate ground truth.

Task 4: Evaluation and Visualization Tool

The project will enhance the existing GEDI tool to provide visualization of results and specialized ground truthing capabilities. Both polygon and bounding box ground truthing capabilities will be provided, as well as the ability to visualize multiple layers at the pixel level using false color.

Task 5: Page Classification Module

The project will develop a page classification module, which will distinguish between images containing document text, image text, and no text. A prototype page layout similarity module will be

provided, and results will be run on the CLEAT dataset. Experiments will demonstrate the feasibility of page similarity for genre classification.

Task 6: Enhancement Module

The project will develop a module that provides enhancement at both the pixel level and macro component level. The algorithms will be evaluated directly on the CLEAT dataset, and purposively by evaluating the improvement enhancement provides on downstream modules. To avoid costly ground truthing of documents at the pixel level, the CLEAT dataset will be augmented with synthetic degradations, where the parameters are known and evaluation of enhancement can be automated. Purposive evaluation will also occur down stream tasks.

Task 7: Layout Analysis Module

The task provide will research and development algorithms for page separation into machine print, handprint, graphics, and noise layers. Each layer then will be analyzed and segmentation results provided for various components as described in Task 8. For the text layers, text zones will be labeled, along with signature regions. For graphics regions, stamps and logos will be identified.

Task 8: Content Labeling module

The project will provide a content labeling module that will work with layout analysis to label signatures and logos and stamp, in the handwritten and graphic layers, respectively. The project will rely on the specifics of the data set collected in Task 1 for evaluation.

Task 9: Evaluation

The project will perform periodic evaluations for internal development purposes. An intermediate and final formal evaluation will result in evaluation reports for the sponsors. Segmentation and layout analysis will be initially evaluated at the zone level and enhancement at the pixel level, using synthetically degraded data.

Task 10: Training

Training will be provided to developers using the system and its tools. Interaction of approximately three-five days is anticipated, either at the sponsors facilities or the University of Maryland. Additional remote support will be provided for the software through the project.

4.2 Milestones

Phase 1 - March 31, 2007

- Deliver completed CLEAT data collection.
- Provide ground truth for subset of data including signatures, stamps, logos, handwritten, and machine printed text.
- Provide document describing evaluation framework.

Phase 2: June 30, 2007

- Deliver completed ground truthing and visualization tool for CLEAT metadata.
- Deliver Prototype version of CLEAT Software API Modules:
    - Document Image Enhancement,
    - Document Text/Image Text/Non-Text Discrimination,
    - Page Layout Similarity Ranking on CLEAT data,
    - Page Layer Segmentation and Zone Labeling, and
    - Content Labeling of Signatures, annotations, Stamps and Logos.

- Provide results of CLEAT API run on CLEAT datasets.
- Provide preliminary evaluation report.
- Provide basic API documentation

Phase 3: September 30, 2007

- Deliver Final version of CLEAT API.
- Provide training on use of CLEAT.
- Provide complete evaluation results on CLEAT data.
- Provide complete documentation of API.
- Provide feasibility report for system extensions.
- Provide a list of publications generated and planned as a result of this effort.

4.3 Summary of Deliverables

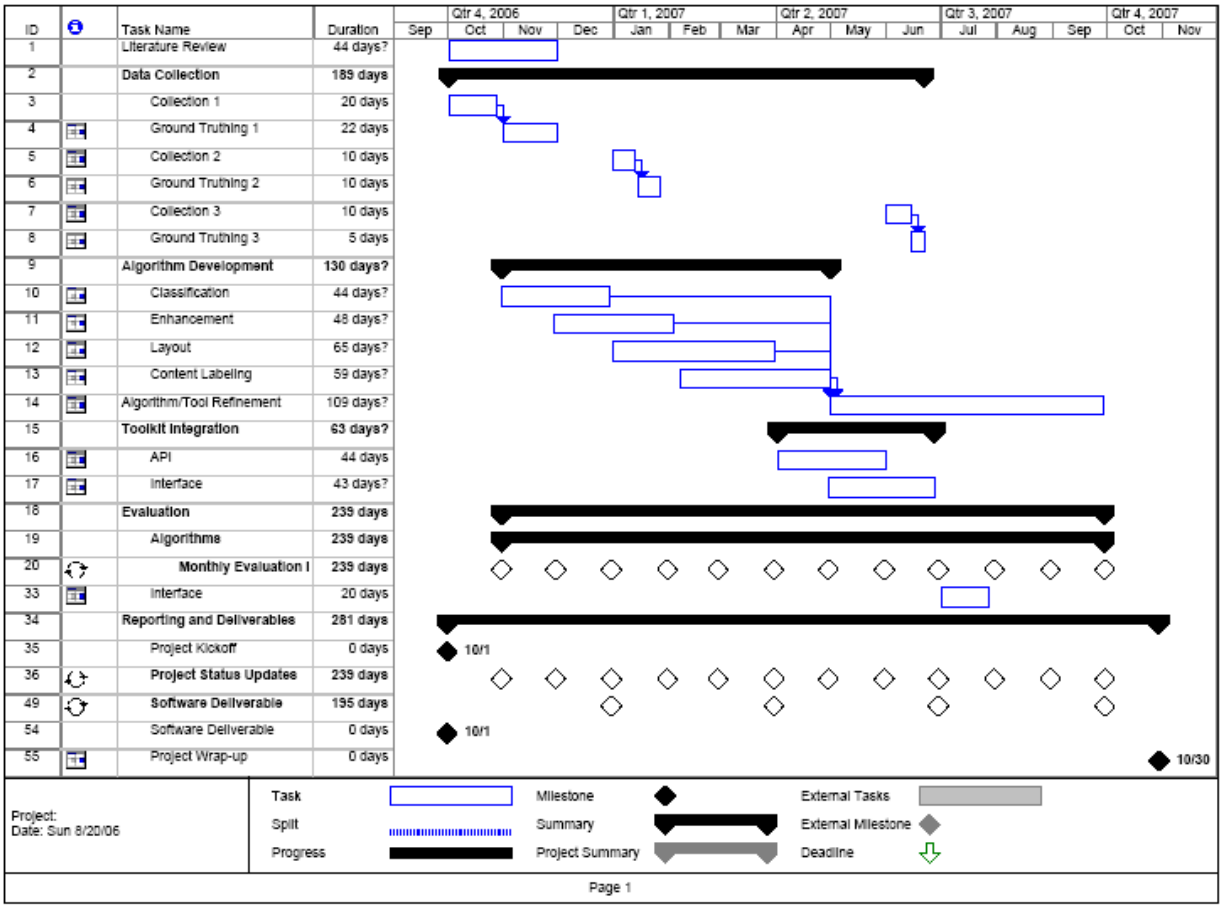
The project will produce deliverables in the form of technical documentation, technical articles and publications, and software. They will occur in accordance with the project schedule above. The deliverables will include:

1. **Reporting** – A quarterly report will highlight accomplishments, plans, and issues related to this project. We will discuss new developments and update the sponsors with information on data collection, algorithm development, and evaluation. The report will be delivered in electronic form in an agreed upon format. A final report will be provided summarizing the status of each component.
2. **Publications -** Algorithms and system design may be published in the form of technical reports, conference, or journal papers. The sponsor will be cited and be given a copy of each report related to this project.
3. **Datasets and Ground Truth –** A dataset containing examples of each class of document we process will be included. The dataset will contain a minimum of 5000 documents and be collected from a variety of sources, including the internet, existing training and testing datasets, public collections, project collections, and scanning. All ground truth will be provided in GEDI format and accompany the images
4. **Software API –** The DCAT software API will appear on DVD, along with associated training and testing samples. The software will be supported on Windows and Linux platforms.
5. **Evaluation Software –** Basic software will evaluate the results of processing. All output will be in a GEDI compatible representation that can be used with provided visualization tools.
6. **Graphical User interface –** A graphical user interface will be provided to visualize the results of processing and annotate new ground datasets. The interface will be implemented in Java and be configurable. The API will have an output method so internal data structures representing zone information can be output and visualized immediately.
7. **Software Documentation –** Full documentation will be provided for all software. Doxygen will be used to document the API, and a users guide will be provided with examples for the user interfaces.
8. **System Installation and Training –** All software and documentation will be provided at various stages throughout the project and one month prior to the project end. To the extent possible, Internet 2 will be utilized to hold training sessions. Some funds have been allocated for a trip to the sponsors.
9. **Final Feasibility report –** A report will identify key technical problems with enhancement and segmentation, either discovered during this project or were outside the scope of the initial project.

**** All software will be supported throughout the project, and for a period of 6 months beyond the end of the project, free of any licensing or maintenance fees**

4.4 Project Schedule

The project schedule will be updated to reflect progress on each of the specified tasks. The project is currently scheduled to begin on October 1, 2006, but will be adjusted as necessary.



The project will also include two face-to-face meetings. One will occur in late March or April 2007 when the sponsors will travel to Maryland and one to coincide with the end of the project to provide final deliverables in November 2007, in Singapore.

Detailed and Itemized Pricing

This budget covers a 12 month period. All costs are in US Dollars. The Graduate Effort is based on 20 hours/week during the academic year, and 40 hours/week during the summer. Undergraduates are paid hourly on a time worked basis.

Principal Investigator - Doermann 20% effort	26,848
2 Graduate Students 12 months	55,804
2 Undergraduates (1250 hours Total)	16,800
Total Salaries	99,452
Fringe Benefits (~25%)	23,728
Total Personnel	123,180
Foreign Travel	10,000
Computing Materials and Supplies	1,500
Total Direct Costs	134,680
Facilities and Administration 48.5 % (**)	65,320
Total	**200,000**

** Facilities and Administration is the University's negotiated overhead rate on direct costs.

5 Appendix: Project Team Staffing

Dr. David Doermann received a B.Sc. degree in Computer Science and Mathematics from Bloomsburg University in 1987, and a MSc. degree in 1989 in the Department of Computer Science at the University of Maryland, College Park. He continued his studies in the Computer Vision Laboratory, where he earned a Ph.D. in 1993. Since 1993, he has served as co-director of the Laboratory for Language and Media Processing in the University of Maryland's Institute for Advanced Computer Studies and as an adjunct member of the graduate faculty. His team of 15-20 researchers focuses on topics related to document image analysis and multimedia information processing. Recent intelligent document image analysis projects include page decomposition, structural analysis and classification, page segmentation, logo recognition, document image compression, duplicate document image detection, image based retrieval, character recognition, generation of synthetic OCR data, and signature verification. In video processing, projects have centered on the segmentation of compressed domain video sequences, structural representation and classification of video, detection of reformatted video sequences, and the performance evaluation of automated video analysis algorithms. He is the co-editor of the International Journal on Document Analysis and Recognition. He has over 25 journal publications and almost 100 refereed conference papers.

Selected Journal Publications:

- Yefeng Zheng and David Doermann. Robust Point Matching for Nonrigid Shapes By Preserving Local Neighborhood Structures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), pages 643-649, April 2006.
- Jian Liang, Daniel DeMenthon and David Doermann. Mosaicing of Camera-captured Documents Without Pose Restriction. Computer Vision and Image Understanding, 2006. (SUBMITTED).
- Yefeng Zheng, Huiping Li and David Doermann. A Parallel-Line Detection Algorithm Based on HMM Decoding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5), pages 777-792, May 2005.
- Jian Liang, David Doermann and Huiping Li. Camera-Based Analysis of Text and Documents: A Survey. International Journal on Document Analysis and Recognition, 7(2+3), pages 83-104, July 2005.
- Huanfeng Ma and David Doermann. Adaptive Hindi OCR Using Generalized Hausdorff Image Comparison. ACM Transactions on Asian Language Information Processing, 26(2), pages 198-213, 2004.
- Yefeng Zheng, Huiping Li and David Doermann. Machine Printed Text and Handwriting Identification in Noisy Document Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(3), pages 337-353, March 2004.
- Huanfeng Ma, Burcu Karagol-Ayan, David Doermann, Doug Oard and Jianqiang Wang. Parsing and Tagging of Bilingual Dictionaries. TAL Traitement Automatique Des Langues, 44(2), pages 125-150, 2003.
- Katheryn Guo, David Doermann and Azriel Rosenfeld. Forgery Detection by Local Correspondence. International Journal on Pattern Recognition and Artificial Intelligence, 15(4), pages 579-641, 2001.
- Christian. Shin, David Doermann and Azriel Rosenfeld. Classification of Document Pages Using Structure-Based Features. International Journal on Document Analysis and Recognition, 3(4), pages 232-247, 2001.
- Huiping Li, David Doermann and Omar Kia. Automatic Text Detection and Tracking in Digital Video. IEEE Transactions on Image Processing - Special Issue on Image and Video Processing for Digital Libraries, 9(1), pages 147-156, January 2000.

- David Doermann, Ehud Rivlin and Azriel Rosenfeld. The Function of Documents. International Journal of Computer Vision, 16, pages 799-814, 1998.
- David Doermann. The Indexing and Retrieval of Document Images: A Survey. Computer Vision and Image Understanding, 70(3), pages 287-298, 1998.
- Kamran. Etemad, David Doermann and Rama Chellappa. Multiscale Document Page Segmentation Using Soft Decision Integration. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 92-96, 1997.
- David Doermann, Ehud Rivlin and Issac Weiss. Applying Algebraic and Differential Invariants for Logo Recognition. Machine Vision and Applications, 9(2), pages 73-86, 1996.

Yi Li (3rd year Graduate Student) - Yi is focusing on research related to handwriting analysis and on handwriting/machine print discrimination**.**

Mudit Agrawal (3rd year Graduate Student) – Mudit has focused on retargetable OCR research and bootstrap training methods for document analysis.

Michael Roth (Senior Undergraduate Student)

6 Appendix: The Organization

6.1 Overview

The University of Maryland, College Park is the flagship of the University of Maryland System. It is located just north of the nation's capital, in Maryland. The University holds top rankings in many disciplines and is among the top research universities in the country. Within the College of Mathematical and Physical Sciences, the University of Maryland Institute for Advanced Computer Studies (UMIACS) is a central point for research programs covering a broad range of areas, addressing both fundamental core computer science issues and fundamental problems at the interfaces between computer science and other disciplines.

The infrastructure provided by UMIACS is geared primarily toward supporting interdisciplinary research, while the core computer science projects are conducted primarily through the Department of Computer Science. Current interdisciplinary projects involve faculty from a number of disciplines and are typically conducted through one of the UMIACS laboratories. The project will housed with Dr. David Doermann in the Laboratory for Language and Media Processing.

The Laboratory for Language and Media Processing (LAMP) houses researchers focusing on providing tools and techniques for access to large heterogeneous databases of multimedia information objects. There are many environments in which large static and dynamic collections of documents, images, and video are being gathered or created, yet these sources remain inaccessible without techniques to index and retrieve the information automatically. As technology moves us toward simplified creation and use of multimedia documents, we will see an even greater increase in the need to transmit, browse, or otherwise process these collections efficiently.

Our recent efforts in document and video analysis have allowed us to develop an environment in which to design and test new algorithms. We have developed a number of prototype systems ranging from analysis of handwriting to compression to recognition of logos. As a natural extension of previous research, we are enhancing this environment to treat higher-level problems. The primary theme of the research is to provide automatic access to information sources by addressing issues involved in initial processing, organization, manipulation, and retrieval.

The LAMP lab has a variety of equipment, from state of the art PCs to camera networks, scanners, and digitization hardware. The lab has dedicated computing facilities tailored to research needs and organized into clusters of systems that share common resources, including mass storage, application installations, and high speed network connectivity.

6.2 Contacts

Contracting	
Name	Jeff Richardson
Address	3112 Lee Building University of Maryland, College Park, MD 20742-5141
Telephone	301 405 6178
Fax	
Email	jeffr@umd.edu
Administrative	
Name	Johanna Weinstein
Address	2127 A.V. Williams Building University of Maryland, College Park, MD 20742-3251
Telephone	301 405 6728
Fax	
Email	jow@umd.edu
Technical	
Name	David Doermann
Address	3451 A.V. Williams Building University of Maryland, College Park, MD 20742-3251
Telephone	301-405-1767
Fax	443-638-0236
Email	doermann@umd.edu