1-1-2007

Unsupervised learning of document image types

Dean Patrick Curtis
University of Nevada, Las Vegas

UNSUPERVISED LEARNING OF DOCUMENT IMAGE TYPES

by

Dean Patrick Curtis

Bachelor of Science
University of Nevada, Las Vegas
2005

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2007

UMI Number: 1452238

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 1452238

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval

The Graduate College
University of Nevada, Las Vegas

OCTOBER 22, 2007

The Thesis prepared by

DEAN PATRICK CURTIS

Entitled

UNSUPERVISED LEARNING OF DOCUMENT IMAGE TYPES

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

John T. Minor

Examination Committee Member

Ajoy Kumar Datta

Examination Committee Member

Graduate College Faculty Representative

1017-53

ABSTRACT

Unsupervised Learning of Document Image Types

by

Dean Patrick Curtis

Evangelos Yfantis, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

In a system where medical paper document images have been converted to a digital format by a scanning operation, understanding the document types that exists in this system could provide for vital data indexing and retrieval. In a system where millions of document images have been scanned, it is infeasible to expect a supervised based algorithm or a tedious (human based) effort to discover the document types. The most sensible and practical way to do that is an unsupervised algorithm. Many clustering techniques have been developed for unsupervised classification. Many rely on all data being presented at once, the number of clusters to be known, or both. Presented in this thesis is a clustering scheme that is a two-threshold based technique relying on a hierarchical decomposition of the features. On a subset of document images, it discovers document types at an acceptable level and confidently classifies unknown document images.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Evangelos Yfantis. He has an incredible commitment to the advancement of his students and to the advancement of science. His focus on the harmony of hard work and innovation has instilled in me the desire to always move forward and think outside the box. He has not only contributed to the construction of this thesis, but has contributed tremendously to my growth as a student.

I thank the members of my committee; Dr. Ajoy Datta, Dr. John Minor and Dr. Angel Muleshkov. I thank them for taking time out of their very busy schedules to be members of my committee.

I also thank the many employees I worked with in the Digital Image Processing Lab; Chris Jones, John Bunch, Jia Tse, Vitaliy Kubushyn, Michael Rogers, Aaron Thomas, Scott Miller, and Jae Adams. In addition to the employees in the lab, I also thank Chris Weiss, Nick Cerjanic and Herb Elliot; employees at Apogen technologies.

Finally, I would like to thank my Dad. He has helped me by providing me with so many opportunities and opening so many doors. Without him, I would have not have had such an enriching and successful academic experience. He is a strong committed father and a great friend.

CHAPTER 1

INTRODUCTION

Overview of Document Image Analysis

In document image processing, vast and numerous algorithms exist which provide solutions to many of the problems posed by document analysis. Many papers have been written, and many theses have been done on DIA. Many of the sources of research gathered have been derived from the IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI). In 2000, Nagy [53] published a paper that compiles ninety-nine articles relevant to the field of DIA and reports on its evolution in the previous twenty years.

Numerous well known applications and algorithms have been devised for the enhancement, structural analysis, and classification of document image features. Nagy et al. [52] describe the impact of the growth of the Internet and its relation to digital character recognition, most significantly to the need for archival and retrieval of technical material, as well as the generation of HTML code from DIA output. General research done in this area includes [12], [43]. Other areas of research include the works of Saund, Fleet, et al. who have developed algorithms for the acquisition and interpretation of information from informal and casual document images [58], [59]. Ha provides a comprehensive description of techniques that can be employed for all phases of a document analysis system [36]. What follows are algorithms that are closely related to the work of this thesis, which have set precedents and share new ideas on DIA.

Adamek et al. [1] created an algorithm which recognizes characters based on holistic word recognition in which scalar and profile-based features are taken from the entire word image. The contour of a word is utilized to follow this approach. Extraction is performed by the following: binarization, localization of lower case letters, connected components labeling, connecting disconnected letters, and contour tracing. Most significantly, the algorithm uses a multiscale convexity/concavity representation in the process of contour tracing that stores information about the convexity and concavity at different scale levels for each contour point, stored in a 2D matrix. The algorithm is capable of word recognition without breaking words into smaller segments.

Agarwal et al. [3] have provided an application for segmentation and classification based on document structure through the automated analysis of bank checks. Recognition of the courtesy amount follows a six step model: input image handler, segmentation, segmentation critic, preprocessing, neural network recognizer, and post-processing. Strings are created based on the proximity and alignment of characters. Then, the correct string is chosen based on a set of rules, one of which is the currency sign. Another system with the complete capabilities of extracting features is done by Adams in [2].

Diana et al. [24] devised a method for document analysis based on three different modules. The first, low-level, processing is comprised of the following three stages: acquisition, binarization, and skew detection. The second, document structuration, processes the image to extract features into a tree structure for organizational purposes. The last module, form class identification, uses a process of graph matching to compare the tree of one form to that of another in a list of previously extracted forms. Through the coordination of these modules, the document can be properly modeled and classified.

Hobby and Ho [38] created a preprocessing method of document enhancement by clustering character images. Image clusters of single symbols are used to compute the average outline from matching bitmaps, replacing all occurrences of the symbol in order to reduce the overall noise degradation of the document.

Jain and Yu [41] describe a method for the storage of a paper document as an electronic version. Important to the process are various techniques for finding the structural and lexical layout. The authors use a bottom-up approach based on the connected component extraction to segment regions in a document. Additionally they propose a top-down model which can represent a document for editing, storage, retrieval, and analysis.

O'Gorman [57] describes an algorithm for processing document images based on layout analysis. The document spectrum, based on bottom-up analysis, uses a nearest neighbor clustering method which measures skew, line spacing, and text blocks. It is independent of skew angle, and text spacing, and it is capable of processing different text orientations in the same image.

Xi and Lee [71] determined an algorithm which extracts table structures from skewed document images through the use of gradient and wavelet analysis. Gradient calculations are used first to process the document image and, subsequently, the vertical and horizontal lines are obtained through the wavelet decomposition. The structure of the form is obtained through the use of a modified wavelet reconstruction algorithm. Finally, through Minkowski Subtraction, the table structure image and the deskewed image can be used to create the table free image as well as a table structure image.

Introduction to Issues in Clustering

Cluster analysis is a type of classification in which the structure of data is determined with only the observed elements being available, whereas the type of classification called discriminant analysis is when groupings of some observations are used to categorize others and infer the structure of the data [26]. For example, discriminant analysis would be used for optical character recognition (OCR) where characters or digits are used to train a statistical classifier, and this training data is used to categorize (recognize) an observation.

Clustering is a technique that provides for unsupervised classification. Clustering has applications in fields such as the life sciences, medical sciences and engineering [5]. There are varying types of clustering algorithms, such as agglomerative clustering [28], K-means, fuzzy [31], hierarchal and sequential [5,33,66]. Other algorithms developed include [3,4,14,16,17,47] including an entropy-like k-means algorithm [65].

Clustering is a technique used in unsupervised learning. Unsupervised learning is a classification where the class labeling is not available [66]. The concern becomes to reveal the organization of patterns into sensible clusters (groups), which will allow one to discover similarities and differences among patterns and to derive useful conclusions about them [66]. Unsupervised learning has applications in fields such as life sciences, medical sciences, social sciences, earth sciences, and engineering [66].

Fraley [27] describes cluster analysis as the the automated search for groups of related observations in a data set and the identification of groups of observations that are cohesive and separated from other groups. Cluster analysis gained popularity recently due to quickly advancing technologies that have fueled the rise of several prominent areas of application. They include the following:

- Data Mining - which began as a search for customer and product groupings in large retail datasets.
- Document Clustering and Indexing - where large sets of web-based and image-based documents are indexed and sorted.
- Gene expression - which arises from the desire to find genes that act together.
- Image Analysis - where cluster analysis is used for image segmentation and quantization [27].

In general, there are five steps to a clustering algorithm, as stated by Theodoridis [66]. These five steps are listed below as follows:

- Feature Selection - features must be created that can effectively describe as much information concerning the task with minimum information redundancy. These features are often encoded and represented as vectors $\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^l$
- Proximity Measure - These are measures that quantify how similar or dissimilar two feature vectors are.

- Clustering Criterion - This is the expert's decision as to what type of clusters will underlie the data set. This can be expressed as a cost function or a set of rules.

- Clustering Algorithm - The algorithm chosen that forms the clusters using the proximity measure and the criterion.

- Cluster Validation - This is a process of ensuring that the algorithm has established a satisfactory clustering. Techniques include manual validation or any number of automatic tests.

Many clustering algorithms require a specified (fixed) number of clusters to be defined, but, in dynamic information systems such as the work done in document classification, the number of document types (clusters) is not known *a priori* . In [7], an algorithm is presented for an online clustering in a dynamic environment.

A classification problem can be on either of the two extremes that one may face. The first is the complete statistical knowledge of the underlying joint distribution of the observation $\mathbf{X}$ and the classes Ω [19].

Banerjee et al. [9] proposed a class of distortion functions that admit an iterative relocation scheme (such as in k-means) where a global objective function based on distortion with respect to cluster centroids is progressively decreased. He proposed and analyzed parametric hard and soft clustering algorithms based on Bregman divergences. Nock [56] proposed a method based on the constrained minimization of a Bregman divergence using a method called boosting and weighting.

Clustering images is an integral part of DIA and computer vision. Various papers describe viable methods for the problems and solutions to clustering. Agarwal et al. [3] describe the problem of clustering in domains where affinity relations between cluster elements are of a higher order than two. The algorithm first constructs a weighted graph and approximates a hypergraph. A clustering algorithm based on a normalized Laplacian is used to partition the vertices. Then, based on the hypergraph approximation, weights are assigned to the edges. Liu et al. [47] describe an algorithm for creating distributed spill trees which can be used for online searches for nearest neighboring points in high dimensional spaces, enabling it to perform clustering on a set of more than a billion images. The algorithm does not depend on object types but only requires feature vectors in a metric space. Haim [37] describes a content-based approach for web image searching.

Sheikholeslami [60] proposed a method of clustering using the multi-resolution properties of wavelets transforms. Using wavelets allows for effecient clustering, the detection of clusters of arbitrary shapes, insensitive to outliers and the order of the input of data.

A type of clustering referred to as spectral clustering performs clustering using the eigenstructure of certain data. Bach [8] used the eigenstructure of a similarity matrix using a cost function with a technique called spectral relaxation. Dhillon [23] provides a connection between kernel k-means and spectral clustering using a weighted kernel k-means objective function with normalized cuts. Littau [45] uses a technique referred to as PDDP for clustering very large data sets.

Burl [14] has shown an application of feature extraction and classification in response to the remote exploration of the solar system and the vast archive of images that followed. The algorithm devised for mining useful information from these images involves various components, the first of which is the focus of attention(FOA) which takes, as input, the images and outputs a list of candidate object locations. The FOA can quickly exclude areas that obviously have no relevance to the search parameters. Subsequently, feature vectors are extracted from the FOA which are then integrated into a neural network that classifies features based on both positive and negative training examples.

Cheng et al. [16,17] have proposed an approach to document segmentation which uses both local texture characteristics and image structure in order to segment documents. The method is based on a multiscale Bayesian probabilistic function which allows modeling of image and structural characteristics. The local texture characteristics are extracted at each resolution via wavelet decomposition. The document is segmented using a fine-to-coarse-to-fine procedure.

In [47], a large scale nearest neighbor algorithm was developed for cluster images on the order of a billion, where the features used were extracted directly from images. Their algorithm is a parallel version of the spill tree algorithm [46]. Additional works in large scale clustering algorithm development are given in [18, 22, 51].

Clustering in Document Image Databases

Document type classification can allow for indexing and document understanding, and facilitate the creation of efficient document navigation systems. Work has been done in document image databases in discovering duplicates [25, 49, 50], and implementing techniques that are useful to document image type discovery.

This algorithm helps in the prediction of an unknown document that needs to be processed or recognized. By searching on the m clusters instead of the N total documents in the system where $m \ll N$, efficient association of the document can be achieved. By associating this unknown document with a cluster, we can assume already extracted information about that document such as the location of various fields (social security number, date, name, etc).

It is important to define what is meant by "documents". Much work has been done in indexing of documents when documents refers to web pages [15]. This research associated documents with the physical paper document. Much research has been accomplished in the field of indexing paper documents based on text extracted using OCR methods [10, 15, 33, 63]. Document retrieval is often the limitation of these OCR based systems. In many applications, it is desirable to have a system that contains robust classifying schemes that capture document relations and structure. In order to incorporate this property, a system must be developed that can create a classification scheme in which the structure and data are permanently embedded within the document feature representation.

Hull and Cullen [39] developed an algorithm for determining the similarity and equivalence of document features through visual means. Pass codes are used as feature vectors on a document by document basis and used to locate documents that contain similarities to the input image. This was determined by the Euclidean distance to the arrangement of pass codes in subsections of each image. A method performing recognition using visual similarity is also presented in [61].

Kenairi et al. [42] described a system which identifies different types of forms, using a statistical approach, without points of reference. Automatic form segmentation was performed to extract the structure of the document and designate it as the main block set. Next, blocks are matched within each class, thereby calculating block attributes. Subsequently, the blocks are identified by calculating the Mahalanobis distance and a weighted statistical distance between them, either accepting or rejecting the results based on whether a minimal distance is achieved and it falls below a threshold.

The structure or layout of the document holds much information that can be used for segmentation and classification. Analysis of the document structure is necessary to understand the type of document which is presented, whether it be a historical document, scientific paper, or free flowing text. Antonacopoulos and Downton [6] provide an overview of weaknesses exposed in the analysis of the structure of historical documents, and new methods to overcome them. Fujihara and Babiker [29] created an algorithm for classifying technical documents based on single generic models as well. The model is based on a point-interval representation which retains the attributes of the block regions. Liu-Gong et al. [48] have developed a method for converting a document image into its layout structure through the use of an analysis system and several models. The layout structure is generic in that it is composed of generic objects and can be used as a rule base. Provided with various parameters, a general model is capable of recognizing different types of documents. The general model is represented by a hierarchical tree and composed of several class-objects. Class-objects contain only attributes which describe the characteristics of layout objects and are used for segmentation. The recognition of the document is achieved by a model that contains the document's information and the recognition method, allowing the analysis system to be independent on the document.

Various algorithms have been developed for hierarchically segmenting a document image. Bitlis et al. [11] have written an algorithm to describe and compare the content and layout of a document, given its image, storing the results in a hierarchical tree for classification. Nakajima et al. [54] dealt with segmenting machine-printed documents recursively, in a process described as the Split Detection Method. Through the use of field separators, lines, edges, and background separation, a rule base on periodicity of occurrence of the listed features is formed. After detection, the segments are then stored in a tree structure, in which all nodes are traversed in accordance with a rule base through the process of reading sequence analysis, allowing for the meaningful interpretation of the results.

Sivaramakrishnan et al. [62] described an algorithm for determination of the zone type given the coordinates of the left most-top and rightmost-bottom points, and the document image. Statistical pattern recognition is used to classify each zone on the basis of its feature vector which consists of all these properties as fields is formed for each zone. Additionally, in the context of zoning, Taghva et al. [63, 64] address retrieval effectiveness and ranking variability when automatic zoning is applied to a document. The paper determines a linear relationship between the rankings of manual zoning and automatic zoning, determining them to be statistically equivalent processes. A collection of 1055 documents were used and ranked according to the measures of recall and precision. The corresponding rank of each document was found in the manual version and represented as a point, which yielded a scatter plot from which a least squares fit was determined and a regression line found. The difference between average precision for the two runs is too small to be considered statistically significant. Equivalently, the difference between automatic zoning and manual zoning is statistically insignificant.

Methods utilizing document concepts are described in [32]. [21] performs a document concept based approach to organizing business letters into similar concepts using document structures.

CHAPTER 2

PROJECT DESCRIPTION

The algorithm developed performs a hierarchical classification using a decomposition of the features. Bitlis et al. [11] describes an algorithm using a hierarchical technique. A tree structure is created to represent a document image and document similarities are established based on the trees created from the document images. Other examples of hierarchical based techniques include [20, 30, 34, 35, 44, 55, 69, 73]. The algorithm presented in this thesis produces a clustering of images, but can also be used as an estimate of the number of clusters that exists. A work by Tibshirani [67] introduces a method for estimating the number of clusters using a statistic he developed.

The classification algorithm is an unconstrained sequential clustering based scheme in which (1) the number of clusters is unknown, (2) the number of samples to be classified is unknown and (3) no *a priori* knowledge is presented. This algorithm is useful for problems in which it is not feasible for the entire data set to reside in memory and the supervised training of the entire set cannot practically be accomplished with a human effort. Another important issue with clustering is the curse of dimensionality [5]. This algorithm inherently implements a form of dimensionality reduction. The hierarchical algorithm we have developed incorporates the idea of a two-threshold algorithm presented in [66]. The algorithm is divided into two main

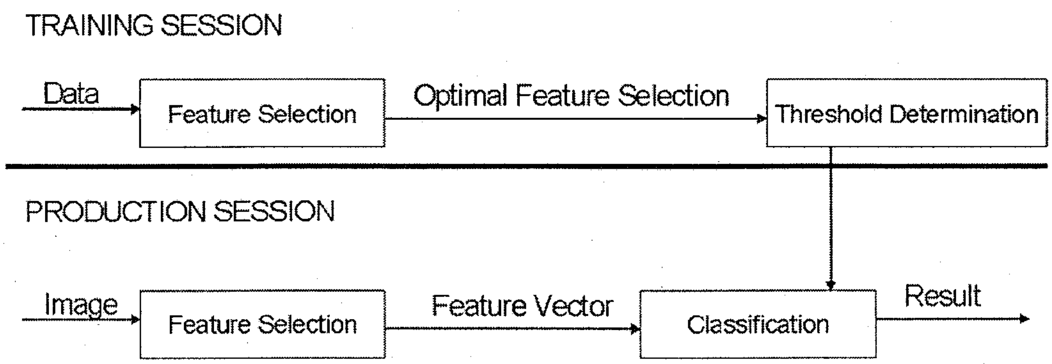


Figure 1: The two sessions for unsupervised classification of document images

sessions: the training session and the production session (Figure 1).

The training session involves two main stages. The first stage is a feature selection step. Fourteen features are able to be extracted from the document image and an algorithm is developed that creates a series of configurations, whereby each configuration maximizes a criterion. A supervised classification was performed using conditional probabilities and the criterion is the classification accuracy of a configuration.

The second stage of the training session establishes a lower/upper bound threshold pair for each feature configuration established in the first stage. A subset of document images are divided into their respective types and statistical analysis of within-class and between-class measures are used to establish the lower/upper bound threshold pair.

The production session involves two stages and expects the training session to be completed. The first stage of the production session is the feature extraction stage. This is when a sample image (raw form) is presented to the system and the image is translated from raw data to a feature vector.

The second stage of the production session is the classification (clustering) phase. The clustering algorithm uses a time series hierarchical approach where, for a sample document image, classes (document types) are eliminated as a potential match for that document image. Elimination at each time step is based on the upper/lower bound threshold pair for that configuration. The classification of an image to a cluster or the creation of a new cluster is performed based on some termination condition.

This thesis is organized by the following chapters. In Chapter 3, a description is given of the features extracted from a document image and how they are encapsulated into a vector format. Chapter 4 discusses the steps taken to develop the classifier. Section 4.1 discusses the algorithm for the construction of the feature configurations. Section 4.2 shows how the thresholds for the lower and upper bounds are determined for each configuration constructed. Section 4.3 provides the hierarchical feature decomposition classification algorithm. In Chapter 5, results of the system are reported and then conclusions and discussions are provided.

CHAPTER 3

DOCUMENT FEATURE EXTRACTION

The features used to perform document classification are based primarily on the structural nature of the form. The focus in this thesis is on the structural features.

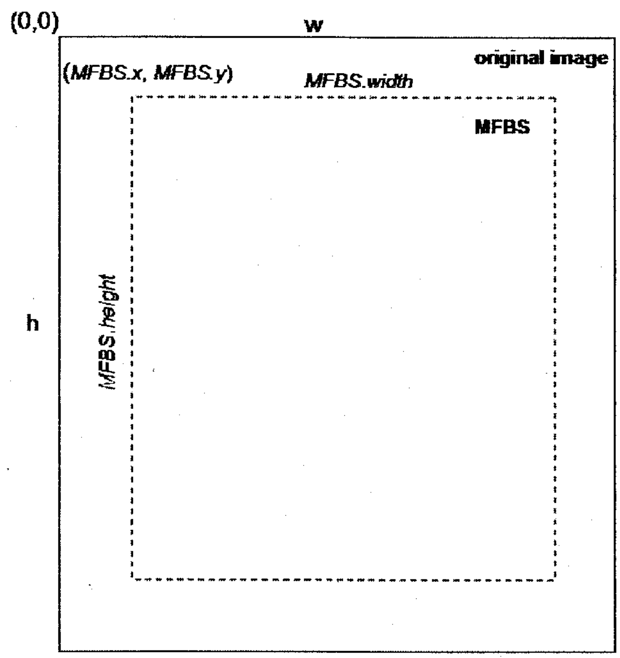


Figure 2: The Major Form Body Segment (**MFBS**) of a document image (bounding box that surrounding the actual content of an image)

The Major Form Body Segment (**MFBS**) is the content of interest for the document image. Extraction of the content of interest requires the removal of margins and some positional adjustments. In [13], the algorithm we developed for **MFBS** extraction is described. This feature is represented by the 4-tuple $\{x, y, width, height\} \in$

MFBS.

The structural features, ψ, extracted are structural line segments, checkboxes and typewritten words (location and OCR result).

The two types of lines that are extracted are horizontal (ψ_{hl}) and vertical (ψ_{vl}) line segments. The set ψ_{hl} contains N_{hl} line segments where $\psi_{hl}^{(i)}, 0 \leq i \leq N_{hl}$ is the i^{th} horizontal line segment. The set ψ_{vl} contains N_{vl} lines where $\psi_{vl}^{(i)}, 0 \leq i \leq N_{vl}$ is the i^{th} vertical line segment. Each line segment, whether it is horizontal or vertical, is described by six parameters,

$$\psi_{hl}^{(i)}.minX, \psi_{hl}^{(i)}.minY,$$
$$\psi_{hl}^{(i)}.maxX, \psi_{hl}^{(i)}.maxY,$$
$$\psi_{hl}^{(i)}.centerX, \psi_{hl}^{(i)}.centerY$$

where

$$< \psi_{hl}^{(i)}.minX, \psi_{hl}^{(i)}.minY > \longrightarrow \textbf{startingpoint}$$
$$< \psi_{hl}^{(i)}.maxX, \psi_{hl}^{(i)}.maxY > \longrightarrow \textbf{endingpoint}$$

describes the starting and ending points for each line segment and

$$\psi_{hl}^{(i)}.centerX = \frac{\psi_{hl}^{(i)}.maxX + \psi_{hl}^{(i)}.minX}{2} \tag{3.1}$$
$$\psi_{hl}^{(i)}.centerY = \frac{\psi_{hl}^{(i)}.maxY + \psi_{hl}^{(i)}.minY}{2} \tag{3.2}$$

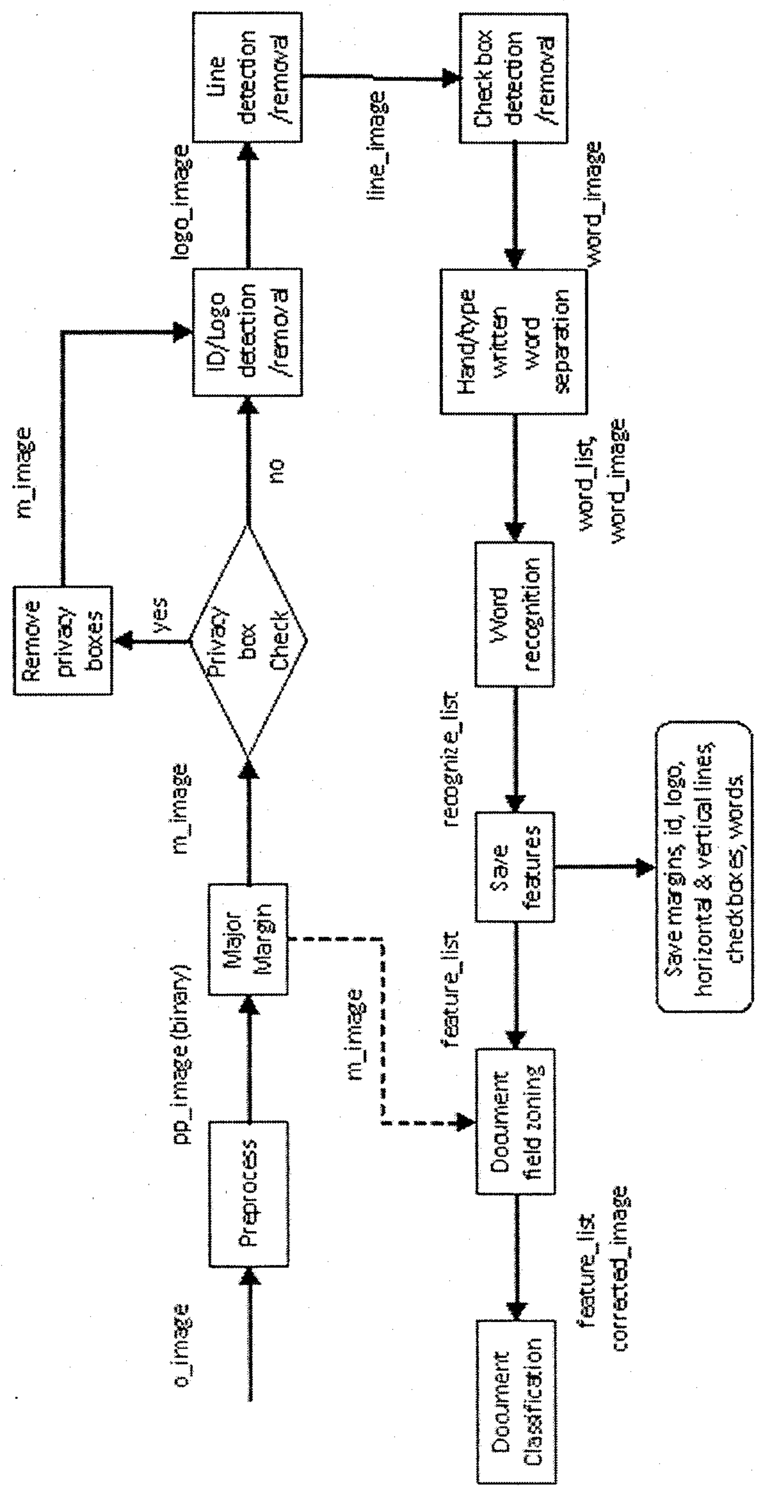


Figure 3: The steps taken in extracting the features from a document image

Processes

Preprocessing	Major Margin	ID/Logo detection and removal	Line detection and removal
•Remove large black areas •Remove noise •Deskew image •Binaryize image •Save (po_image)	•Find the major margins of the image •Extract the image bounded by the margins •Save (margins_info)	•Find the Id and logo of image •Remove Id and logo from image •Save (id_rect) •Save (logo_rect)	•Find the vertical and horizontal lines of image •Remove lines from image •Save (line_info)

Check box detection and removal	Handwritten / typewritten word separation	Word recognition	Document field zoning
•Find the checkboxes of image •Remove boxes from image •Save (box_info)	•Find the word island of image •Separates typewritten word from handwritten word •Perform word segmentation •Save (word_list)	•Character Segmentation •Character recognition •Save (recognize_list)	•Detect, zone and recognize social security numbers

Definitions

- •o_image: Original image (gray/rgb)
- •pp_image: preprocessed image
- •m_image: image enclosing only majaor margins
- •logo_image: image with id & logo removed
- •line_image: image with lines removed
- •word_image: image consisting of only words
- •word_list: A list of rectangles whose coordinates denote the location of the word in the image
- •po_image: grayscale version of pp_image
- •margins_info: coordinates of the margin location
- •Id_rect / logo_rect: rectangle coordinates of the id / logo location
- •line_info: List of horizontal & vertical line locations
- •box_info: List of rectangles of box locations
- •word_list: List of rectangles of typewritten word locations
- •recognize_list: List of strings representing word_list

Figure 4: The details of each of the steps taken in the feature extraction process shown in figure 3

The method for extracting the line segments is a gradient-wavelet based approach. The parameters for ψ_{hl} and ψ_{vl} are relative to the **MFBS** shown in Figure 2 and normalized between 0 and 1. Where the absolute position of a horizontal line segment, $\psi_{hl}^{(i)}$, in the original image is given by

$$minX = (\psi_{hl}^{(i)}.minX * MFBS.width) + MFBS.x$$
$$minY = (\psi_{hl}^{(i)}.minY * MFBS.height) + MFBS.y$$

for the starting point and

$$maxX = (\psi_{hl}^{(i)}.maxX * MFBS.width) + MFBS.x$$
$$maxY = (\psi_{hl}^{(i)}.maxY * MFBS.height) + MFBS.y$$

for the ending point. The same is true for ψ_{vl}.

Checkboxes are indicated by ψ_{cb} where $\psi_{cb}^{(i)}, 0 \leq i \leq N_{cb}$ is the i^{th} checkbox out of N_{cb} checkboxes. Checkboxes are described using rectangles, so a checkbox, $\psi_{cb}^{(i)}$ has the parameters

$$\psi_{cb}^{(i)}.x, \psi_{cb}^{(i)}.y,$$
$$\psi_{cb}^{(i)}.width, \psi_{cb}^{(i)}.height,$$
$$\psi_{cb}^{(i)}.centerX, \psi_{cb}^{(i)}.centerY$$

where the center point of the rectangle is $(\psi_{cb}^{(i)}.centerX, \psi_{cb}^{(i)}.centerY)$ and

$$\psi_{cb}^{(i)}.centerX = \frac{\psi_{cb}^{(i)}.maxX + \psi_{cb}^{(i)}.minX}{2} \tag{3.3}$$

$$\psi_{cb}^{(i)}.centerY = \frac{\psi_{cb}^{(i)}.maxY + \psi_{cb}^{(i)}.minY}{2} \tag{3.4}$$

A template search based algorithm was developed for checkbox extraction in [40]. The parameters for the checkboxes are stored relative to the **MFBS** . The normalized values of a checkbox, $\psi_{cb}^{(i)}$, are related to the absolute position in the original by

$$\begin{aligned} x &= (\psi_{cb}.x * MFBS.width) + MFBS.x \\ y &= (\psi_{cb}.y * MFBS.height) + MFBS.y \end{aligned}$$

for the (x, y) coordinate and

$$\begin{aligned} width &= \psi_{cb}.width * MFBS.width \\ height &= \psi_{cb}.height * MFBS.height \end{aligned}$$

for the width and height of the checkbox rectangle.

Once the typewritten words have been separated from the handwritten words [68], then both the OCR result of the word and the location (rectangle) of the typewritten words, ψ_w, is extracted. The i^{th} typewritten word, $\psi_w^{(i)}, 0 \leq i \leq N_w$, where

N_w is the number of typewritten words, has the parameters

$$\psi_w^{(i)}.x, \psi_w^{(i)}.y,$$

$$\psi_w^{(i)}.width, \psi_w^{(i)}.height,$$

$$\psi_w^{(i)}.centerX, \psi_w^{(i)}.centerY,$$

$$\psi_w^{(i)}.word$$

where $(\psi_w^{(i)}.centerX, \psi_w^{(i)}.centerY)$ represents the center point of the rectangle and is computed in the same way as in Equations (3.3) and (3.4).

Structural Feature Encapsulation

Using the features, $\{\psi_{hl}, \psi_{vl}, \psi_{cb}, \psi_w\}$, extracted from a document image, p, the structural vector, $\mathrm{s}^{(p)}$, is constructed. $\mathrm{s}^{(p)}$ is the composition of n feature vectors from the set of vector representations, V, where $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$, and $\mathrm{s}^{(p)} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ where $1 \leq n \leq M$ and M is the maximum number of vectors ($M = 14$). Each vector, $\mathbf{v}_i$, is constructed based on a meaningful representation of the features extracted, $\{\psi_{hl}, \psi_{vl}, \psi_{cb}, \psi_w\}$. Each vector, $\mathbf{v}_i$ is represented by its symbol (shown in figure 1). Following is the process for constructing each vector, $\mathbf{v}_i$, $1 \leq i \leq M$.

The simplest features to construct are the count based features, **hlc**, **vlc**, **cbc** and

Feature	size	symbol
Horizontal Line Count	1	**hlc**
Vertical Line Count	1	**vlc**
Checkbox Count	1	**cbc**
Typewritten Word Count	1	**twc**
Checkbox Grid	16	**cbg**
Checkbox Relational Grid	16	**cbrg**
Horizontal Word Profile	25	**hwp**
Horizontal Island Profile	50	**hip**
Horizontal Line Grid	100	**hlg**
Horizontal Line Profile	50	**hlp**
Typewritten Word Grid	100	**twg**
Typewritten Word Relational Grid	100	**twrg**
Vertical Line Grid	100	**vlg**
Vertical Line Profile	25	**vlp**

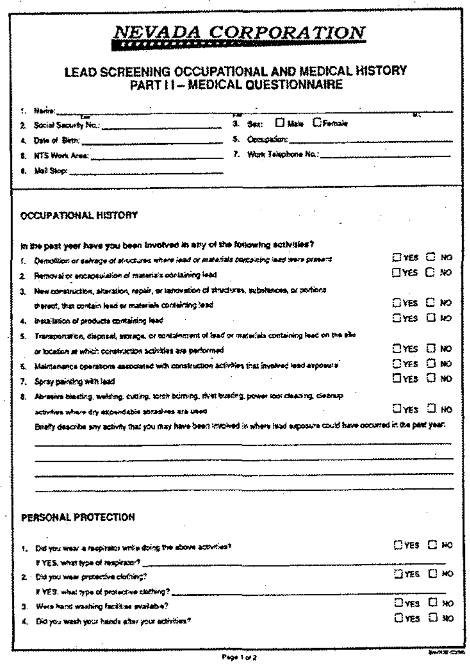
Table 1: List of the vector set V used in the construction of the structural vector $\mathbf{s}^{(p)}$ for image p

twc. They are computed as follows

$$\begin{aligned}\mathbf{hlc}_0 &= |\psi_{hl}| \\ \mathbf{vlc}_0 &= |\psi_{vl}| \\ \mathbf{cbc}_0 &= |\psi_{cb}| \\ \mathbf{twc}_0 &= |\psi_{w}|\end{aligned}$$

where the notation $|\bullet|$ refers to the cardinality of the set.

The features **hlg**, **vlg**, **cbg** and **twg** are based on an Image Grid Decomposition (IGD) of the **MFBS** . Figure 5 shows how an image is overlayed by a grid of dimension n x n. This method of translation is similar to that stated in the *Triangle Proportionality Theorem* . Given the images, p and q of the same type that differ by



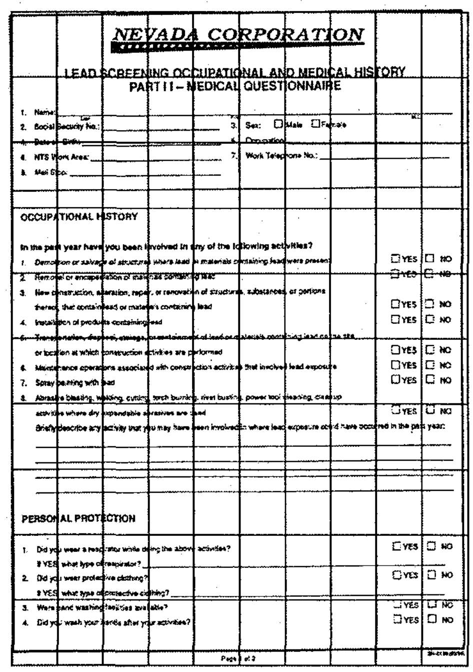
NEVADA CORPORATION

LEAD SCREENING OCCUPATIONAL AND MEDICAL HISTORY
PART II – MEDICAL QUESTIONNAIRE

OCCUPATIONAL HISTORY

PERSONAL PROTECTION

(a) Original Image



NEVADA CORPORATION

LEAD SCREENING OCCUPATIONAL AND MEDICAL HISTORY
PART II – MEDICAL QUESTIONNAIRE

OCCUPATIONAL HISTORY

PERSONAL PROTECTION

(b) IGD

Figure 5: (a) Overlaying of the original image (b) with an $n\ x\ n$ grid (IGD)

a scale, γ, the features of p, $\psi^{(p)}$ and the features of q, $\psi^{(q)}$ will be related by

$$\psi^{(q)} \equiv \gamma \psi^{(p)}$$

By using the IGD of an image, the grid based features will become equivalent

$$\mathbf{hlg}^{(p)} \equiv \mathbf{hlg}^{(q)}$$

for an image p and q of the same type, independent of γ.

The transition function for horizontal line segments, f_{hlg}, creates a vector, **hlg** where $\mathbf{hlg}_i^{(p)}$ is the count of the number of horizontal line segments passing through element i of the p^{th} image. Given a grid dimension with r rows and c columns, the element at (x, y) in the IGD corresponds to the i^{th} element in $\mathbf{hlg}^{(p)}$ by

$$i = (y * columns) + x \tag{3.5}$$

For example, if the count of the horizontal line segments passing through grid element $(5, 4)$ is 10 and the grid is 10 x 10. Then, $i = 45$ and $\mathbf{hlg}_i^{(p)} = 10$. The transition function, $f_{\mathbf{hlg}}$, for some image, p, is shown in algorithm 1. It is important to note that for some horizontal line segment j, the following condition holds

$$\psi_{hl}^{(p,i)}.minY = \psi_{hl}^{(p,i)}.maxY$$

Algorithm 1: Translation for **hlg**

Input: $\psi_{hl}^{(p)}$
Output: **hlg**
for $i = 0$ TO N_{hl} **do**
  $start = (\psi_{hl}^{(p,i)}.minY * columns) + \psi_{hl}^{(p,i)}.minX$
  $\mathbf{hlg}_{start}^{(p)} = \mathbf{hlg}_{start}^{(p)} + 1$
  $end = (\psi_{hl}^{(p,i)}.maxY * columns) + \psi_{hl}^{(p,i)}.maxX$
  $\mathbf{hlg}_{end}^{(p)} = \mathbf{hlg}_{end}^{(p)} + 1$
  for $j = start$ TO end **do**
    $\mathbf{hlg}_j^{(p)} = \mathbf{hlg}_j^{(p)} + 1$
  end
end

The transition function for vertical line segments, $f_{\mathbf{vlg}}$, is very similar to the $f_{\mathbf{hlg}}$.

The following condition holds

$$\psi_{vl}^{(p,i)}.minX = \psi_{vl}^{(p,i)}.maxX$$

and the algorithm is shown as Algorithm 2. The incrementing step from *start* to *end* must take place vertically. In $f_{\mathbf{vlg}}$, since the feature is horizontal, incrementing i goes from left-to-right. For $f_{\mathbf{vlg}}$, going from *start* to *end* involves changing i to be the next element vertically by

$$i = (y * columns) + \psi_{hl}^{(p,i)}.minX$$

where $\psi_{vl}^{(p,i)}.minY < y < \psi_{vl}^{(p,i)}.maxY$.

Algorithm 2: Translation for **vlg**

Input: $\psi_{vl}^{(p)}$
Output: **vlg**

for $i = 0$ TO N_{vl} **do**
  $s = (\psi_{vl}^{(p,i)}.minY * columns) + \psi_{vl}^{(p,i)}.minX$
  $\mathbf{vlg}_s^{(p)} = \mathbf{vlg}_s^{(p)} + 1$
  $e = (\psi_{vl}^{(p,i)}.maxY * columns) + \psi_{vl}^{(p,i)}.maxX$
  $\mathbf{vlg}_e^{(p,end)} = \mathbf{vlg}_e^{(p,end)} + 1$
  for $y = \psi_{vl}^{(p,i)}.minY$ TO $\psi_{vl}^{(p,i)}.maxY$ **do**
    $i = (y * columns) + \psi_{vl}^{(p,i)}.minX$
    $\mathbf{vlg}_i^{(p,i)} = \mathbf{vlg}_i^{(p)} + 1$
  end
end

The transition functions for typewritten word locations, $f_{\mathbf{twg}}$ and $f_{\mathbf{twrg}}$, are closely related. $f_{\mathbf{twg}}$ forms an IGD on the **MFBS** and $f_{\mathbf{twrg}}$ forms an IGD relative to only typewritten words (as supposed to the **MFBS**).

The translation function, $f_{\mathbf{twg}}$, uses a similar process as $f_{\mathbf{hlg}}$ and $f_{\mathbf{vlg}}$. Equation (3.5) is used to convert from the two-dimensional grid to the one dimensional vector, **twg**. The element $\mathbf{twg}_i^{(p)}$ represents the number of typewritten words rectangle centers contained in element i of the p^{th} image. Equations (3.3) and (3.4) show the computation for the center point of a rectangle, where $(\psi_w^{(p,i)}.centerX, \psi_w^{(p,i)}.centerY)$ is the center point of the i^{th} typewritten word rectangle. The algorithm for computing f_{twg} with an IGD of $n \times n$ is shown as Algorithm 3.

Algorithm 3: Translation for **twg**

Input: $\psi_w^{(p)}$
Output: **twg**
for $\underline{i = 0 \text{ TO } N_w}$ **do**
    $i = (\psi_w^{(p,i)}.centerY * n) + \psi_w^{(p,i)}.centerX$
    $\mathbf{twg}_i^{(p)} = \mathbf{twg}_i^{(p)} + 1$
end

$f_{\mathbf{twrg}}$ forms an IGD relative to only typewritten words (as opposed to the **MFBS**), meaning that bounds for the grid are based on the spatial proximity amongst typewritten words. So, the first step is to establish a bounding rectangle over all typewritten words by finding the values, $\{x_1, y_1, x_2, y_2\}$. x_1 is the minimum x coordinate and y_1 is the minimum y coordinate and (x_1, y_1) is the upper left corner. x_2 is the maximum x coordinate and y_2 is the maximum y coordinate and (x_2, y_2) is the bottom right corner. Then, the bounding rectangle is divided into an $n \times n$ grid.

The second step is to decide which grid element each typewritten words belongs to. This transition is similar to the transition for $f_{\mathbf{twg}}$. The algorithm for computing

$f_{\mathbf{twrg}}$ is shown as Algorithm 4.

Algorithm 4: Translation for **twrg**

Input: $\psi_w^{(p)}$

Output: **twrg**

$scaleX = \frac{1}{x_2 - x_1}$

$scaleY = \frac{1}{y_2 - y_1}$

for $i = 0$ TO N_w **do**

  $x' = (\psi_w^{(p,i)}.centerX - x_1) * scaleX * n$

  $y' = (\psi_w^{(p,i)}.centerY - y_1) * scaleY * n$

  $i = (y' * n) + x'$

  $\mathbf{twrg}_i^{(p)} = \mathbf{twrg}_i^{(p)} + 1$

end

The translation function for checkboxes, $f_{\mathbf{cbg}}$ and $f_{\mathbf{cbrg}}$, performs the same operations on the checkbox rectangles to construct the vector $\mathbf{cbg}^{(p)}$ and $\mathbf{cbrg}^{(p)}$.

The functions for **hip**, **hlp** and **vlp** construct a projection of the feature described. The vector, **hip**, is a projection of the islands along the vertical orientation of the image. **hlp** and **vlp** are the projections of the features ψ_{hl} and ψ_{vl} respectively.

Before the algorithm for constructing **hip** is discussed, the definition of an island must be provided. Islands are groups of word-location pairs that are related by a set of rules. Given the set of all islands, T, each island, $\tau_j \in T, i = 1, 2, \ldots, M$, contains a set of word-location pairs where $\tau_j \subset P$, and

$$\{\forall_{j,l \in T}\ \tau_j, \tau_l \in T \ : \ \tau_j \cap \tau_l \equiv \emptyset\}$$

τ_j^l is the l^{th} element of the j^{th} island. Each $p_i \in \tau_j, i = 1, 2, \ldots, N_{\tau_j}$ are related their absolute proximity to each other. First, they are related by a vertical alignment function, $vrtalign(p_{i-1}, p_i)$ by

$$\{\forall_{i \in \tau_j} p_i \in \tau_j : \; i \geq 1 \wedge vrtalign(\tau_j^m, p_i)\} \tag{3.6}$$

where $vrtalign(p_{i-1}, p_i)$ satisfies all four of the following conditions

1. $p_i.minY \leq p_{i-1}.centerY \wedge$
2. $p_{i-1}.centerY \leq (p_i.maxY) \wedge$
3. $p_{i-1}.minY \leq p_i.center \wedge$
4. $p_i.center \leq (p_{i-1}.maxY)$

where

$$p_i.center = \frac{p_i.minY + p_i.maxY}{2}$$

Then each $p_i \in \tau_j, i = 1, 2, \ldots, m$ satisfies a horizontal relationship expressed by

$$\{\forall_{i \in \tau_j} p_i \in \tau_j : \; i \geq 1 \wedge (p_i.minX - \tau_j^m.maxX) < \Theta\} \tag{3.7}$$

where Θ represents a threshold for the maximum distance (measured in pixels for this applications) between two words and τ_j^m is the last element of τ_j (which represents the rightmost word of the island).

An island, τ_i, has the coordinates of rectangle similar to that of the features for ψ_{cb} and ψ_w. Thus, an island has the parameters

$$\tau_i.x, \tau_i.y,$$

$$\tau_i.width, \tau_i.height,$$

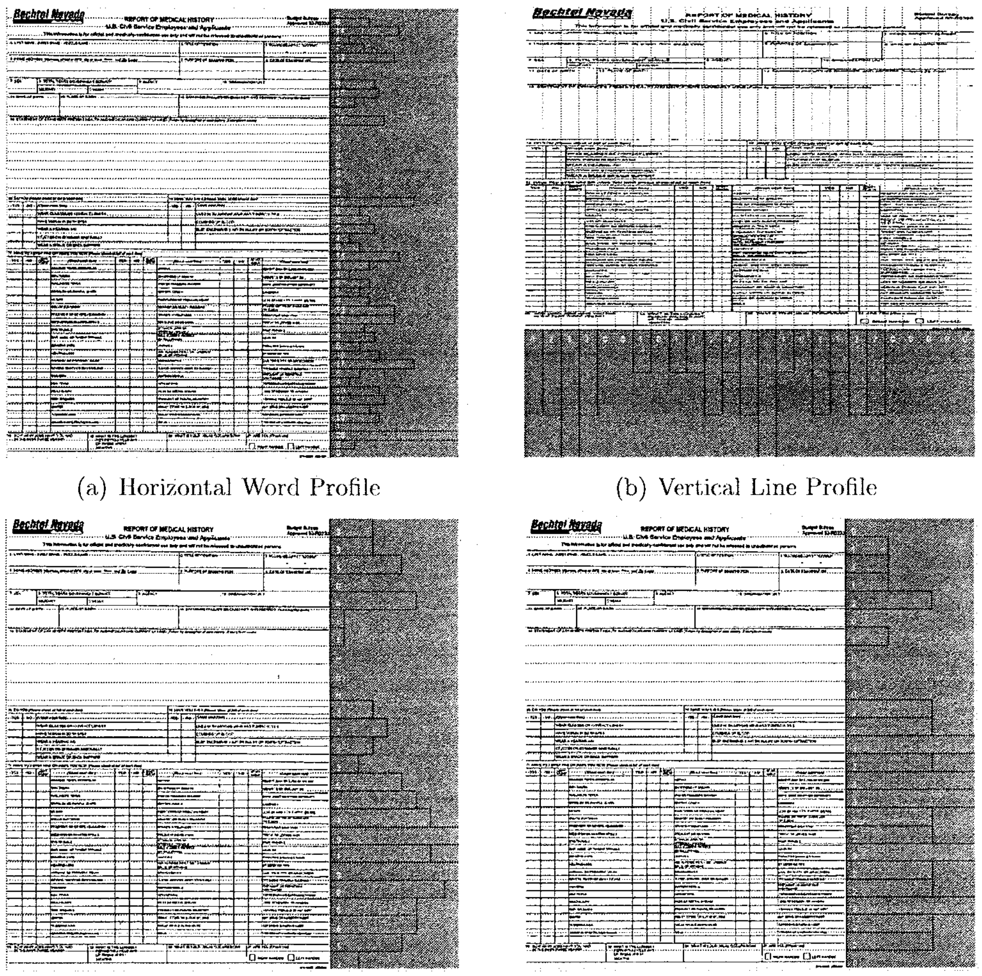
$$\tau_i.centerX, \tau_i.centerY$$

where $(\tau_i.centerX, \tau_i.centerY)$ is the center point of the island rectangle and is computed as in equations (3.3) and (3.4).

Having now defined an island, τ_i in terms of the rules in (3.6) and (3.7), an island builder algorithm is presented as Algorithm 5.

The **hip** forms a profile along the vertical axis of the image where $\mathbf{hip}_i$ is the number of islands from, $\{\tau_1, \tau_2, \ldots, \tau_m\}$, for which the center y, $\tau.centerY$, pass through histogram element i. The image is divided into B equal sized bins. The algorithm for constructing **hip** computes the number of islands that are in bin i

$$\mathbf{hip}_i = \sum_{j=0}^{m} \phi_r(\tau_j, i, B)$$



(a) Horizontal Word Profile

(b) Vertical Line Profile

(c) Horizontal Island Profile

(d) Horizontal Line Profile

Figure 6: Four different profile features

Algorithm 5: Translation for island construction

Input: P
Output: $T = \{\tau_1, \tau_2, \ldots, \tau_M\}$

```
Sort the set P by p_i(r).minX
create first island, τ_1
add p_1, τ_1.add(p_1)
found = false
index = −1
i = 2
for i = 1 TO 3 do                                   /* Run 2 to 3 times */
    while !found AND i < n do
        for j = 1 TO k do
            if vrtalign(τ_j^m, p_i) AND (p_j.minX − τ_j^m.maxX) < Θ then
                index = j
                found = true
                break
            end
        end
    end
    if found then
        p_i to τ_index, τ_index.add(p_i)
    else
        increment the number of islands, k = k + 1
        create a new τ_k and add p_i to it, τ_k.add(p_i)
    end
end
```

where

$$\phi_r(\theta, k, B) = \begin{cases} 1, & k = \theta.centerY * B \\ 0, & otherwise \end{cases} \tag{3.8}$$

The remaining profile features, **hlp**, **vlp** and **twp** are constructed the same way where ϕ_r is a horizontally based projection function for rectangles, ϕ_l is a horizontally based projection function for structural line segments and φ_l is the vertical based projection function for line segments. Thus,

$$\mathbf{hwp}_i = \sum_{j=0}^{N_w} \phi_r(\psi_w^{(p,j)}, i, B)$$

where N_w is the number of typewritten words, $psi_w^{(p,j)}$ is the j^{th} word in the p^{th} image, and B is the number of bins in which the image is divided vertically. Then,

$$\mathbf{hlp}_i = \sum_{j=0}^{N_h l} \phi_l(\psi_{hl}^{(p,j)}, i, B_{hlp})$$

$$\mathbf{vlp}_i = \sum_{j=0}^{N_v l} \varphi_l(\psi_{vl}^{(p,j)}, i, B_{vlp})$$

and

$$\phi_l(\theta, k, B) = \begin{cases} 1, & k = \theta.minY * B \\ 0, & otherwise \end{cases}$$

$$\varphi_l(\theta, k, B) = \begin{cases} 1, & k = \theta.minX * B \\ 0, & otherwise \end{cases}$$

CHAPTER 4

HIERARCHICAL CLASSIFICATION ALGORITHM

Optimal Feature Selection

The hierarchical clustering scheme relies on an ordering of feature vectors. Previous work uses a Fisher class separability measure [70]. Presented here is a technique using conditional probabilities and iterative construction. Let $\theta^{(t)}$ be an ordered configuration of n_t vectors at time t where $\theta_i^{(t)}$ specifies the feature configuration in the i^{th} element of $\theta^{(t)}$ where $\theta_i^{(t)} \in V = \{hlc, vlc, cbc, \ldots, vlp\}$ (Figure 1) and $1 \leq n_t \leq M$.

Then we create a feature vector constructor function, z, that creates a vector $\mathbf{s}$ from an ordered configuration $\theta^{(t)}$ from time t where

$$\begin{aligned} \mathbf{s} &= z(\theta^{(t)}) \\ \mathbf{s} &= z(\{hlc, vlc, cbc, \ldots, vlp\}) \\ \mathbf{s} &= [\mathbf{hlc}, \mathbf{vlc}, \mathbf{cbc}, \ldots, \mathbf{vlp}] \end{aligned}$$

The feature vector constructor function, z, uses the feature translation functions discussed in Section 3.1 to create the individual vectors, and then concatenates features in the order specified by $\theta^{(t)}$.

As mentioned earlier, the hierarchical clustering scheme relies on a particular ordering of the feature vectors. Vector configurations are ordered based on classification accuracies of a given configuration, $\theta^{(t)}$ at time t, $1 \leq t \leq K$ for K different configurations.

More formally, a subset of document images, $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_d\}$ are partitioned into k disjoint subsets, $\{\pi_1, \pi_2, \ldots, \pi_k\}$, where each π_i represents a class of document types and $\mathbf{s}$ denotes the vector representation of a document image. Let Ω be the set of document types where

$$\Omega = \bigcup_{j=1}^{k} \pi_j = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_d\}, \quad \pi_j \cap \pi_l = \phi, \ j \neq l.$$

Let L_i be the event that the recognized document is the i^{th} document type and let l_k be the event that the actual document is the k^{th} document type, where $i, k \in \Omega$.

The Conditional Probability rule states that given two events A and B that

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

where $P(A|B)$ is the probability for A, if B has happened which gives

$$P(A \cap B) = P(A|B)P(B) \tag{4.1}$$

So, $P(L_i)$ can be expressed as

$$\begin{aligned} P(L_i) &= P(L_i \cap \Omega) \\ &= P(L_i \cap \{l_1, l_2, \ldots, l_N\}) \\ &= P(L_i \cap l_1) + P(L_i \cap l_2) + \ldots + P(L_i \cap l_N) \end{aligned}$$

and from equation (4.1), we get

$$
\begin{aligned}
P(L_i) &= P(L_i|l_1)P(l_1) + P(L_i|l_2)P(l_2) \\
&\quad + \ldots + P(L_i|l_N)P(l_N) \\
&= \sum_{k=1}^{N} P(L_i|l_k)P(l_k)
\end{aligned} \tag{4.2}
$$

For a given θ, the parameter, Θ, is defined as $\Theta(\theta) = [\mathbf{r}(\theta), \boldsymbol{P}]$ where $\boldsymbol{P} = \{P_1, P_2, \ldots, P_k\}$ and $P_i = P(L_i)$ and $\mathbf{r}_j(\theta)$ is the mean of the feature construction θ for π_j. Thus, the conditional probability for an input vector $\mathbf{x}_i$ and a class π_j is $P(\pi_j|\mathbf{x}_i; \Theta(\theta))$ and $\mathbf{x}_i$ is classified to the class π_j, if

$$
P(\pi_j|\mathbf{x}_i; \Theta(\theta)) > P(\pi_l|\mathbf{x}_i; \Theta(\theta)), \quad \forall l \in \Omega, j \neq l \tag{4.3}
$$

Then, a function a is created that describes the accuracy of classification for a given θ where

$$
a(\theta) = \frac{total\ correctly\ classified}{d}, \quad 1 \leq j \leq k \tag{4.4}
$$

where, d is the total number of document images classified and k is the total number of classes. Then a series of configurations, $\theta^{(t)}$, $1 \leq t \leq K$ is created where

$$
a(\theta^{(t)}) \geq a(\theta^{(t+1)}) \tag{4.5}
$$

The algorithm for selecting an optimal feature construction is based on an algo-

Individual Feature Classification Accuracy

	a
hlc	.3285
vlc	.1991
cbc	.1336
twc	.4633
cbg	.3515
cbrg	.3023
hip	.8281
hlg	.6533
hlp	.7838
twg	.8567
twrg	.8221
vlg	.5604
vlp	.6211
hwp	.8925

Table 2: The classification accuracy of each individual feature.

rithm presented in [66]. The first step is to establish the most accurate individual feature. So, a classification is performed using each feature by itself as a configuration. The results for each feature of this test are seen in Table 2. As seen, the most accurate individual feature is the horizontal word profile ($\theta = \{hwp\}$). The remainder of the algorithm is now presented (Algorithm 6).

Each step after the initialization is the appending of vectors onto the previous winner. For example, after the first step, each possible two-dimensional combination with the winner from the initialization step and each remaining feature from V is generated, and then a is computed for that combination. Then the feature that, when added, minimizes a (line 6), is the feature that is added to the configuration for time t (line 7).

Algorithm 6: Optimal Feature Selection Algorithm

Input: X, Ω, V

Output: series of configuartions θ

initially $\theta = \emptyset$

compute j where $V_j = \arg\min_{1 \leq i \leq |V|} a(V_i)$

$\theta^{(0)} = V_j$

remove V_j from V

for $\underline{t = 1 \text{ TO } |V|}$ **do**

  compute $V_j = \arg\min_{1 \leq i \leq |V|} a\left(\{\theta^{(t-1)}\} \bigcup V_i\right)$

  $\theta^{(t)} = \{\theta^{(t-1)}\} \bigcup V_j$

  remove V_j from V

end

return θ

Thresholding

The distance function, $\phi(I, C; \theta)$ computes the distance between an input image I and a cluster C. Where the parameter, θ specifies the feature configuration to use to compute the distance. The distance function, $\phi(I, C; \theta)$, uses a relative distance measure, $d(\mathbf{x}, \mathbf{y})$, between two vectors, $\mathbf{x}$ and $\mathbf{y}$,computed by

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(\frac{\mathbf{x}_j - \mathbf{y}_j}{\mathbf{x}_j + \mathbf{y}_j} \right)^2}$$

For each $\theta^{(t)}$ at a given time t, there are two bounds for the distance measure $\phi(I, C; \theta^{(t)})$. These bounds correspond to the range in which an image I has membership within a cluster C. The lower bound for $\theta^{(t)}$ is $\Phi(\theta^{(t)})$ where $\Phi_t = \Phi(\theta^{(t)})$. The upper bound for $\theta^{(t)}$ is $\Upsilon(\theta^{(t)})$ where $\Upsilon_t = \Upsilon(\theta^{(t)})$. The bounds were determined by computing statistics of inner-class and between class relationships.

Recall from Section 4.1 that a subset of images $S = \{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_d\}$ are partitioned into k disjoint subsets, $\{\pi_1, \pi_2, \ldots, \pi_k\}$, where each π_i represents a class of document types and $\mathbf{s}$ denotes the vector representation of a document image. Let Ω be the set of document types where

$$\Omega = \bigcup_{j=1}^{k} \pi_j = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_d\}, \quad \pi_j \cap \pi_l = \phi, \ j \neq l.$$

Given a document type, k, the inner class distances matrix for the feature decomposition $\theta^{(t)}$, $H^{(\theta^{(t)},\pi_k)}$, is computed where $H_{ij}^{(\theta^{(t)},\pi_k)}$ is the relative distance between $\mathbf{s}_i$ and $\mathbf{s}_j$ and

$$H_{ij}^{(\theta^{(t)},\pi_k)} = d(\mathbf{s}_i, \mathbf{s}_j)$$

where $\mathbf{s}_i, \mathbf{s}_j \in \pi_k$. The two statistics computed are the mean, $\mu(\theta^{(t)}, \pi_k)$, and standard deviation, $\sigma(\theta^{(t)}, \pi_k)$ of $H^{(\theta^{(t)},\pi_k)}$ where

$$\mu(\theta^{(t)}, \pi_k) = \frac{1}{d^2} \sum_{m=1}^{d} \sum_{n=1}^{d} H_{mn}^{(\theta^{(t)},\pi_k)} \tag{4.6}$$

and

$$\sigma(\theta^{(t)}, \pi_k) = \sqrt{\frac{1}{d^2} \sum_{m=1}^{d} \sum_{n=1}^{d} (H_{mn}^{(\theta^{(t)},\pi_k)} - \mu(\theta^{(t)}, \pi_k))^2} \tag{4.7}$$

Figure 8 shows the plot for inner distance measures for *hwp* over all clusters specified by $H^{(\theta_{hwp},\Omega)}$ where

$$H^{(\theta_{hwp},\Omega)} = \bigcup_{j=1}^{k} H^{(\theta_{hwp},\pi_j)} \tag{4.8}$$

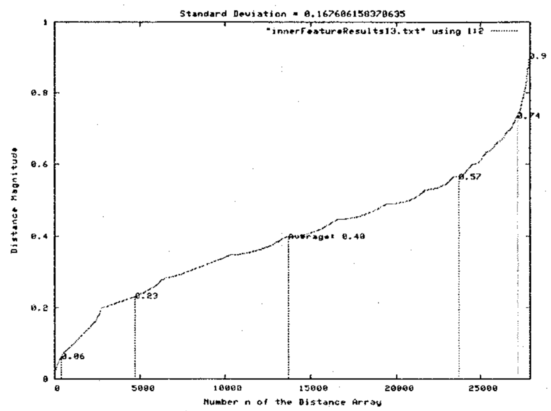


Figure 7: The sorted values of $H^{(\theta^{hwp},\Omega)}$ with the corresponding $\mu(\theta^{(t)},\pi_k) = .40$ and $\sigma(\theta^{(t)},\pi_k) = .1676$ plotted.

The first choice for a Φ_t is the mean, $\mu(\theta^{(t)},\Omega)$ where

$$\mu(\theta^{(t)},\Omega) = \frac{1}{k}\sum_{j=1}^{k}\mu(\theta^{(t)},\pi_j) \tag{4.9}$$

and Φ_t is not varied more than one standard deviation, $\sigma(\theta^{(t)},\Omega)$ from the mean $\mu(\theta^{(t)},\Omega)$ where

$$\sigma(\theta^{(t)},\Omega) = \sqrt{\frac{1}{kd^2}\sum_{l=1}^{k}\sum_{m=1}^{|\pi_k|}\sum_{n=1}^{|\pi_k|}(H_{mn}^{(\theta^{(t)},\pi_k)} - \mu(\theta^{(t)},\pi_k))^2} \tag{4.10}$$

This analysis allows us to find values for each, Φ_t for each, $\theta^{(t)}$. The process for determining Υ is very similar.

The between class matrix for the feature decomposition $\theta^{(t)}$, $B^{(\theta^{(t)})}$ is computed where $B_{ij}^{(\theta^{(t)})}$ is the relative distance between $\mathbf{c}_i$ and $\mathbf{c}_j$

$$B_{ij}^{(\theta^{(t)})} = d(\mathbf{c}_i,\mathbf{c}_j) \tag{4.11}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are concept vectors chosen from the $\pi_i = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\pi_i|}\}$ and $\pi_j = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\pi_j|}\}$. Every cluster, π_i, has a representative, $\mathbf{r}_i$, and $\mathbf{c}_i$ is chosen by

$$\mathbf{c}_i = \arg \min_{1 \leq j \leq |\pi_i|} d(\mathbf{s}_j, \mathbf{r}_i) \tag{4.12}$$

where $\mathbf{c}_i$ represents the small of all pair-wise relative distance measures between a member $\mathbf{s}_j$ and the representative $\mathbf{r}_i$.

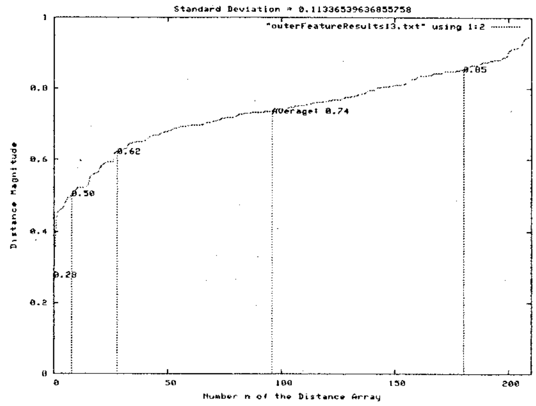


Figure 8: The sorted values of $B^{(\theta_{hwp})}$ with the corresponding $\mu(\theta_{hwp}) = .74$ and $\sigma(\theta_{hwp}) = .1134$ plotted.

As for $H^{(\theta^{(t)}, \Omega)}$, the mean $\mu(\theta^{(t)})$ and standard deviation $\sigma(\theta^{(t)})$ are computed for $B^{(\theta^{(t)})}$ where

$$\mu(\theta^{(t)}) = \frac{1}{k^2} \sum_{m=1}^{k} \sum_{n=1}^{k} B_{mn}^{(\theta^{(t)})} \tag{4.13}$$

and

$$\sigma(\theta^{(t)}) = \sqrt{\frac{1}{k^2} \sum_{m=1}^{k} \sum_{n=1}^{k} \left(B_{mn}^{(\theta^{(t)})} - \mu(\theta^{(t)}) \right)^2} \tag{4.14}$$

Then Υ_t is chosen to be within one standard deviation, $\sigma(\theta^{(t)})$ of the mean $\mu(\theta^{(t)})$.

Classification

The classification algorithm is an unconstrained sequential clustering based scheme in which (1) the number of clusters is unknown, (2) the number of samples to be classified is unknown and (3) no *a priori* knowledge is presented. This algorithm is useful for problems in which it is not feasible for the entire data set to reside in memory and the supervised training of the entire set cannot practically be accomplished with a human effort.

Let θ be an ordered set of n vectors from the set $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$ where $\theta_i, \theta_j \in \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$ and $1 \leq i, j \leq M,\ i \neq j$ and $1 \leq |\theta| \leq M$.

The hierarchical algorithm, Algorithm 7 and Figure 9, uses the ordered set θ and the list of lower and upper bounds, Φ and Υ respectively.

For an arbitrary image, I, the algorithm uses each feature, θ, to essentially eliminate those document types (clusters) to which I could not belong. Then, if after populating the set $inSet$ (line 11), the set $inSet$ has only one element, then that is a termination condition that causes I to be classified to that cluster (line 22). If the set $inSet$ has more than one member, then those clusters belonging to that set are placed in $currSet$ and set to be classified for the next feature in θ (line 20).

If, after going through each cluster (for loop on line 9), and $inSet$ is empty, $|inSet| = \emptyset$, then there are no clusters currently to which I could belong. This constitutes the creation of a new document type (cluster) where I becomes the first element of that cluster (line 26-27).

When I is marked as unclassified, this means that I was neither added to a cluster nor constituted the creation of a cluster. This occurs, if at the end of the

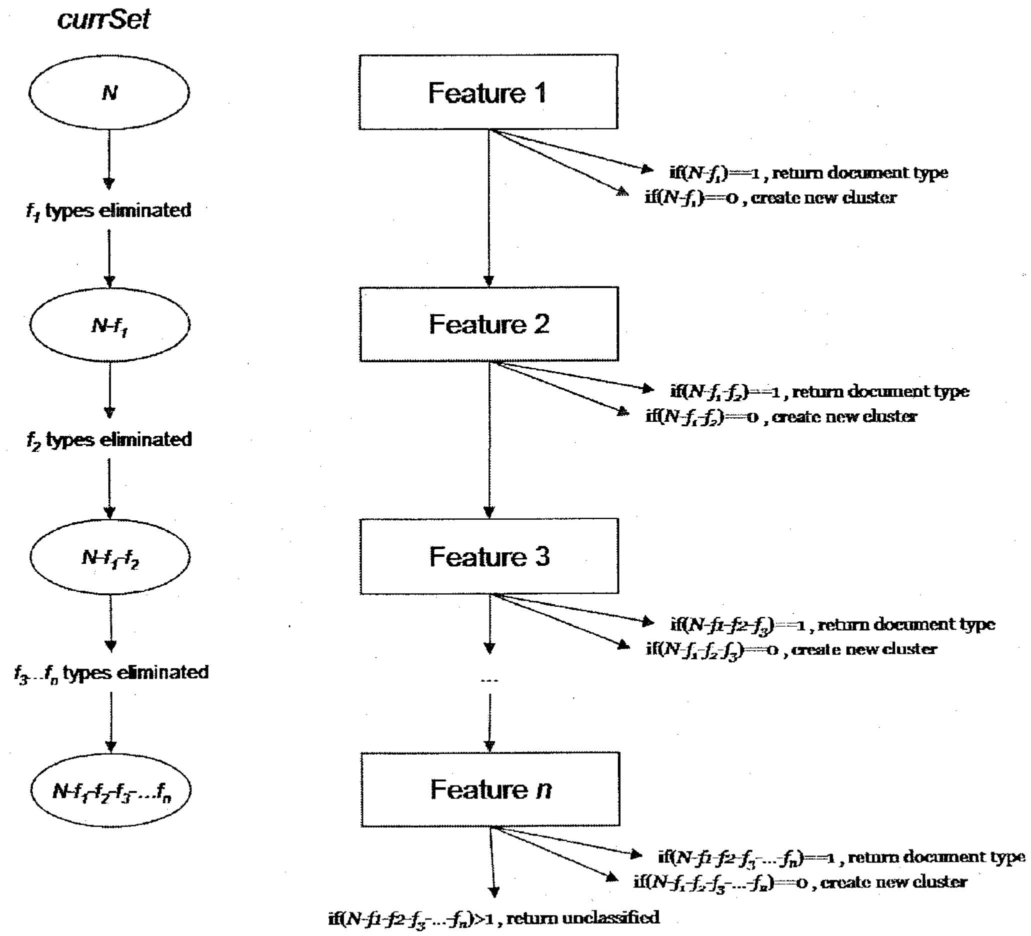


Figure 9: An execution of the Hierarchical Document Type Classification algorithm shows how document types are eliminated at each feature level. At each phase ($Feature_1, Feature_2, \ldots, Feature_n$), potential document types are eliminated based upon a distance measure and a threshold determined for that distance measure. At the end, if there are more than 1 potential document types, then that input image, I, is marked as unclassified.

Algorithm 7: Hierarchical Document Type Classification

```
   Input: I, θ, Φ, Υ, C
1  classified = false, i = 1
2  inSet = ∅, outSet = ∅, midSet = ∅
3  if |C| == 0 then
4      C.numberOfClusters++
5      C_numberOfClusters = {x}
6  end
7  currSet = C
8  while classified == false AND i < |θ| do
9      for j = 1 TO |currSet| do
10         if φ(I, currSet_j; θ_i) < Φ_i then
11             inSet.add(currSet_j)
12         else if φ(I, currSet_j; θ_i) > Υ_i then
13             outSet.add(currSet_j)
14         else
15             midSet.add(currSet_j)
16         end
17     end
18     if |inSet| > 1 then
19         i++
20         currSet = inSet
21         outSet = ∅, midSet = ∅
22     else if |inSet| == 1 then
23         add I to cluster represented by inSet
24         classified = true
25     else // |inSet| = ∅
26         create new cluster
27         make I first member of that cluster
28         classified = true
29 end
30 if classified == false then
31     mark I as unclassified
32 end
```

while loop (line 8), the variable $classified$ is still equal to false, $classified = false$.

Automatic Cluster Adjustment Stage

The method described in [72] uses a cluster intensity function to determine boundaries and separability of clusters. Presented here are methods that not only detect these features, but determine their strength and makes decisions whether or not to merge or split a cluster based on its boundaries and separability. An important step in the classification process is a refinement step referred to as Automatic Cluster Adjustment (ACA). Throughout the process, clusters can grow in unpredictable and sometimes undesirable directions. Thus, ACA is implemented after a classification has occurred on n document images. ACA is a three step process: 1) the first step is a merging procedure, 2) the second step is a splitting procedure, and 3) the third step is an attempt to classify those images marked as unclassified.

It is possible that two (or more) different clusters can exist that actually represent the same document image type. To handle this problem, a merging procedure was developed that first detects if this situation exists, and then does the necessary work to merge. The algorithm for merging is presented as Algorithm 8. The input to the merging procedure is the specification of what feature configuration, $\theta^{(t)}$, to use and the current clustering, C, of the system. The merging of two clusters is based on which configuration and the corresponding threshold, Φ_t. The *add*() operation takes each individual member of C_j and adds it to C_i. This adding of individual members automatically updates the representative for the cluster C_i.

Algorithm 8: Merging Procedure

Input: $\theta^{(t)}$,C

Find $C_i, C_j (i < j)$ where $\phi(C_i, C_j; \theta^{(t)}) = \min_{k,r=1,2,\ldots,|C|,k\neq r} \phi(C_k, C_r; \theta^{(t)})$

if $\underline{d(C_i, C_j) < \Phi_t}$ **then**

    merge C_j into C_i using the *add*() operation

    eliminate C_j

end

For clusters that grow but have low cohesion, a method of splitting is employed that marks each member of that cluster as temporarily unclassified. The difficult part is detecting clusters that must be split. This process uses the feature vector representation of the members and makes a decision based on what percentage fall within a certain range. For the i^{th} cluster, $C^{(i)}$, each member, $C_j^{(i)}, 1 \leq j \leq |C^{(i)}|$ is translated into its vector representation, $\mathrm{s}^{(i,j)}$ by the translation function, z, described in Section 4.1. The first step is to create the mean vector, $\overline{\mathrm{s}}^{(i)}$ for $C^{(i)}$ where

$$\overline{\mathrm{s}}_k^{(i)} = \frac{1}{|C^{(i)}|} \sum_{r=1}^{|C^{(i)}|} \mathrm{s}_k^{(i,r)} \tag{4.15}$$

and then create a vector for the standard deviation, $\sigma^{(i)}$, where

$$\sigma_k^{(i)} = \sqrt{\frac{1}{|C^{(i)}|} \sum_{r=1}^{|C^{(i)}|} \left(\mathrm{s}_k^{(i,r)} - \overline{\mathrm{s}}_k^{(i)}\right)^2} \tag{4.16}$$

The splitting is presented in Algorithm 9. The algorithm first goes through each element of cluster i and, if at least R elements of that vector are within one standard deviation of the mean vector for that cluster, then that element is counted (lines 9-10). Then if there are less than P members who satisfy the above rule, then that cluster is split, otherwise, the cluster is not split (lines 14-15).

The final step of the update procedure is an attempt to classify those images that have been marked as unclassified. Those images that were split in the previous step using the cluster splitting algorithm, are marked as temporarily unclassified.

Algorithm 9: Cluster Splitting

Input: $C^{(i)}$

$inCount = 0$
$memberCount = 0$
for $r = 1$ TO $|C^{(i)}|$ **do**
    for $k = 1$ TO $|\mathrm{s}^{(i,r)}|$ **do**
        if $-1 \leq \frac{\mathrm{s}_k^{(i,r)} - \bar{\mathrm{s}}_k^{(i)}}{\sigma_k^{(i)}} \leq 1$ **then**
            $inCount++$
        end
    end
    if $inCount > R$ **then**
        $memberCount++$
    end
    $inCount = 0$
end
if $memberCount < P$ **then**
    splitCluster($C^{(i)}$)
end

This distinguishes them from those images that were already unclassified before the update procedures started. The classification of unclassified images is done using the hierarchical document type classification shown as Algorithm 7. After classification is complete, then those images marked as temporarily unclassified are marked as classified.

CHAPTER 5

RESULTS

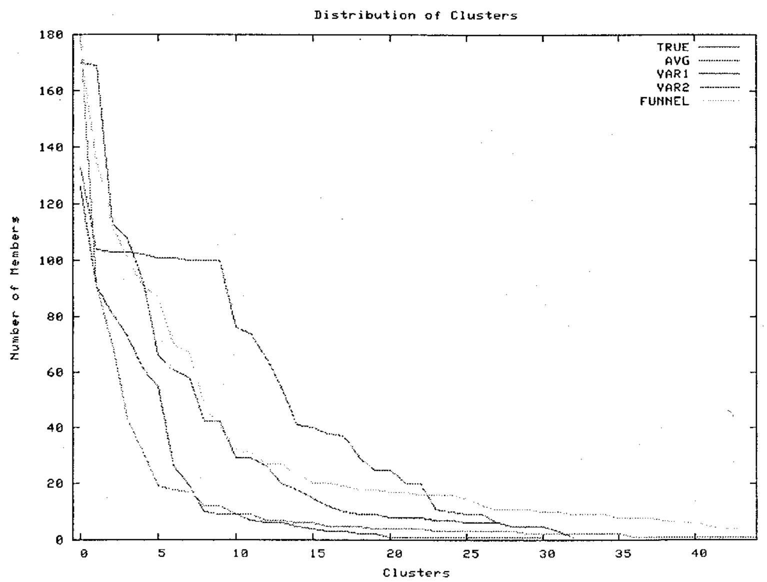
Since the class labels of an image I is not known *a priori* , examining the results of a clustering algorithm is more of a qualitative process. 1647 images were divided into their respective types by hand and given a name (these names were not available during the classification). The algorithm creates arbitrary groups of images, which are not related to the names given to each image. So, results are done based on looking at the members and providing statistics based on member names.

The experimental setup includes four different configurations of thresholds in two different environments. One environment is without the ACA process and the other environment is with the ACA process. The four different threshold configurations used in each environment are

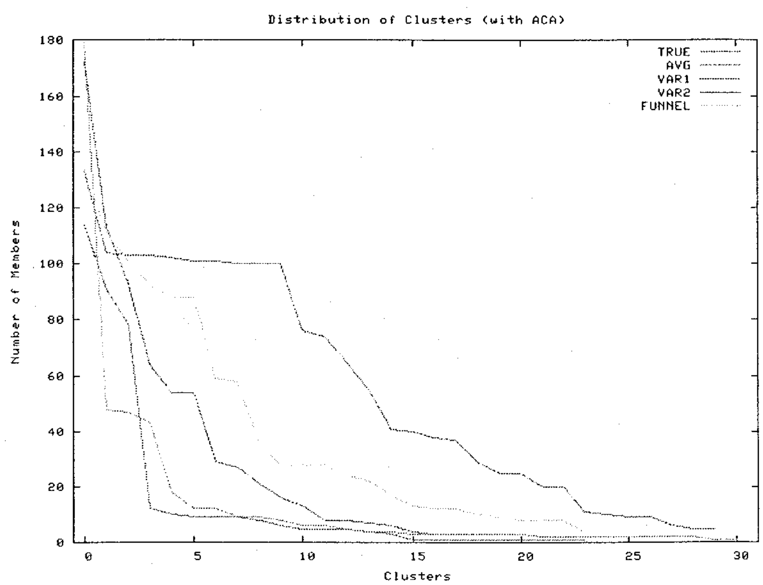
- AVG - This configuration uses the averages of the inner-class distances (for lower bounds) and between-class distances (for upper bounds) as described in Section 4.2.

- VAR1 - This is the first variation of threshold adjustments. Adjustments are made within one standard deviation of the mean for inner-class distances (for lower bounds) and between-class distances (for upper bounds).

- VAR2 - This is the second variation of threshold adjustments. Adjustments are



(a) Distribution without ACA



(b) Distribution with ACA

Figure 10: The distribution of the images in clusters.

Cluster	NTC	Count	PMT
15214	1	88	1.00
15172	2	10	0.90
15173	5	8	0.375
15174	2	92	0.99
15175	2	28	0.96
15177	2	4	0.75
15178	4	88	0.95
15190	3	24	0.54
15179	2	101	0.99
15182	2	28	0.96
15181	2	17	0.94
15211	1	8	1.00
15186	2	135	0.75
15187	1	35	1.00
15192	3	111	0.97
15207	1	58	1.00
15208	1	22	1.00
15212	1	12	1.00
15213	1	13	1.00
15205	2	28	0.71
15204	1	9	1.00
15199	2	12	0.92
15210	1	8	1.00
15201	1	59	1.00

Table 3: Cluster Analysis for FUNNEL

made within two standard deviation of the mean for inner-class distances (for lower bounds) and between-class distances (for upper bounds).

- FUNNEL - Funneling for the thresholds refers to the process of making the choice for the lower/upper bounds at the first level of the decomposition very lenient and at each level decrease the leniency (relative to the average and standard deviation for the configuration at a given level). This funnels the images towards the optimal configuration.

One of the important results is how many different document types there are in each cluster (NTC) and the percentage of the majority type (PMT). Table 3 shows the result of a clustering performed using a FUNNEL configuration with ACA. Many of the clusters have a PMT equal to one or in the high to low nineties. This means the clustering algorithm is not only finding clusters, it is also classifying with high accuracy. Cluster 15173 is a topic of interest and future work. The document types in this cluster were closely related based on the features selected. We address work being done to solve this problem in our conclusions and discussions.

Figure 10 shows the distribution of the images based on how many members there are in each cluster versus the distribution of the members in the true classification for an experiment performed using no ACA (Figure 10a) and with ACA (Figure 10b). The important aspect of this result is that the trend is similar. Even without ACA, the distribution of the clusters discovered creates a trend similar to the true classification. With ACA, though, the trend becomes closer. There are drop offs and level areas in the same relative regions. Without ACA, there is a spike in the beginning, meaning that clusters were found that contained many images, but with ACA, it is seen that there are no such spikes. This is due to the splitting method of ACA.

Additional statistics useful in analyzing a clustering experiment are presented in Table 4 and Table 5. The *Percent Correct out of Total* tells us how many images are correctly classified out of all possible images in the experiment (1647 for this particular experiment). Being constituted as *correctly classified* means an image is the same as the majority type of that cluster. The *Percent Correct out of Classified* computes accuracy only on those images that were marked as classified. Then the information is provided with the number of images that were correct and exactly how many images were left unclassified.

	AVG	VAR1	VAR2	FUNNEL
Percent Correct out of Total	.3339	.5664	.3333	.7523
Percent Correct Out of Classified	.8607	.7954	.9015	.8806
Number of Clusters	75	33	31	44
Images Correct	550	933	549	1239
Image Unclassified	1008	474	1038	240

Table 4: Threshold Variation Results Without ACA

	AVG	VAR1	VAR2	FUNNEL
Percent Correct out of Total	.23	.2028	.3479	.5616
Percent Correct Out of Classified	.81	.89	.83	.9269
Number of Clusters	60	24	17	24
Images Correct	387	334	573	925
Image Unclassified	1171	1270	956	649

Table 5: Threshold Variation Results With ACA

When looking at whether a configuration is successful, the most important number is its Percent Correct Out of Classified. As seen from Table 4 and Table 5, accuracies are in the 80's and 90's. Upon further observation, it is seen that there are sometimes many images left unclassified. This is a negative result, along with the number of clusters created. Since we know that the true number is 30, we want to mitigate the creation of clusters beyond 30. As seen, the FUNNEL without ACA had high accuracy and left little images unclassified, but created 44 clusters. Then when run with ACA, FUNNEL decreases the number of clusters (through merging and splitting) and increases accuracy. Although, more images are marked unclassified. This result is still positive because the production session is designed to never terminate, so as more information (i.e. more documents are classified) is presented to the system, clusters will be formed later on that could possibly begin classifying these unclassified images.

The key issues with the system are single point of failures. Current efforts are creating relationships between the size of the *midSet* and *outSet* in Algorithm 7. This would add intelligence on the nature of the relationship between an image I and the current state of the clustering. Results show that such an algorithm can be successful in automatic discovery of classes in a classification environment and classification of samples into discovered classes.

CHAPTER 6

CONCLUSIONS AND DISCUSSIONS

Presented in this thesis is an algorithm for clustering that performs unsupervised learning on document image types. This algorithm is useful for problems in which it is not feasible for the entire data set to reside in memory and the supervised training of the entire set cannot practically be accomplished with a human effort. The classification algorithm is an unconstrained, sequential clustering-based scheme in which (1) the number of clusters is unknown, (2) the number of samples to be classified is unknown and (3) no *a priori* knowledge is presented. The hierarchical feature decomposition allows for an efficient classification of an image I by eliminating at each stage those clusters for which I could not belong.

The algorithm performed at an exceptional rate and was successful at learning document types autonomously. The FUNNEL method for establishing thresholds was the most successful threshold configuration, achieving a 92% accuracy of classified document images.

APPENDIX A

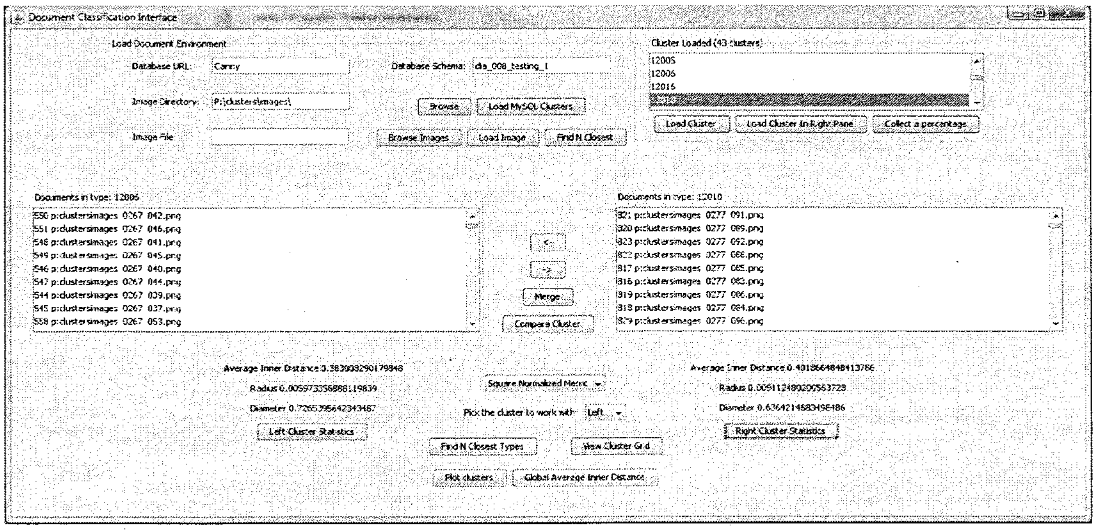


Figure 11: The main panel for the Document Classification Interface

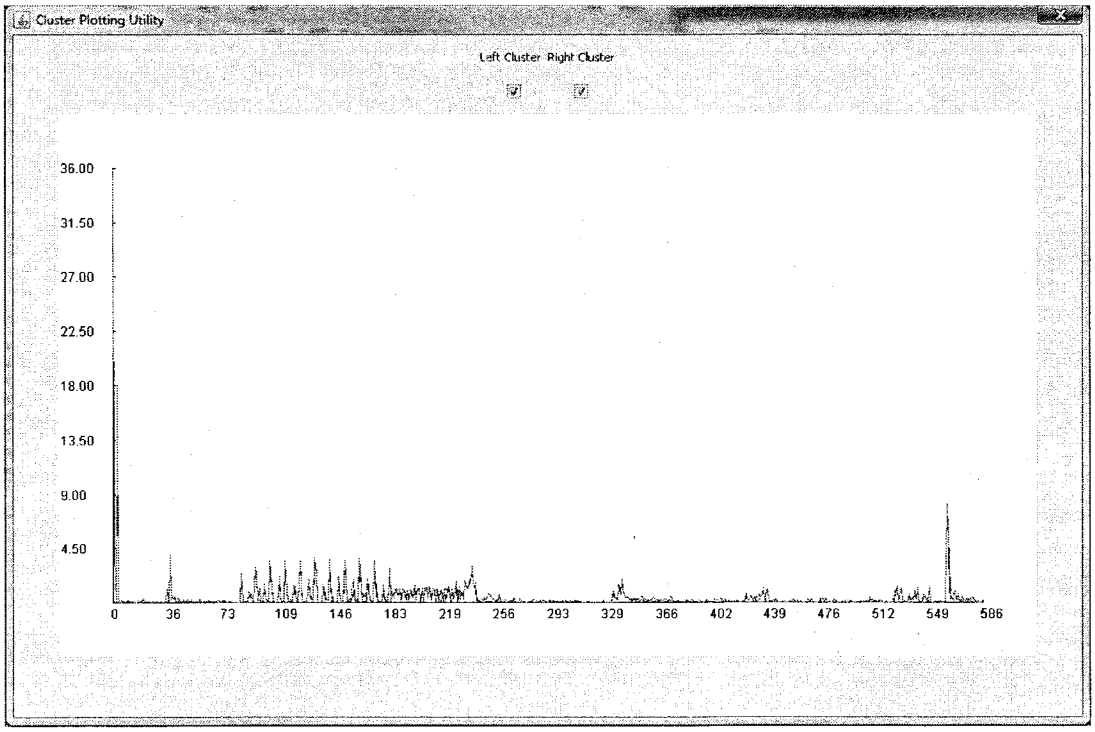


Figure 12: Plot comparison for the feature vector of two cluster representatives

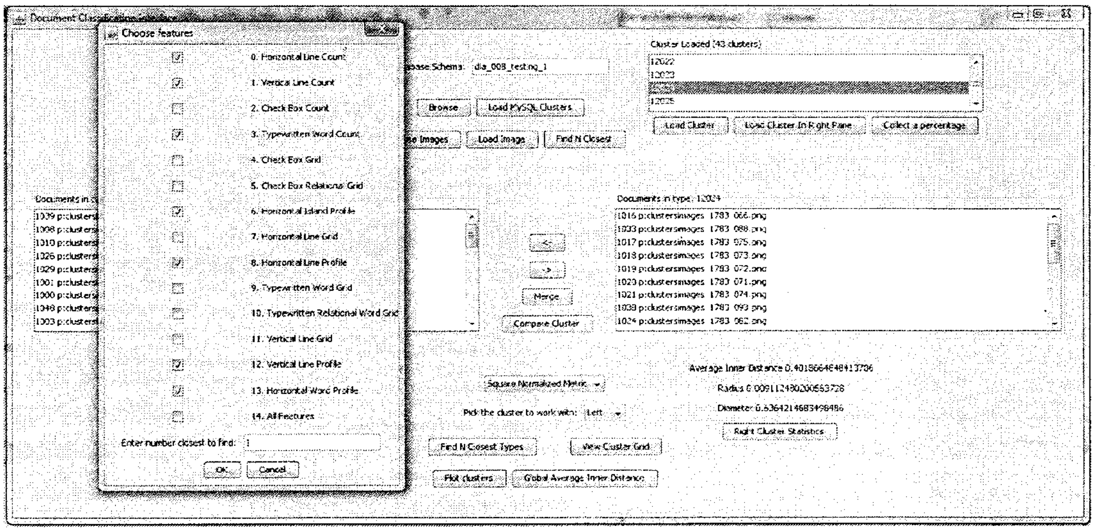


Figure 13: One of the capabilities is in finding the n-closest images. Shown is the selection of the features that must takes place. A distance measure must also be provided.

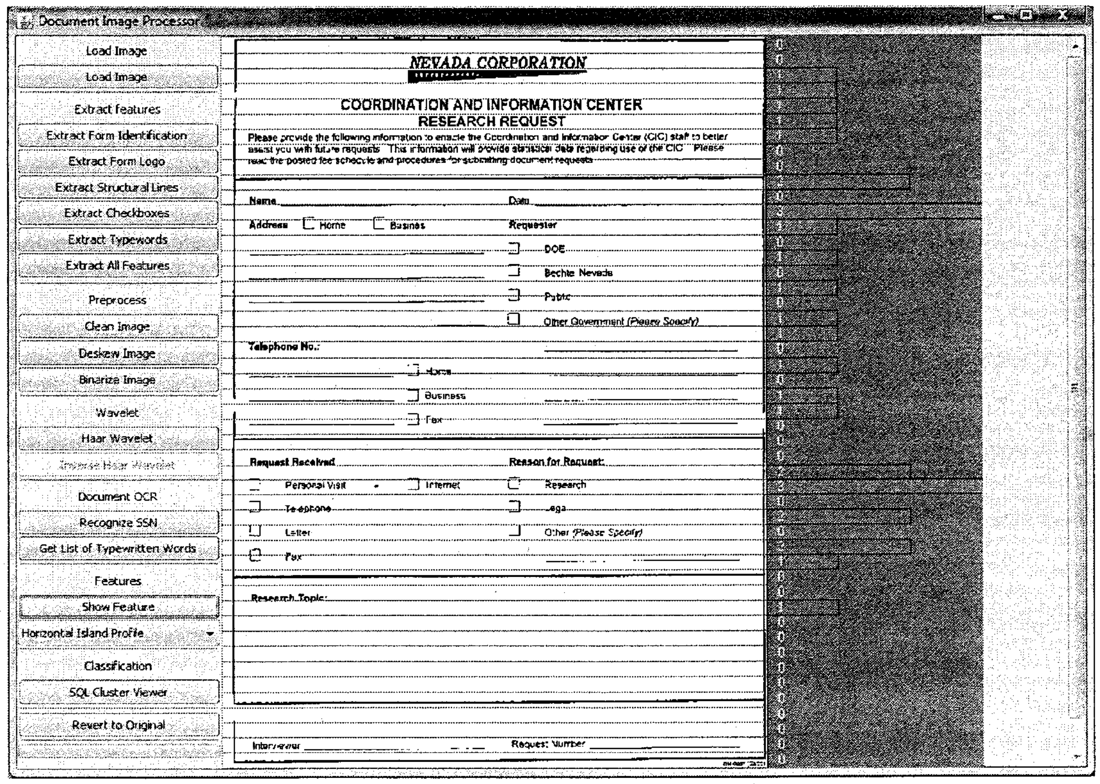


Figure 14: For testing of feature extraction of feature vector construction on individual images, the Document Image Processor was developed

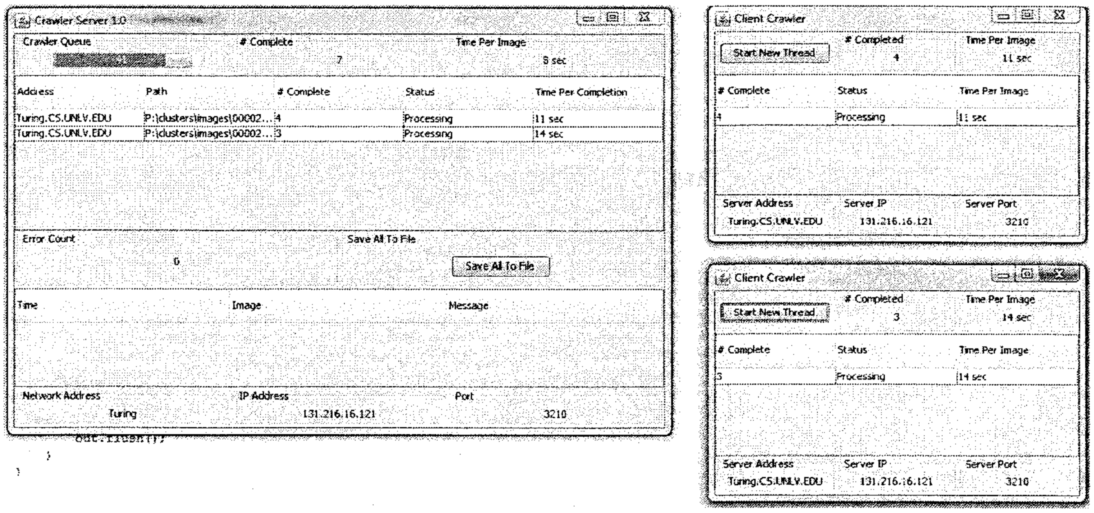


Figure 15: To test on a large set of images, a multi-threaded crawler was developed. The crawler would crawl through the repository of images and distribute images to connected clients. The crawler also is responsible for implementing the document classification algorithm. This it made it possible to perform feature extraction and classification on large set of images. Largest set performed was a little under 300,000

BIBLIOGRAPHY

[1] Tomasz Adamek, Noel E. O'Connor, and Alan F. Seamton "Word Matching Using Single Closed Contours for Indexing Handwritten Historical Documents" *International Journal on Document Analysis and Recognition* vol. 9, issue 2. April 2007

[2] Jae Adams, E.A. Yfantis, D. Curtis and T. Pack. Feature Extraction Methods for Form Recognition Applications. WSEAS trans. on Information Science and Applications, Issue 3, Volume 3 March 2006 Pages 666-671.

[3] A. Agarwal, A. Gupta, K. Hussein, P.S.P Wang "Bank Check Analysis and Recognition by Computers" *Character Recognition and Document Image Analysis* pp.623-651 1997

[4] Sameer Agarwal, Jongwoo Lim, Lihi Zelnik-Manor, Pietro Perona, David Kriegman, and Serge Belongie "Beyond Pairwise Clustering" *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* vol. 2. pp. 838-845

[5] M.R. Anderberg. Cluster Analysis for Applications. *Academic Press* 1973.

[6] Apostolos Antonacopoulos and Andy C. Downton "Special Issue on the Analysis of Historical Documents" *International Journal on Document Analysis and Recognition* vol. 9. issue 2. April 2007

[7] J.A. Aslam, E. Pelekhov and D. Rus "The Star Clustering Algorithm for Information Organization" Grouping Multidimensional Data pp. 1-23 2006

[8] Francis Bach and Michael Jordan "Learning Spectral Clustering" *Technical Report* UC Berkley 2003

[9] A. Banerjee, S. Merugu, I. Dhillon and J. Ghosh "Clustering with Bregman divergences" *Machine Learning Research 6, Journal of* pp. 1705-1749 2005

[10] Thomas Bayer, Ulrich Bohnacker and Ingrid Renz "Information Extraction from Paper Documents" Handbook of Character Recognition and Document Image Analysis pp. 653-677 1997

[11] Burak Bitlis, Xiaojun Feng, Jacob L. Harris, Ilya Pollak, Charles A. Bouman, Mary P. Harper, Jan P. Allebach "A Hierarchical Document Description and Comparison Method" *Proceesings IS&T Archiving Conference* San Antonio, Texas. April 2004

[12] D. Blostein, R. Zanibbi, G. Nagy and R. Harrap "Document Representations" *GREC* 2003

[13] J. Bunch, D. Curtis, C. Jones, J. Tse and E.A. Yfantis "Extracting the Major Form Body Segment from Unconstrained Document Images" *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 75-80

[14] Michael C. Burl "Mining Large Image Collections" *Data Mining for Scientific and Engineering Applications* pp. 63-84 2001

[15] Samuel W.K. Chan and Mickey W.C. Chong "Unsupervised Clustering for Non-textual Web Document Classification" *Decis. Support Syst.* Vol. 37 No. 3 pp. 377-396 2004

[16] Hui Cheng, Charles A. Bouman, and Jan P. Allebach "Multiscale Document Segmentation" *IS&T 50th Annual Conference* pp. 417-425, Cambridge, MA, May 18-23, 1997

[17] Hui Cheng and Charles A. Bouman "Trainable Context Model for Multiscale Segmentation" *Image Processing, 1998. ICIP 98. Proceedingd. 1998 International Conference* vol. 1. pp 620-614. October 1998

[18] Daniel A. Coleman; David L. Woodruff "Cluster Analysis for Large Datasets: An Effective Algorithm for Maximizing the Mixture Likelihood" *Journal of Computational and Graphical Statistics* Vol. 9 No. 4 pp. 672-688 Dec. 2000

[19] T.M. Cover and P.E. Hart "Nearest Neighbor Pattern Classification" *Information Theory, IEEE Transactions on* pages 21-27 Jan. 1967

[20] Dean Curtis, Vitaliy Kubushyn, E.A. Yfantis, and Michael Rogers "A Hierarchical Feature Decomposition Clustering Algorithm for Unsupervised Classification of Document Image Types" *IEEE* 6^{th} *International Conference on Machine Learning and Application* December 13-15, 2007

[21] A. Dengel, F. Dubiel "Clustering and Classification of Document Structure - A Machine Learning Approach" *Document Analysis and Recognition, 1995.,*

Proceedings of the Third International Conference on, Vol.2, Iss., 14-16 Aug 1995 Page 587-591

[22] I. Dhillon, J. Fan and Y. Guan "Efficient Clustering of Very Large Document Collections" book chapter in *Data Mining for Scientific and Engineering Applications* pages 357-381 Kluwer 2001

[23] Indejit Dhillon, Yuqiang Guan and Brian Kulis "Kernel K-means, Spectral Clustering and Normalized Cuts" *KDD'04* ACM, August 22-25 2004 pp. 551-556

[24] Sébastien Diana, Eric Tupin, Yves Lecourtier, Jacques Labiche "Document Modeling for Form Class Identification" *Document Analysis Systems: Theory and Practice* vol.1655 pp.176-187 1998

[25] David Doermann, Huiping Li and Omid Kia "The Detection of Duplicates in Document Image Databases" In *Proceedings of the International Conference on Document Analysis and Recognition* pp. 314-318 1997

[26] Chris Fraley and Adrian raftery "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis" *The Computer Journal* Vol. 41 No. 8 1998

[27] Chris Fraley and Adrian E. Raferty "Model-Based Clustering, Discriminant Analysis, and Density Estimation" *Technical Report No. 380* University of Washingtion October 2000

[28] Pasi Fränti, Olli Virmajoki and Ville Hautamäki "Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Vol. 28 No. 11 November 2006

[29] Hiroko Fujihara and Elmamoun Babiker "Qualitative/Fuzzy Approach to Document Recognition" *Artificial Intellegence for Applications, 1992, Proceedings of the Eight Conference"* pp. 254-269. March 1992

[30] B. C. M. Fung, K. Wang and M. Ester "Hierarchical Document Clustering Using Frequent Itemsets" *Proceedings of the SIAM Internation Conference on Data Mining* 2003

[31] I. Gath and A.B. Gey "Unsupervised Optimal Fuzzy Clustering" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Volume 11 Issue 7 July 1989 pp. 773-780

[32] Y.H. Liu-Gong, B. Dubuisson, H.N. Pham "A general analysis system for document's layout structure recognition" *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on,* Vol.2, Iss., 14-16 Aug 1995 Page 597-600

[33] Yuqiang Guan "Large-Scale Clustering: Algorithms and Applications" *Doctorate Dissertation* Univervisity of Texas, Austin 2006

[34] S. Guha, R. Rastogi and K. Shim "CURE: An Efficient Clustering Algorithm for Large Databases" In *ACM SIGMOD International Conference on Management Data* pp. 73-84 June 1998

[35] S. Guha, R. Rastogi, and K. Shim "Rock. A Robust Clustering Algorithm For Categorical Attributes" In *Proc. of the* 15^{th} *Int'l Conf. on Data Eng.* 1999.

[36] Thien Ha and H. Bunke "Image Processing Methods for Document Image Analysis" Handbook of Character Recognition and Document Image Analysis pp. 1-47 1997

[37] Nadav Ben-Haim, Boris Babenko and Serge Belongie "Improving Web-Based Image Search via Content Based Clustering" *SLAM* 2006

[38] John D. Hobby, Tin Kam Ho "Enhancing Degraded Document Images via Bitmap Clustering and Averaging" *International Conference on Document Analysis and Recognition* pp.394-400 1997

[39] Jonathan J. Hull and John F. Cullen "Document Image Similarity and Equivalence Detection" *International Conference on Document Analysis and Recognition* pp.308-312 1997

[40] J. Istle "Optical Character Recognition for Checkbox Detection" Master Thesis, Department of Computer Science, UNLV 2004

[41] Anil K. Jain and Bin Yu "Document Representation and Its Application to Page Decomposition" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Vol. 20, No. 3 March 1998

[42] Saddok Kebairi, Bruno Taconet, Abderrazak Zahour, Said Ramdane "A Statistical Method for an Automatic Detection of Form Types" *Advances in Document Image Analysis* pp.84-96 1997

[43] Alireza Khotanzad and Yaw Hua Hong "Invariant Image Recognition by Zernike Moments" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Vol. 12 No. 5 May 1990

[44] S. Krishnamachari and M. Abdel-Mottaleb "Hierarchical clustering algorithm for fast image retrieval" In *IS&T/SPIE Conference on Storage and Retrieval for Image and Video databases VII.* pp.427-435 1999

[45] D. Littau and D. Boley "Clustering Very Large Data Sets with Principal Direction Divisive Partitioning" Grouping Multidemensional Data, Springer-Link, 2006 pp. 99-126

[46] T. Liu, A.W. Moore, A. Gray and K. Yang "An Investigation of Practical Approximate Nearest Neighbor Algorithms" In *Advances in Neural Information Processing Systems* , Vancouver, BC, Canada, 2004.

[47] Ting Liu, Charles Rosenberg and Henry A. Rowley "Clustering Billions of Images with Large Scale Nearest Neighbor Search" *wacv* , p. 28, Eighth IEEE Workshop on Applications of Computer Vision (WACV'07), 2007.

[48] Y. H. Liu-Gong, B. Dubuisson, and H. N. Pham "A General Analysis System for Document's Layout Structure Recognition" *Proceedings of the Third International Conference on Document Analysis and Recognition* vol. 2. 1995

[49] Daniel Lopresti "Models and Algorithms for Duplicate Document Detection" *Fifth International Conference on Document Analysis and Recognition* September 1999

[50] Daniel Lopresti "String Techniques for Detecting Duplicates in Document Databases" *IJDAR* 2000 Vol. 2 pp. 186-199

[51] Andrew McCallum, Kamal Nigam and Lyle Ungar "Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching" *Knowledge Discovery and Data Mining* pp. 169-178 2000

[52] George Nagy, Sharad Seth, Mahesh Viswanathan "DIA, OCR, and the WWW" *Character Recognition and Document Image Analysis* 1997 pp. 729-754

[53] G. Nagy "Twenty Years of Document Analysis in PAMI" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Vol. 22 pp. 38-62 2000

[54] Noboru Nakajima, Keiji Yamada, jun Tsukumo "Document Layout and Reading Sequence Analysis by Extended Split Detection Method" *Advances in Document Image Analysis* pp.336-347 1997

[55] Martin Nilsson "Hierarchical Clustering Using Non-Greedy Principal Direction Divisive Partitioning" Journal of Information Retrieval, Springer-Link, November 2004, pp.311-321

[56] Richard Nock and Frank Nielsen "On Weighting Clustering" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Vol. 28 No. 8 August 2006

[57] Lawrence O'Gorman "The Document Spectrum for Page Layout Analysis" *Pattern Anaylsis and Machine Intelligence, IEEE Transactions on* Vol. 15 No. 11 November 1993

[58] Eric Saund, David Fleet, James Mahoney, and Daniel Larner "Rough Document Interpretation by Perceptual Organization" *2003 Symposium on Document Image Understanding Technology* April 2003

[59] Eric Saund, David Fleet, Daniel Larner, and James Mahoney "Perpetually-supported Image Editing of Text and Graphics" *ACM Transactions on Grapics"* vol. 23. Issue 3. August 2004

[60] Gholamhosein Sheikholeslami, Surojit Chatterjee and Aidong Zhang "Wavecluster: A Mulit-Resolution Clustering Approach for Very Large Spatial Databases" *Proceedings of the* 24^{th} *VLDB Conference* 1998 pp. 428-439

[61] Christian Shin and David Doermann "Classification of Document Page Images Based on Visual Similarity of Layout Structures" In *Proc. SPIE* Document Recognition and Retrieval VII Vol. 3967 pp. 182-190 2000

[62] Ramaswamy Sivaramakrishnan, Ihsin T. Phillips, Jaekyu Ha, Suresh Subramanium, and Robert M. Haralick "Zone Classification in a Document using the Method of Feaure Vector Generation" *Proceedings of the Third International Conference on Document Analysis and Recognition* vol. 2. 1995

[63] Kazem Taghva, Julie Borsack and Allen Condit "Information Retrieval and OCR" Handbook of Character Recognition and Document Image Analysis pp. 755-777 1997

[64] Kazem Taghva, Julie Borsack, Steven Lumos, and Allen Condit "A Comparison of Automatic and Manual Zoning: An information retrieval prospective" *In-*

ternational Journal on Document Analysis and Recognition vol. 6 issue 4. April 2003

[65] M. Teboulle, P. Berkhin, I. Dhillon, Y. Guan and J. Kogan "Clustering with Entropy-like k-means Algorithms" book chapter in *Grouping Multidimensional Data : Recent Advances in Clustering* pages 127-160 Springer 2006

[66] S. Theodoridis and K. Koutroumbas "Pattern Recognition" *Academic Press* 2nd Edition 2003

[67] Robert Tibshirani, Guenther Walther and Trevoe Hastie "Estimating the Number of Clusters in a dataset via the Gap Statistic" *Technical Report 208* Department of Statistics, Stanford University 2000

[68] J. Tse, D. Curtis, J. Bunch, C. Jones, E.A. Yfanits and A. Thomas "Handwritten and Typewritten Word and Character Separation in Unconstrained Document Images" *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 40-43

[69] E.M. Voorhees "The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval" PhD Thesis, Cornell University, 1986.

[70] Xudong Wang and Vassilis Syrmos "Optimal Cluster Selection Based on Fisher Class Separability Measure" *Proceedings of the 2005 American Control Conference* June 8-10, 2005 pp.1929-1934

[71] Dihua Xi, Seong-Whan Lee "Table Structure Extraction from Form Documents Based on Gradient-Wavelet Scheme" *Advances in Document Image Analysis* pp.240-254 1997

[72] Andy Yip, Chris Ding, and Tony Chan "Dynamic Cluster Formation Using Level Set Methods" *Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on* Volume 28, No. 6 June 2006 pp. 877-889

[73] T. Zhang, R. Ramakrishnan and M. Livny "Birch: An Effecient Data Clustering Method for Very Large Databases" In *Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal, Canada* 1996

VITA

Graduate College
University of Nevada, Las Vegas

Dean Patrick Curtis

Home Address:
6905 Wine River Drive, Las Vegas, Nevada 89119

Degrees:
Bachelor of Science, Computer Science, 2005
University of Nevada, Las Vegas

Special Honors and Awards:
Dean's Honor List, 2002, 2003, 2005

Captain of UNLV programming team that finished eighth in 2005

Best student poster at the Sixth International Meeting on Nuclear Applications of Accelerator Technology (AccApp 03), San Diego, CA, June 2003, American Nuclear Society

Publications:
Dean Curtis, Vitaliy Kubushyn, Chris Jones, E.A. Yfantis and Michael Rogers "Clustering of Medical Document Image Types using Iterative Feature Decomposition" *IEEE Pattern Analysis and Machine Intelligence, Transactions on* (submitted for publication).

Jia Tse, Chris Jones, Dean Curtis, Vitaliy Kubushyn, and E.A Yfantis "Shortest-Path Grayscale Character Segmentation in Document Image Analysis" *International Journal of Document Analysis and Recognition* (submitted for publication)

Dean Curtis, Vitaliy Kubushyn, E.A. Yfantis, and Michael Rogers "A Hierarchical Feature Decomposition Clustering Algorithm for Unsupervised Classification of Document Image Types" *IEEE* 6^{th} *International Conference on Machine Learning and Application* December 13-15, 2007 accepted for publication

Jia Tse, Dean Curtis, Christopher Jones, E.A. Yfantis and Brian Correia "An OCR-Independant Character Segmentation Using Shortest-Path in Grayscale Document Images" *IEEE* 6^{th} *International Conference on Machine Learning and Application* December 13-15, 2007 accepted for publication

John Bunch, Dean Curtis, Chris Jones, Jia Tse and E.A. Yfantis "Extracting the Major Form Body Segment from Unconstrained Document Images" *Proceedings of* *the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 75-80

Dean Curtis, Jia Tse, Christopher Jones, E.A. Yfantis, Scott Miller "Using Sequential Clustering for Unsupervised Classification of Unconstrained Document Images" *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 88-94

Christopher Jones, Vitaliy Kubushyn, John Bunch, Dean Curtis, E.A. Yfantis "Grouping, Segmentation and Recognition of Handwritten Social Security Images" *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 23-29

Jia Tse, Dean Curtis, John Bunch, Chris Jones, E.A. Yfanits and A. Thomas "Handwritten and Typewritten Word and Character Separation in Unconstrained Document Images" *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition* June 25-28, 2007 pp. 40-43

Dean Curtis, E.A. Yfantis, Jae Adams, C. Bellomo "Unconstrained Separation of Handwriting for Form Classification Applications" *44th ACM Southeast Conference* 2006 pp. 758-759

Dean Curtis, E.A. Yfantis, Jae Adams and Trenton Pack "Methods and Techniques in Handwritten form Recognition" *Information Science and Applications, WSEAS Tran. on* , Issue 3, Volume 3, March 2006, pp. 656-661

Jae Adams, E.A. Yfantis, Dean Curtis, and Trenton Pack "Feature Extraction Methods for Form Recognition Applications" *Information Science and Applications, WSEAS Tran. on* , Issue 3, Volume 3, March 2006, pp. 666-671

Dean Curtis, Denis Beller, Carter Hull, Alexander Rimsky-Korsakov, and Thomas Ward "Modeling Neutron Multiplicities in a 60-element 3He Detector System" *Proc. of the Sixth International Meeting on Nuclear Applications of Accelerator Technology (AccApp 03)* , San Diego, CA, June 2003, American Nuclear Society, pp. 190-194 (2004)

Thesis Title:
Unsupervised Learning of Document Image Types

Thesis Examination Committee:
Chairperson, Evangelos Yfantis, PhD.
Committee Member, Ajoy Datta, PhD.
Committee Member, John Minor, PhD.
Graduate Faculty Representitive, Angel Muleshkov, PhD.