



US 20090116746A1

(19) **United States**

(12) **Patent Application Publication**  
**Neogi et al.**

(10) **Pub. No.: US 2009/0116746 A1**

(43) **Pub. Date: May 7, 2009**

(54) **SYSTEMS AND METHODS FOR PARALLEL  
PROCESSING OF DOCUMENT  
RECOGNITION AND CLASSIFICATION  
USING EXTRACTED IMAGE AND TEXT  
FEATURES**

(75) Inventors: **Depankar Neogi**, Wilmington, MA  
(US); **Steven K. Ladd**, North  
Andover, MA (US); **Dilnawaj  
Ahmed**, Bangalore (IN); **Girish  
Welling**, Nashua, NH (US)

Correspondence Address:  
**WILMERHALE/BOSTON**  
**60 STATE STREET**  
**BOSTON, MA 02109 (US)**

(73) Assignee: **Copanion, Inc.**, Andover, MA (US)

(21) Appl. No.: **12/266,468**

(22) Filed: **Nov. 6, 2008**

**Related U.S. Application Data**

(60) Provisional application No. 60/985,851, filed on Nov.  
6, 2007.

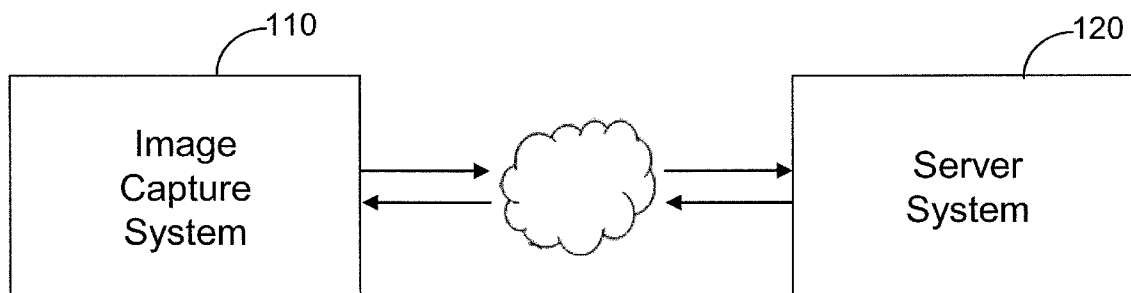
**Publication Classification**

(51) **Int. Cl.**  
**G06K 9/46** (2006.01)

(52) **U.S. Cl.** ..... **382/190; 382/224**

(57) **ABSTRACT**

A method of parallel processing jobs received from a plurality of users by a document analysis system that automatically classifies documents to organize each job, automatically separates each job into its constituent electronic document and automatically separate the document into subsets of electronic pages. For each page of each subset, the method automatically extracts image features that are indicative of how the document is laid out or textually-organized. For each subset, the method automatically compares the extracted features with feature sets associated with each document category to determine a comparison score for the subset. The method then classifies the electronic document as being one of the categories of documents using the comparison score for each of the subsets and organize the job according to the categories of documents the job contains.



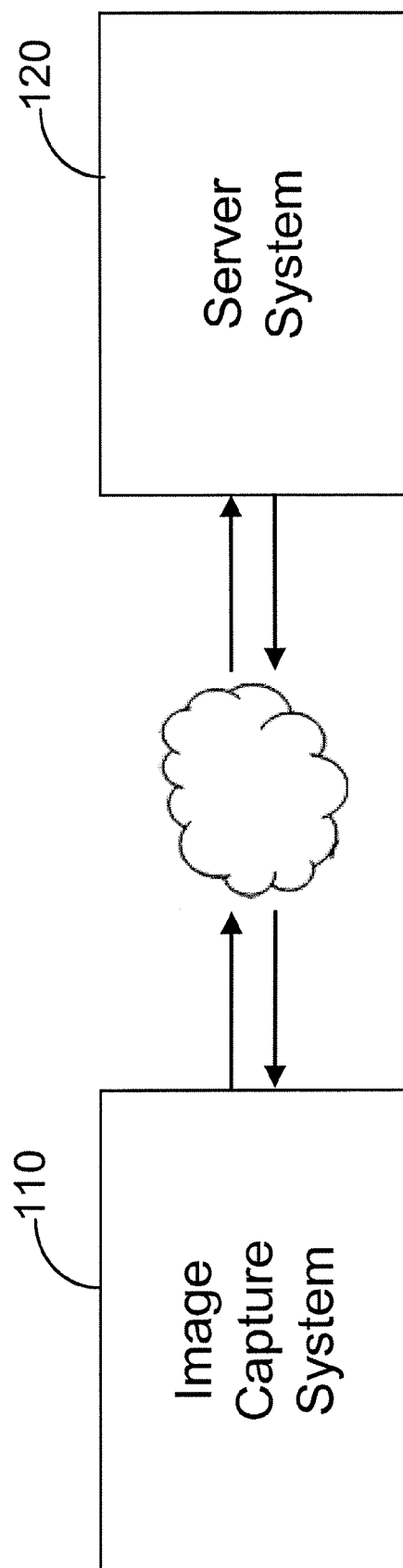
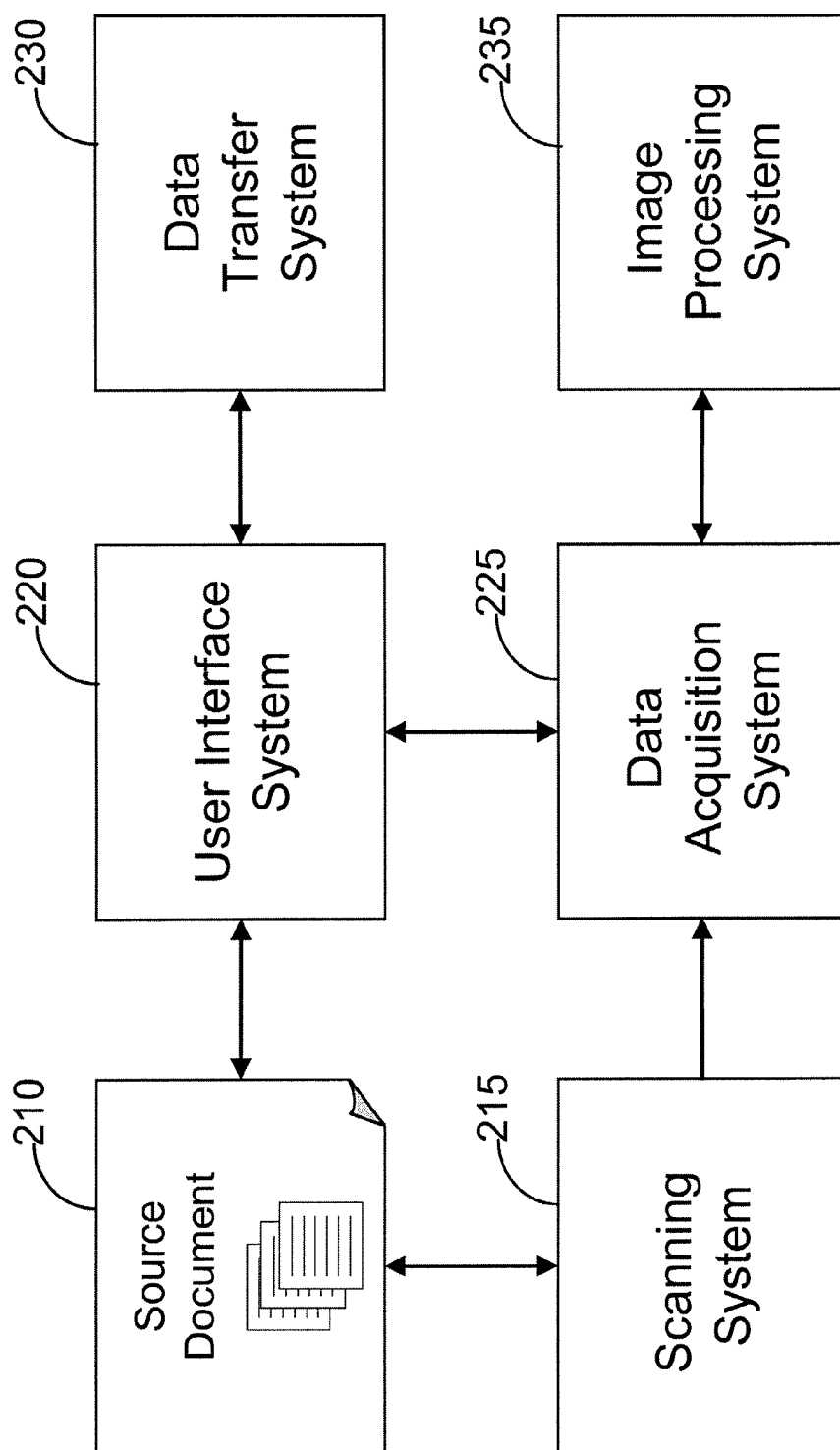


FIG. 1



**FIG. 2**

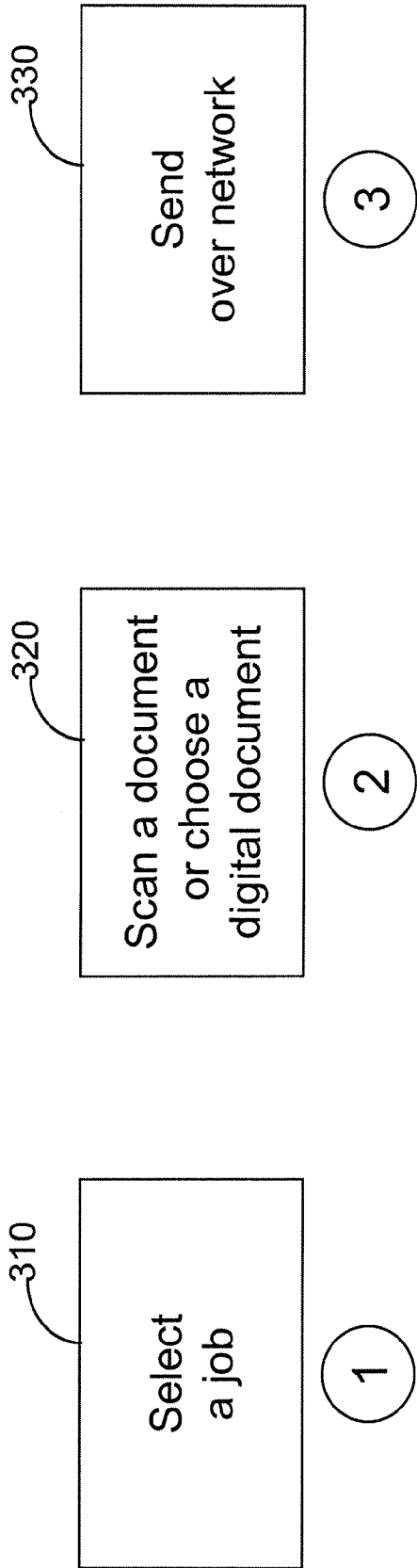
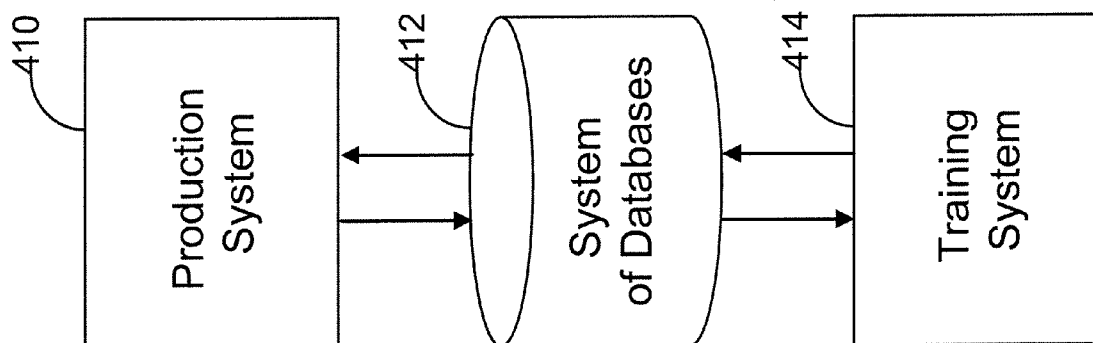


FIG. 3



**FIG. 4**

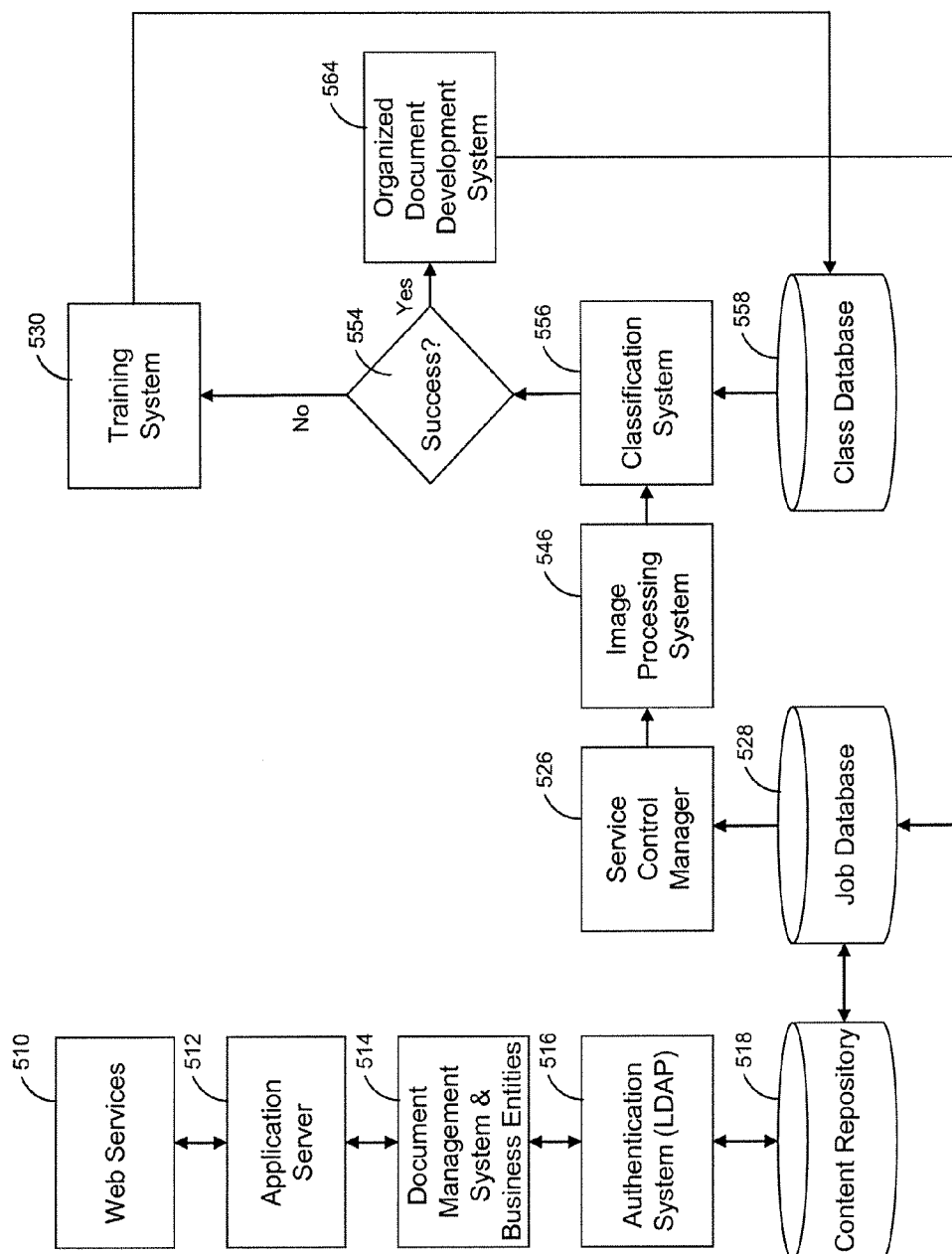


FIG. 5

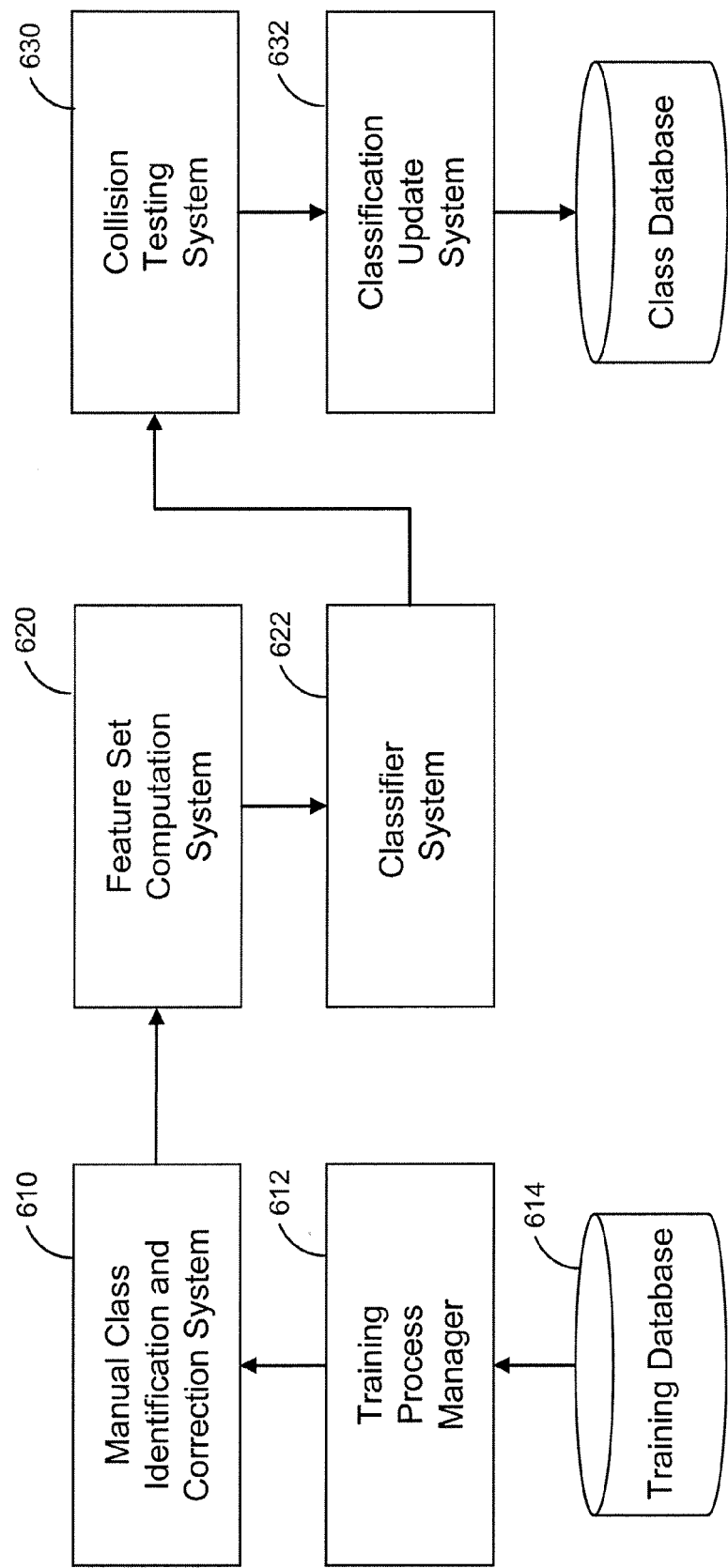


FIG. 6

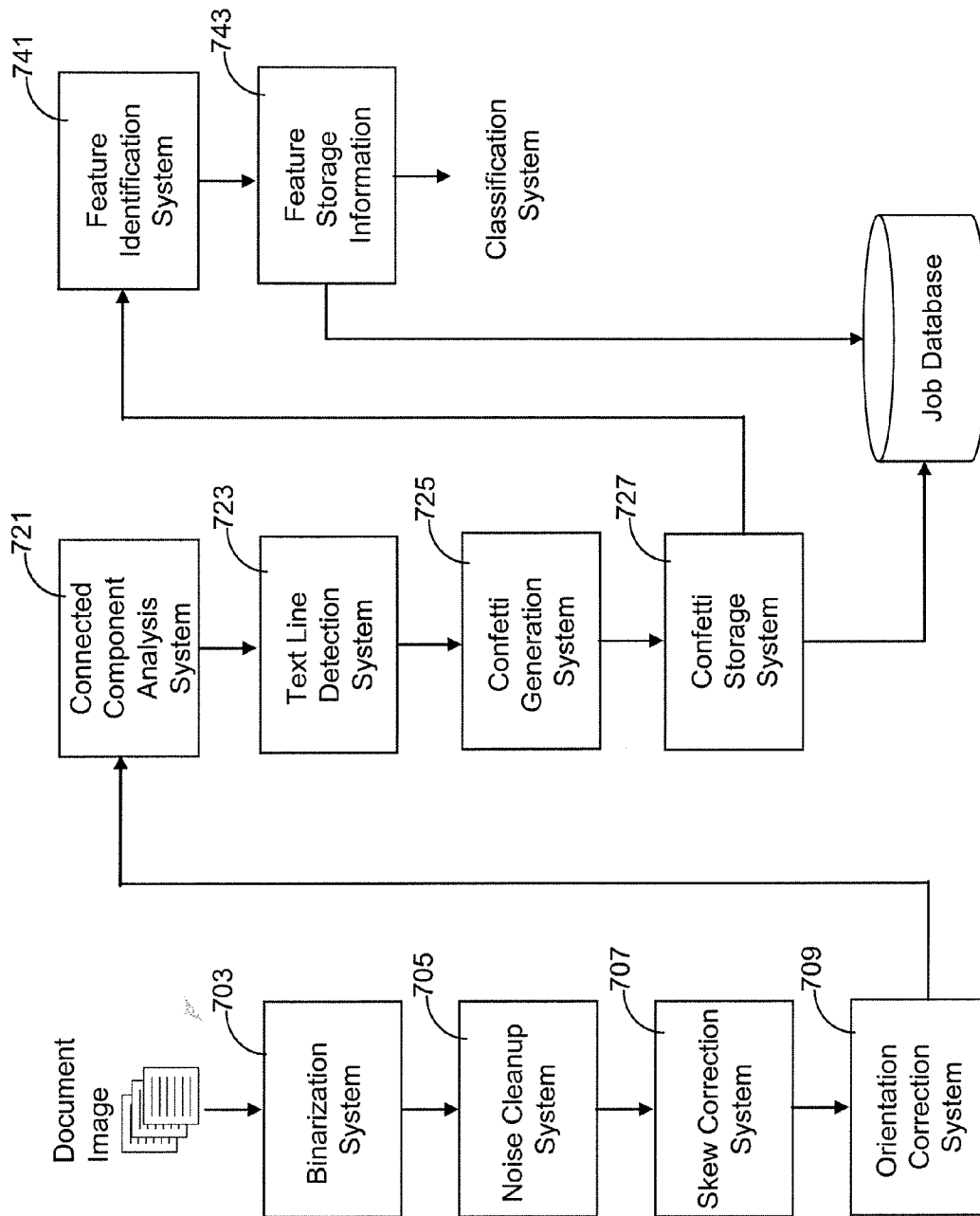


FIG. 7

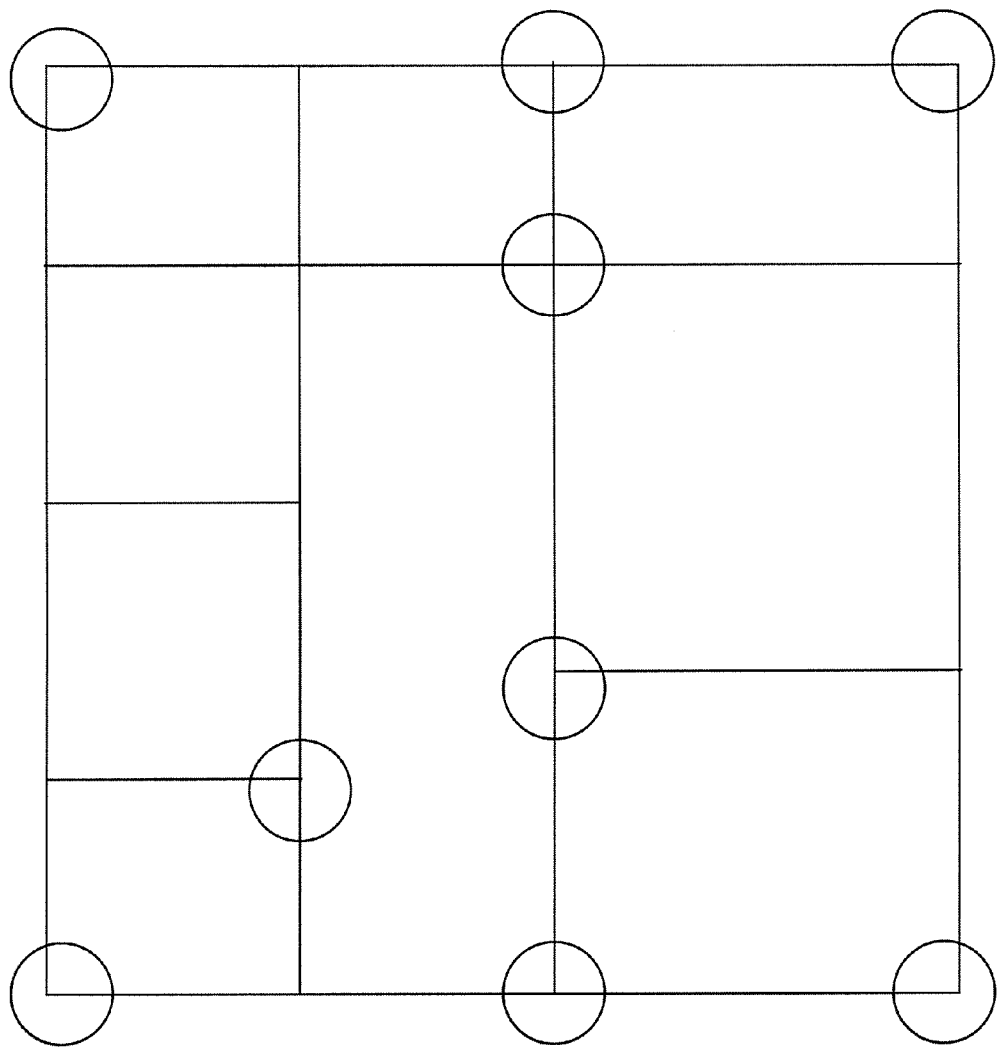


FIG. 8

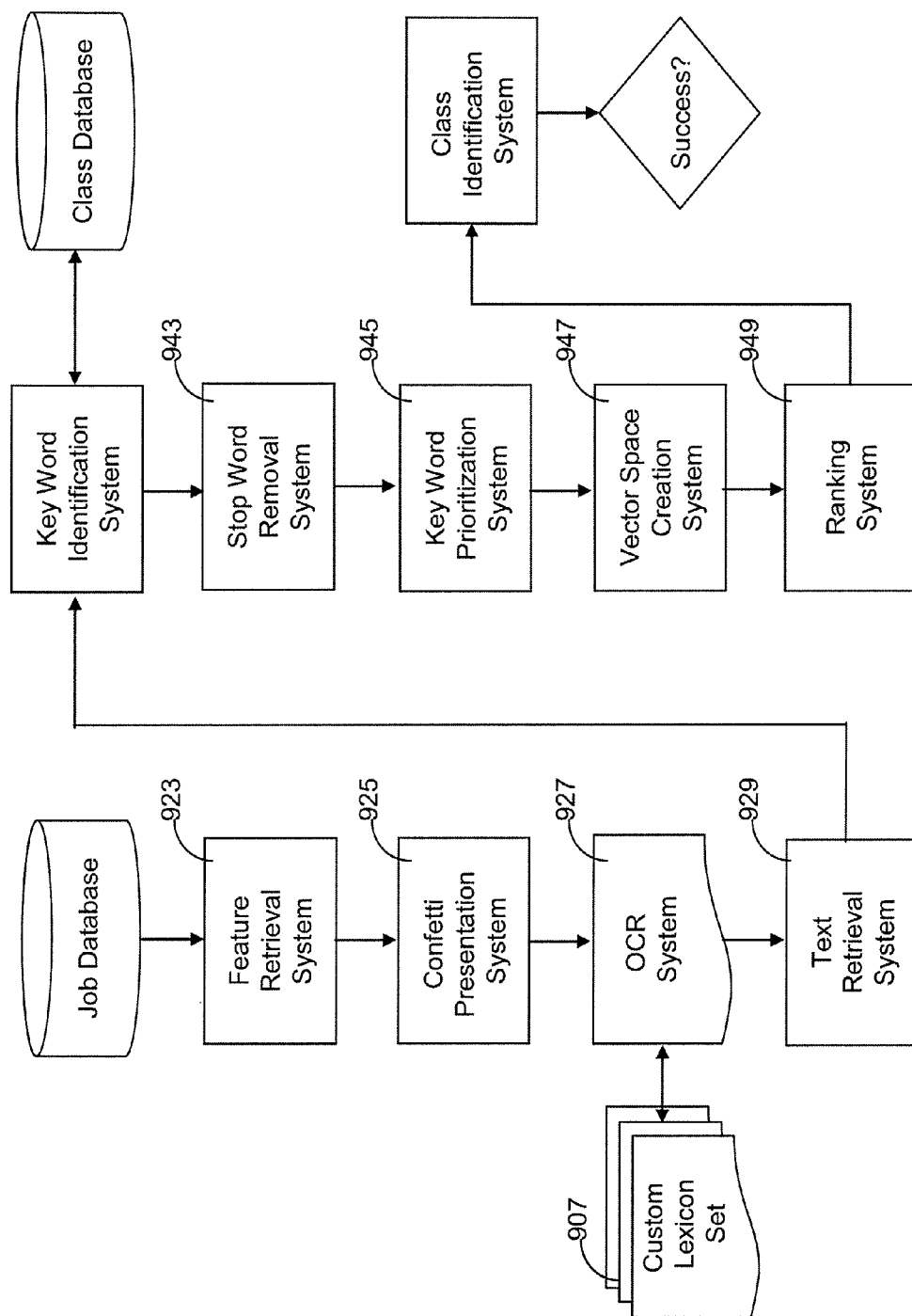


FIG. 9

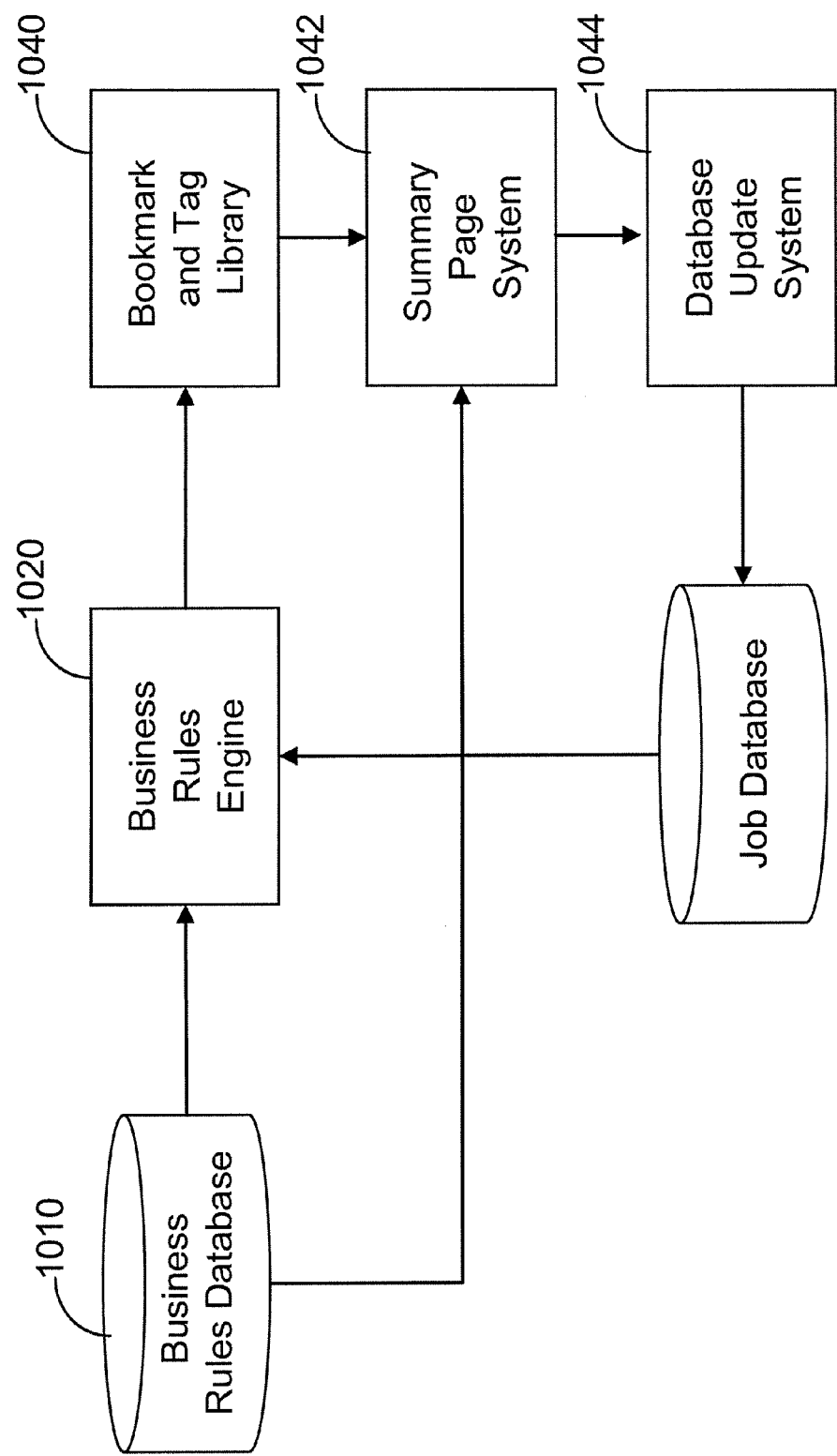
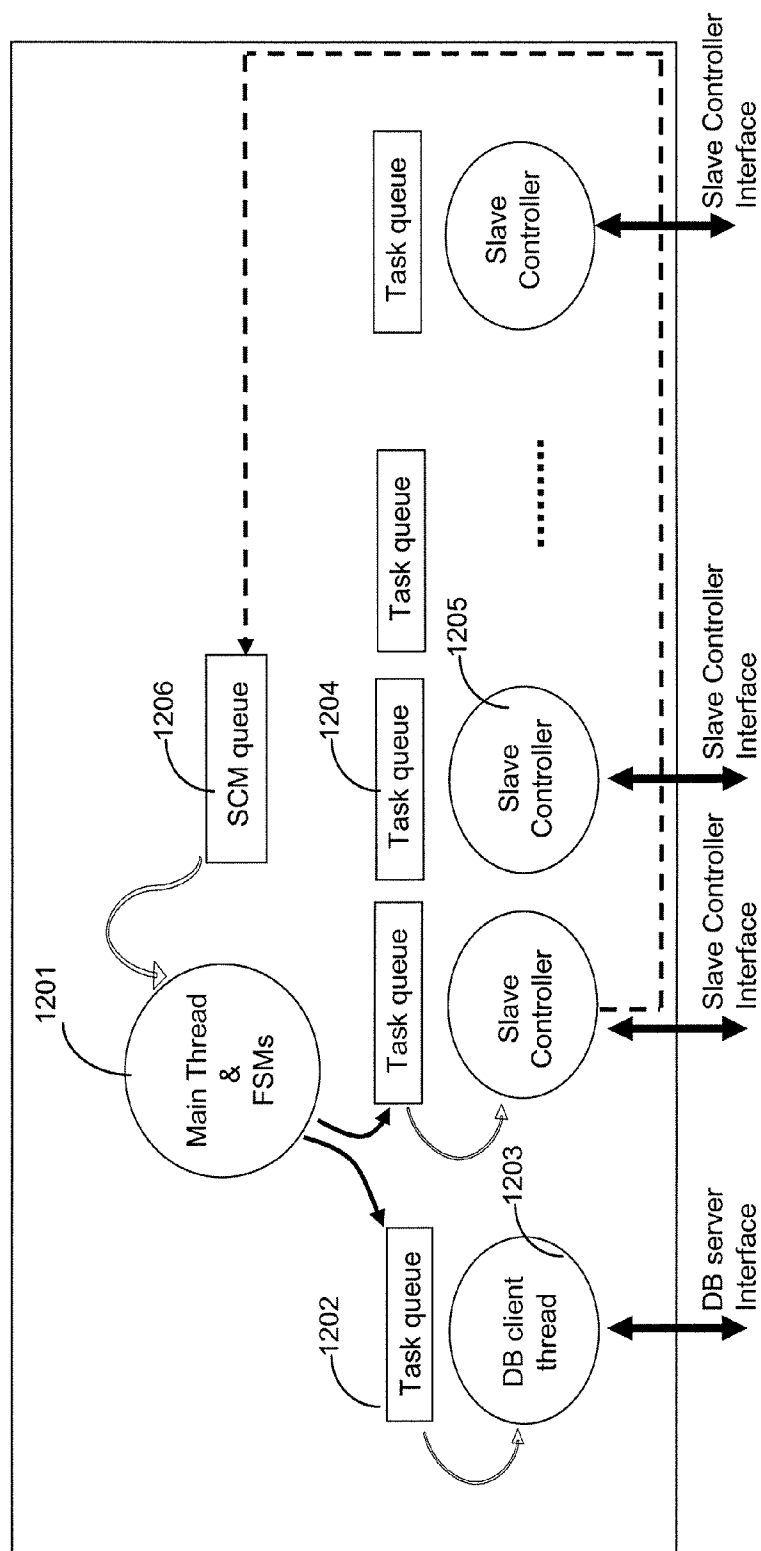


FIG. 10



**FIG. 11**



**FIG. 12**

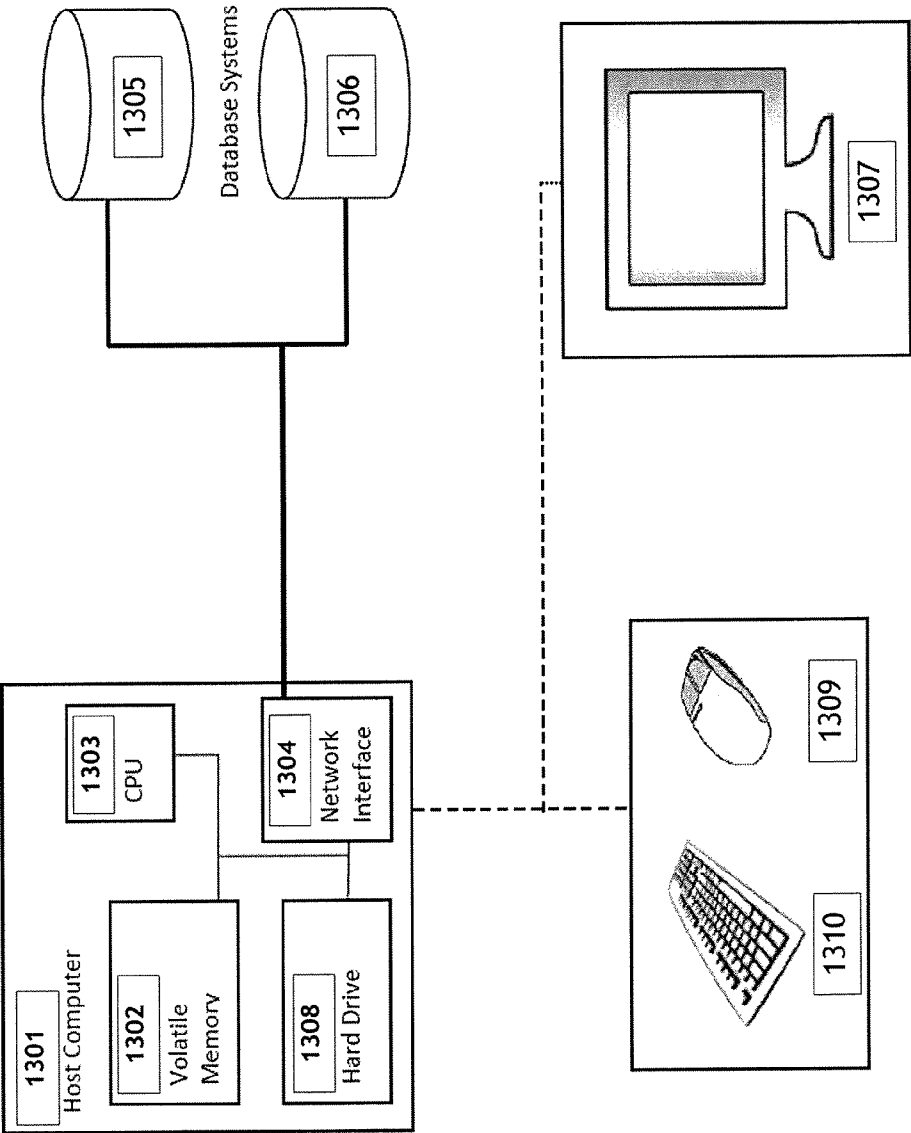


FIG. 13

# SYSTEMS AND METHODS FOR PARALLEL PROCESSING OF DOCUMENT RECOGNITION AND CLASSIFICATION USING EXTRACTED IMAGE AND TEXT FEATURES

## CROSS REFERENCE TO RELATED APPLICATIONS

**[0001]** This application claims the benefit under 35 U.S.C. §119(e) of U.S. Provisional Patent Application Ser. No. 60/985,851, filed on Nov. 6, 2007, which is hereby incorporated by reference herein its entirety.

**[0002]** This application is related to the following applications filed concurrently herewith, the entire contents of which are incorporated by reference:

**[0003]** U.S. Patent Application No. (TBA), entitled "Systems and Methods for Classifying Electronic Documents by Extracting and Recognizing Text and Image Features Indicative of Document Categories;"

**[0004]** U.S. Patent Application No. (TBA), entitled "Systems and Methods for Training a Document Classification System Using Documents from a Plurality of Users;"

**[0005]** U.S. Patent Application No. (TBA), entitled "Systems and Methods for Handling and Distinguishing Binarized, Background Artifacts in the Vicinity of Document Text and Image Features Indicative of a Document Category;"

**[0006]** U.S. Patent Application No. (TBA), entitled "Systems and Methods to Automatically Classify Electronic Documents Using Extracted Image and Text Features and Using a Machine Learning Subsystem;" and

**[0007]** U.S. Patent Application No. (TBA), entitled "Systems and Methods for Enabling Manual Classification of Unrecognized Documents to Complete Workflow for Electronic Jobs and to Assist Machine Learning of a Recognition System Using Automatically Extracted Features of Unrecognized Documents."

## BACKGROUND

**[0008]** Many software programs are currently available that allow a user to scan a number of paper documents and save them in a single electronic document. The electronic document is typically arranged as a sequence of individual pages. The software programs allow recipients to view, modify, print and store the electronic document. One example of such a document editing program is Adobe Acrobat from Adobe Systems Incorporated of San Jose, Calif.

**[0009]** In many instances, however, the paper documents are scanned in a random, unorganized sequence, which makes it difficult and time-consuming to find a particular page within the electronic document. One solution can be to manually organize the paper documents prior to scanning; however, the individual organizing the paper documents or performing the scanning may not have the skill, knowledge or time needed to correctly organize the paper documents. Additionally, organizing the paper documents prior to scanning can be very time-consuming and expensive. Further, organizing the pages prior to scanning might properly order the pages, but it does not generate a table of contents, metadata, bookmarks or a hierarchical index that would facilitate finding a particular page within the complete set of pages.

**[0010]** Ultimately, the recipient may want the pages of the electronic document organized in a specific order to facilitate finding a particular page in timely and inexpensive manner.

For example, an assistant may scan forty pages of tax documents in a random order and save the result in an electronic document. In this example, an accountant may then need to organize the pages of the electronic document in a specific order so that navigating through the electronic document during the preparation and review of an income tax return can be performed in an accurate and efficient manner.

**[0011]** One way that the recipient of an electronic document can organize the pages is by using the thumbnail, metadata and/or bookmark features of the document editing software program. Manually organizing an electronic document, including typing a table of contents, metadata, bookmarks or a hierarchical index, is time-consuming and expensive. Manual organization tends to be ad-hoc, failing to deliver a standardized table of contents, metadata, bookmarks or a hierarchical index for the electronic document.

**[0012]** Another way that the recipient of an electronic document can organize the pages is by using software that assists in manually categorizing document pages. The software provides a user a pre-identified set of types of documents and associates each page with the type selected by the user. This approach requires the recipient to manually categorize each page, a time-consuming and expensive process.

## SUMMARY OF THE INVENTION

**[0013]** Systems and methods for parallel processing of document recognition and classification using extracted image and text features are provided. In some embodiments, a method of parallel processing jobs that are received from a plurality of users by a document analysis system, which automatically classifies documents to organize each job according to the categories of documents the job contains, is provided. For each job, the method automatically separates the job into its constituent electronic documents it contains. For each received electronic document, the method automatically separates the document into subsets of electronic pages. For each page of each subset, the method automatically extracts image features that are indicative of how the document is laid out or textually-organized and therefore indicative of a corresponding document category and text features it contains, in which feature extraction for each subset is done independently and in parallel of such automatic extraction for the other subsets of the document. For each subset, the method automatically compares the extracted features with feature sets associated with each category of document to determine a comparison score for the subset. And using the comparison score for each of the subsets, the method automatically classifies the electronic document as being one of the categories of documents and organizes the job according to the categories of documents the job contains.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0014]** The invention is illustrated in the figures of the accompanying drawings which are meant to be exemplary and not limiting, in which like references are intended to refer to like or corresponding part, and in which:

**[0015]** FIG. 1 is a system diagram of a document identification and classification system 100 according to a preferred embodiment of the disclosed subject matter;

**[0016]** FIG. 2 is a system diagram of the image capture system 110 according to a preferred embodiment of the disclosed subject matter;

[0017] FIG. 3 is an illustration of three-step document submission process according to a preferred embodiment of the disclosed subject matter;

[0018] FIG. 4 is a system diagram of the server system 120 according to a preferred embodiment of the disclosed subject matter;

[0019] FIG. 5 is a system diagram of the production system 410 according to a preferred embodiment of the disclosed subject matter;

[0020] FIG. 6 is a system diagram of the training system 530 according to a preferred embodiment of the disclosed subject matter;

[0021] FIG. 7 is a system diagram of the image processing system 546 according to a preferred embodiment of the disclosed subject matter;

[0022] FIG. 8 is an illustration of an example of nine types of point patterns according to a preferred embodiment of the disclosed subject matter;

[0023] FIG. 9 is a system diagram of the classification system 556 according to a preferred embodiment of the disclosed subject matter;

[0024] FIG. 10 illustrates an organized document development system according to a preferred embodiment of the disclosed subject matter;

[0025] FIG. 11 illustrates an example of bookmarked document according to a preferred embodiment of the disclosed subject matter;

[0026] FIG. 12 is a flow diagram of the service control manager 526 according to a preferred embodiment of the disclosed subject matter; and

[0027] FIG. 13 illustrates an exemplary computer system on which the described invention may run according to a preferred embodiment of the disclosed subject matter.

#### DETAILED DESCRIPTION

[0028] While the prior art attempts to reduce the cost of electronic document organization through the use of software, none of the above methods of document organization (1) eliminates the human labor and accompanying requirements of education, domain expertise, training, and/or software knowledge, (2) minimizes time spent entering and quality checking page categorization, (3) minimizes errors and (4) protects the privacy of the owners of the data on the electronic documents being organized. What is needed, therefore, is a method of performing electronic document organization that overcomes the above-mentioned limitations and that includes the features enumerated above.

[0029] Preferred embodiments of the present invention provide a method and system for converting paper and digital documents into well-organized electronic documents that are indexed, searchable and editable. The resulting organized electronic documents support more rapid and accurate data entry, retrieval and review than randomly sequenced sets of pages.

[0030] FIG. 1 is a system diagram of a document identification and classification system 100 according to a preferred embodiment of the invention. System 100 has an image capture system 110 and a server system 120. The image capture system is connected to the production servers by a network such as a local-area network (LAN,) a wide-area network (WAN) or the Internet. The preferred implementation transfers all data over the network using Secure Sockets Layer (SSL) technology with enhanced 128-bit encryption. Encryption certificates can be purchased from well respected certifi-

cate authorities such as VeriSign and that or can be generated by using numerous key generation tools in the market today, many of which are available as open source. Alternatively, the files may be transferred over a non-secure network, albeit in a less secure manner.

[0031] Under typical operation, System 110 is an image capture system that receives physical documents and scans them. The image capture system is described in greater detail below.

[0032] Under typical operation, System 120 is a server system that receives the scanned documents over the Internet. Once received, the server system organizes the classified pages per a predetermined scheme into a new, organized document. The server system includes a mechanism for learning documents. The server system is described in greater detail below.

[0033] FIG. 2 is system diagram of the image capture system 110 according to a preferred embodiment of the invention. System 110 has a source document 210, a user interface system 220, a data transfer system 230, a scanning system 215, a data acquisition system 225 and an image processing system 235. Source documents in the form of papers are physically placed on an input tray of a commercial scanner. Source documents in the form of data files are received over a network by the user interface system. The user interface system communicates with the data transfer system via software within a computer system. The user interface system communicates with the data acquisition system via software within a computer system. The data acquisition system communicates with the scanning system via a physical connection, such as a high-speed Universal Serial Bus (USB) 2.0, or, optionally, over a network. The data acquisition system communicates with the image processing system via software within a computer system.

[0034] Element 210 is a source document in the form of either one or more physical papers or a digital file containing images of one or more papers. The digital file can be in one of many formats, such as PDF, TIFF, BMP, or JPEG.

[0035] System 220 is a user interface system. Under preferred embodiments, the user interface system runs in a browser and presents a user with a three-step means for submitting documents to be organized as shown in FIG. 3. In step one, the user interface system provides a mechanism for selecting a job from a list of jobs; additionally, it allows jobs to be added to the job list. In step two, the user interface system provides a mechanism for initiating the scanning of physical papers; additionally, it provides a browsing mechanism for selecting a file on a computer or network. Optionally, one or more sets of papers can be scanned and one or more files can be selected. In step three, the user interface system provides a mechanism for sending the job information and selected documents over a network to the server system. Under preferred embodiments, the user interface system also presents a user with the status of jobs that have been submitted as submitted or completed; optionally, it presents the expected completion date and time of submitted jobs that have not been completed. The user interface system also presents a user with a mechanism for receiving submitted documents and organized documents. The user interface system also provides a mechanism for deleting files from the system.

**[0036]** System **230** is a data transfer system. Under preferred embodiments, the data transfer system manages the SSL connection and associated data transfer with the server system.

**[0037]** System **215** is a scanning system. Under preferred embodiments, conventional scanning systems may be used such as those from Bell+Howell, Canon, Fujitsu, Kodak, Panasonic and Xerox. The scanning system captures an image of the scanned document as a computer file; the file is often in a standard format such as PDF, TIFF, BMP, or JPEG.

**[0038]** System **225** is a data acquisition system. The data acquisition system controls the settings of the scanning system. Many scanning systems in use today require users to manually set scanner settings so that images are captured, for example, at 300 dots per inch (dpi) as binary data (black-and-white.) Commercial scanners and scanning software modify the original source document image that often include high resolution and, possibly, color or gray-scale elements. The resolution is often reduced to limit file size. Color and gray-scale elements are often binarized, e.g. converted to black or white pixels, via a process known as thresholding, also to reduce file size. Under preferred embodiments, the data acquisition system sets the scanning system to scan pages double-sided at 300 dpi with eight bits of gray scale. The data acquisition system commands the scanning system to begin operation and receives the scanned document computer file from the scanning operation.

**[0039]** System **235** is an optional image processing system. The image processing system enhances the image quality of scanned images for a given resolution and other scanner settings. The image processing system may be implemented as part of the image capture system as depicted on FIG. 2 or as part of the server system as depicted on FIG. 8. Details of the image processing system are described in further detail below as part of the server system.

**[0040]** FIG. 4 is a system diagram of the server system **120** according to a preferred embodiment of the invention. System **120** has a production system **410**, a system of databases **412** and a training system **414**. Under preferred embodiments, the production system is connected to the system of databases by a gigabit Ethernet connection. Under preferred embodiments, the training system is connected to the system of databases by a gigabit Ethernet connection.

**[0041]** The production system classifies each of the pages in the document as one of a pre-identified set of types of documents. The production system organizes the classified pages per a predetermined scheme into a new, organized document. The production system stores the original scanned document and the organized document. The production system is described in greater detail below.

**[0042]** The system of databases is comprised of a content repository, a job database, a class database and a training database. The system of databases is described in greater detail below.

**[0043]** The training system utilizes supervised learning to provide a growing set of documents with characterized feature sets to the class database. The training system is described in greater detail below.

**[0044]** FIG. 5 is a system diagram of the production system. System **410** has a web services system **510**, an application server **512**, a document management system and business entities (DMS) **514**, a LDAP authentication system **516**, a content repository **518**, a job database **528**, a service control manager **526**, an image processing system **546**, a classifica-

tion system **556**, a success evaluation system **554**, an organized document development system **564**, a training database **532**, a training system **530** and a class database **558**.

**[0045]** FIG. 5 shows the overall system that represents the server-side operation of the automatic organization of electronic documents. The system is comprised of several modules that help in automating and improving the accuracy of the results. The system may be built in a highly distributed architecture and consists of several independent processes, data repositories and databases which communicate and pass messages to each other via well defined standard and proprietary interfaces. Even though the system may be built in a loosely coupled manner to achieve maximum scalability and throughput, the same results can be achieved if the system was more tightly coupled in a single process with each module being a logical entity of the same process. Furthermore, the design of the system considers multiple different product types which may need to process anywhere from hundreds to millions of documents every day for tens to thousands of customers in different markets. Another advantage of the above system design is that it allows the server(s) to be hosted in a secure data center. Documents from healthcare, insurance, banking, government, tax and other applications which will go through the recognition and organization processing system will need security applied per policies that are HIPAA, GLBA, SAS70, etc. compliant.

**[0046]** The web services **510** system provides the server system connection to the network that interfaces with the image capture system. Such a network could be a local-area network (LAN), a wide-area network (WAN) or the Internet. As described above, the preferred implementation transfers all data over the network using Secure Sockets Layer (SSL) technology with enhanced 128-bit encryption. Standard web services include Apache, RedHat JBoss Web Server, Microsoft IIS, Sun Java System Web Server, IBM Websphere, etc. The primary web service requirement is that the module should be able to handle multiple HTTP or HTTPS requests from different users as they upload their source electronic documents or download their organized electronic document, in a secure manner. The web service should also be able to relay any necessary parameters to the application servers which will process the electronic document.

**[0047]** The application server **512** provides necessary clustering, caching, load-balancing and persistence of the application for a distributed deployment of large scalable enterprise applications. The application layer manages transaction context as documents are uploaded and downloaded to the system and maintains all necessary service integrity. The application server also provides messaging services, mail services, security services, and connection pool, all of which make the service available to handle a large number of requests simultaneously.

**[0048]** The document management system (DMS) **514** and the business object layer capture the business entities. The DMS is generally a computer-based system or set of servers used to track and store electronic documents and/or images of paper documents. The DMS also commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities. Simple functions could include adding or retrieving an electronic document of a user. System **514** handles complex business hierarchies for a large number of users across multiple organizations. This is achieved by designing appropriate business objects that access the data access objects (DAO) with appropriate privileges and permis-

sions. The data access objects implement the access mechanism to the data sources. The data source could be a persistent store like an RDBMS, an external service like a B2B exchange, a repository such as the LDAP database of System 516, or a business service accessed via CORBA Internet Inter-ORB Protocol (IIOP) or low-level sockets. The business component that relies on the DAO uses the simpler interface exposed by the DAO for its clients. The DAO completely hides the data source implementation details from its clients. Because the interface exposed by the DAO to clients does not change when the underlying data source implementation changes, this pattern allows the DAO to adapt to different storage schemes without affecting its clients or business components. Essentially, the DAO acts as an adapter between the component and the data source.

[0049] The authentication system 516 allows secure and authorized access to the content repository. Under preferred embodiments, an LDAP authentication system is used; however, other authentication systems can also be used. In general, an LDAP server is used to process queries and updates to an LDAP information directory. For example, a company could store all of the following very efficiently in an LDAP directory:

[0050] The company employee phone book and organizational chart

[0051] External customer contact information

[0052] Infrastructure services information, including NIS maps, email aliases, and so on

[0053] Configuration information for distributed software packages

[0054] Public certificates and security keys.

[0055] Under a preferred embodiment, document organization and access rights are managed by the access control privileges stored in the LDAP repository.

[0056] The content repository 518 can be simple file system, a relational database or an object oriented database. Under a preferred embodiment, the content repository is based on Java Standard Requests 170 (JSR 170). JSR 170 is a standard implementation-independent way to access content bi-directionally on a granular level within a content repository. The content repository is a generic application "data store" that can be used for storing both text and binary data (images, word processor documents, PDFs, etc.). One key feature of a content repository is that one does not have to worry about how the data is actually stored: data could be stored in a relational database (RDBMS) or a file system or as an XML document. In addition to providing services for storing and retrieving the data, most content repositories provide advanced services such as uniform access control, searching, versioning, observation, locking, and more.

[0057] Under preferred embodiments, documents in the content repository are available to the end user via a portal. For example, in the current implementation of the system, the user can click on a web browser application button "View Source Document" in the portal and view the original scanned document over a secure network. Essentially, the content repository can become an off-site secure storage facility for the user's electronic documents.

[0058] The job database 528 is used to receive, then process and finally post the user's job back to the content repository. A "job" is defined as the steps of automatically organizing the electronic document from their original scanned images. Module 649 can be file system storage, a relational database, XML document or a combination of these. In the current

implementation, the system uses both file system storage to store large blob (binary large objects) and a relational database to store pointers to the blobs and other information pertinent to processing the job.

[0059] The service control manager (SCM) 526 is a system that controls the state machine for each job. The state machine identifies the different states and the steps that a job has to progress through to achieve its final objective, in this case being an organized electronic document. In the current system, the SCM is designed to be highly scalable and distributed. Under preferred embodiments, the SCM is multi-threaded to handle hundreds of jobs at any given time. It also implements message queues to communicate with other processes regarding their own states. The SCM can be implemented in other architectures as well. For example, one can implement a complete database driven approach to step through all the different steps required to process such a job.

[0060] In preferred implementations the SCM subscribes to events for each new incoming job that need to be processed. Once a new job arrives, the SCM pre-processes the job by taking the electronic document and separating each image (or page) into its own bitmap image for further processing. For example, if an electronic document had 30 pages, the system will create 30 images for processing. Each job in the system is given a unique identity. Furthermore, each page is given a unique page identity that is linked to the job identity. After the SCM has created image files by pre-processing the document into individual pages, it transitions the state of each page to image processing.

[0061] The image processing system 546 removes noise from the page image and properly orients the page so that document image analysis can be performed more accurately. The accuracy of the document recognition greatly depends on the quality of the image; thus image processing is included under preferred embodiments. The image processing system performs connected component analysis and, utilizing a line detection system, creates "confetti" images which are small sections of the complete page image. Under preferred embodiments, the confetti images are accompanied the coordinates of the image sub-section. The image processing system is discussed in greater detail below.

[0062] The classification system 556 recognizes the page as one of a pre-identified set of types of documents. A major difficulty in categorizing a page as one of a large number of documents is the high dimensionality of the feature space. Conventional approaches that depend on text categorization alone are faced with a native feature space that consists of many unique terms (words as well as phrases) that occur in documents, which can be hundreds or thousands of terms for even a moderate-sized collection of unique documents. In one domain, multiple systems that categorize income tax documents such as W-2, 1099-INT, K-1 and other forms have experienced poor accuracy because of the thousands of variations of tax documents. The preferred implementation uses a combination of image pattern recognition and text analysis to distinguish documents and machine learning technology to scale to large numbers of documents. The classification system is described in greater detail below.

[0063] The class database 558 contains the trained set of information produced and used by the systems learning engine. As the system grows "smarter" by recognizing more classes and variations of documents, the class database grows. As the machine learning system sees more trained documents, its classification accuracy increases.

[0064] The success evaluation system 554 determines how the document is treated once the classification process has been completed. If the classification system successfully classifies the document, the document is directed to System 564, the organized document development system described below. If the classification system fails to recognize the document with a high level of confidence, the document is directed to System 530, a training system, described below.

[0065] The training system 530 performs computations on the data in its document database corresponding to the classification systems that are in place and generates datasets used by the classification system for recognizing source documents. The results of the training and re-training process are classification datasets that are updated in the class database. The training system is described in greater detail below.

[0066] Thus, the system implements a continuous learning process in which a document that is not accurately identified is sent for training. Training results in an expanded data set in the class database 558, thereby improving the accuracy of the system over time. As the class database grows, the system requires an asymptotically lower percentage of documents to be trained.

[0067] FIG. 6 is a system diagram of the training system. System 530 has a manual class identification and correction system, a feature set computation system, a collision testing system, a training process manager, a classifier system, and a classification update system.

[0068] Preferred implementations use machine learning supported by the training system that adapts to a growing set of documents. Additional documents add additional features that must be analyzed. Preferred implementations of the training system include tuning and optimization to handle noise generated during both the training phase and the testing phase. The training phase is also called learning phase since the parameters and weights are tuned to improve the learning and adaptability of the system by fitting the model that minimizes the error function of the dataset.

[0069] The learning technique in the preferred implementation is supervised learning. Applications in which training data comprises examples of input vectors along with their corresponding target vectors are known as supervised learning problems. Example input vectors include key words and line patterns of the document layouts. Example target vectors include possible classes of output in the organized document. Supervised learning avoids the unstable states that can be reached by unsupervised learning and reinforcement learning systems.

[0070] The learning system receives documents into a training database 614 from the success evaluation system. These documents are not trained and do not have corresponding classification model data in the class database. All such documents are made persistent in the training database.

[0071] The training database 614 has several tables which contain the document class information as well as image processing information (which is discussed in greater detail below.) The following tables are part of training database:

[0072] Form class (classification view)

[0073] Page table (details of the page of the electronic document)

[0074] Manual classification table (manual work information)

[0075] Manual training table (trainers' information)

[0076] Confetti table (confetti information, original text, corrected text, etc.).

[0077] The training process manager 612 is a system that manages the distribution of the training task. Under preferred embodiments, a user, called a "trainer," logs into the system in which the trainer has privileges at one of three levels.

[0078] Trainer levels:

[0079] Top tier: add new classes to the system and perform classification and training

[0080] Middle tier: perform manual classification and training

[0081] Bottom tier: only perform training (manual text correction).

[0082] The training process manager directs document processing based on the document state:

[0083] Unclassified page is scheduled for manual classification

[0084] Manual classification is done as per policy and form class is assigned

[0085] Job database is updated with form class information and page/job states are changed so that the page can go to next state

[0086] If the form class state is not trained, the form is scheduled for training, else no action is needed.

[0087] After form training, the form class state is changed to trained, not synched if allowed by policy. The document class has the following states:

[0088] Untrained

[0089] Partially trained

[0090] Trained, need synch with classification database

[0091] Trained, synched with classification database.

[0092] The manual identification and text correction system processes each document that requires training. The trainer follows two independent steps:

[0093] Manually classifying the form and assigning a class and subclass

[0094] Manually correcting OCR text (name required training for now).

[0095] The manual identification and text correction system 610 is comprised of a number of elements:

[0096] Receive pages from the training manager which manages the flow of pages between various trainers and implements training policy and restrictions

[0097] Manual classification user interface (UI) which presents the page and asks the user to classify it

[0098] Manual text correction UI which presents the page with marked up confetti. The user views the confetti and corrects the text extracted from the confetti

[0099] Training viewer UI is used to view the training database in an UI. The preferred implementation includes reports and representations of the training database

[0100] Classification verification UI presents a page and its classification to a trainer.

[0101] All user interfaces are integrated into a single system.

[0102] The feature set computation system 620 combines the document image, the manually classified information and the corresponding text.

[0103] The feature set computation system is the point-pattern matching data which is described in greater detail below.

[0104] The classifier system 622 creates a global dictionary (GD) and global priority (GPr) of a word based on the words

it receives from each document after OCR correction. Global Priority is formulated as:

$$GPr = MPr * \log(\text{total no. of classes} / \text{total no. classes in which the word appears})$$

where GPr is the global priority of the word and MPr is the mean priority of the word among the classes it is presented. For example, if word1 is present in class1 with priority p1, in class2 with priority p2, in class3 with priority p3, then:

$$MPr = (p1 + p2 + p3) / 3$$

If in total there are Nc classes, then:

$$GPr \text{ of word1} = (p1 + p2 + p3) / 3 * \log(Nc / 3).$$

[0105] Thus, we take inverse word frequency as log (total no of classes/total no classes in which the word appears). So that if a particular word is present in almost all classes then its priority will be low.

[0106] With the value of the priority, a particular word can be determined to be a stop word or not. The preferred implementation stores all the word in a Trie data structure to maximize search speed. With all the words in the global dictionary, a trellis of bi grams of letters is built which is used during prediction of a letter during document OCR.

[0107] Below is an example of 2 different classes (W-2 and 1099-INT) and their priorities and key words:

Form Name: 1099-INT

[0108]

|              |          |
|--------------|----------|
| payer        | 0.666667 |
| name         | 0.333333 |
| address      | 0.333333 |
| account      | 0.333333 |
| deferrals    | 0.333333 |
| income       | 0.333333 |
| rents        | 0.333333 |
| royalties    | 0.333333 |
| omb          | 0.666667 |
| federal      | 0.333333 |
| compensation | 0.333333 |
| tax          | 0.333333 |
| parachute    | 0.333333 |
| misc         | 1        |
| substitute   | 0.333333 |
| recipient    | 0.666667 |
| int          | 1        |
| foreign      | 0.666667 |
| with drawl   | 0.333333 |
| rtn          | 0.666667 |
| penalty      | 0.333333 |
| investment   | 0.333333 |

Form Name: W2

[0109]

|           |          |
|-----------|----------|
| employer  | 0.666667 |
| wages     | 1        |
| employee  | 0.333333 |
| social    | 0.333333 |
| security  | 0.333333 |
| dependent | 0.333333 |
| federal   | 0.333333 |
| name      | 0.333333 |

-continued

|              |          |
|--------------|----------|
| address      | 0.333333 |
| compensation | 0.666667 |
| tax          | 0.333333 |
| omb          | 0.666667 |
| income       | 0.333333 |

The global dictionary and global priorities for the above example are:

|              |          |
|--------------|----------|
| payer        | 0.462098 |
| name         | 0        |
| address      | 0        |
| account      | 0.231049 |
| deferrals    | 0.231049 |
| income       | 0        |
| rents        | 0.231049 |
| royalties    | 0.231049 |
| omb          | 0        |
| federal      | 0        |
| compensation | 0        |
| tax          | 0        |
| parachute    | 0.231049 |
| misc         | 0.693147 |
| substitute   | 0.231049 |
| recipient    | 0.462098 |
| int          | 0.693147 |
| foreign      | 0.462098 |
| withdrawal   | 0.231049 |
| rtn          | 0.462098 |
| penalty      | 0.231049 |
| investment   | 0.231049 |
| employer     | 0.462098 |
| wages        | 0.693147 |
| employee     | 0.231049 |
| social       | 0.231049 |
| security     | 0.231049 |
| dependent    | 0.231049 |

[0110] The above table shows that words like “omb,” “name,” “address,” etc have priorities zero as they are present in all the documents; hence, they are stop words. Words that occur more frequently in all the forms have less priority.

[0111] The collision testing system 630 performs tests across a large set of trained documents to ensure that previously trained documents do not collide or break with the newly added information. Under preferred embodiments, a large regression suite is built as part of the collision testing system.

[0112] The classification update system 632 inserts new trained data that passes regression testing into the class database.

[0113] FIG. 7 is a system diagram of the image processing system. System 546 has a binarization system, a noise cleanup system, a skew correction system, an orientation correction system, a connected component analysis system, a text line detection system, a confetti generation system, a confetti storage system, a feature identification system and a feature storage system.

[0114] Source document images can have salt-pepper noise, skew, orientation in any direction and/or color or gray scale elements. A document can be captured as a color, gray-scale or binary image by a scanning device. Common problems seen in images from scanning devices include:

[0115] poor contrast due to lack of sufficient or controllable lighting

[0116] non-uniform image background intensity due to uneven illumination

[0117] immoderate amount of random noises due to limited sensitivity of the sensors.

[0118] Many document images are rich in color and have complex backgrounds. Accurately processing such documents typically requires time-consuming processing and manual tuning of various parameters. Detecting text in such documents, which is necessary for text analysis, is difficult for typical text recognition systems that are optimized for binary images on clean backgrounds. For the classification system to work well, document images must be binarized and the text must be readable. Typically, general purpose scanners binarize images using global thresholding utilizing a single threshold value, generally chosen on statistics of the global image. Global thresholding is not adapted well for images that suffer from common illumination or noise problems. Global thresholding often results in characters that are broken, merged or degraded; further, thousands of connected components can be caused by binarization noise. Images degraded by global thresholding are typically candidates for low quality pattern recognition and text analysis.

[0119] The preferred embodiment of the binarization system utilizes local thresholding where the threshold value varies based on the local content in the document image. The preferred implementation is built on a adaptive thresholding technique which exploits local image contrast (reference: *IEICE Electronics Express*, Vol. 1, No 16, pp. 501-506.) The adaptive nature of this technique is based on flexible weights that are computed based on local mean and standard deviations calculated for the gray values in the primary local zone or window. The preferred embodiment experimentally determines optimum median filters across a large set of document images for each application space.

[0120] The noise cleanup system removes dots, specks and blobs from documents. In the preferred embodiment, minimum and maximum dot size are specified. The preferred embodiment also performs image reversal so that white text or line objects on black backgrounds are detected and inverted to black-on-white. The preferred embodiment also performs two noise removal techniques.

[0121] The first technique starts with any small region of a binary image. The preferred implementation takes a 35×35 pixel region. In this region all background pixels are assigned value "0." Pixels adjacent to background are given value "1." A matrix is developed in this manner. In effect each pixel is given a value called the "distance transform" equal to its distance from the closest background pixel. The preferred implementation runs a smoothing technique on this distance transform. Smoothing is a process by which data points are averaged with their neighbors in a series. This typically has the effect of blurring the sharp edges in the smoothed data. Smoothing is sometimes referred to as filtering, because smoothing has the effect of suppressing high frequency signals and enhancing low frequency signals. Of the many different methods of smoothing, the preferred implementation uses a Gaussian kernel. In particular, the preferred implementation performs Gaussian smoothening with a filter using variance of 0.5 and a 3×3 kernel or convolution mask on the distance transform. Thresholding with a thresholding value of 0.85 is performed on the convolved images and the resulting data is converted to its binary space. This method has been tested across a large number of noisy documents and the results have been found to be good.

[0122] The second technique uses connected component analysis (discussed in greater detail below) to identify small or bad blocks. In this method a sliding mask is created of a known size. The preferred implementation uses a mask that is 35×35 pixels wide. This mask slides over the entire image and is used to detect the number of blobs (connected components) that are less than 10 pixels in size. If the number of blobs is greater than five, then all blobs are removed. This process is repeated by sliding the mask over the entire image.

[0123] The skew correction system fixes small angular rotations of the entire document image. Skew correction is important for the document analysis module because it improves text recognition, simplifies interpretation of page layout, improves baseline determination, and improves visual appearance of the final document. Several available image processing libraries do skew correction. The preferred implementation of skew detection is part of the open source Lep-tonica image processing library.

[0124] The orientation correction system aligns document images so that they can be most easily read. Documents, originally in either portrait or landscape format may be rotated by 0, 90, 180 or 270 degrees during scanning. There are three preferred implementations of orientation correction.

[0125] The first method detects blocks of text in the image and measures each with respect their block height and width. In portrait documents, the average width is more than average height. An average count of the width and height is performed and if the width to height ratio is above a certain threshold, the document is determined to be portrait or landscape.

[0126] The second method performs a baseline analysis, counting the pixels in ascenders and descenders along any line in a document. Heuristically, the number of ascenders is found to be more than the number of descenders in English language documents that are correctly oriented. The document is oriented so that ascenders outnumber descenders.

[0127] The third method performs OCR is on small words or phrase images at all four orientations: 0, 90, 180 and 270 degrees. Small samples are selected from a document and the confidence is averaged across the sample. The orientation that has the highest confidence determines the correct orientation of the document.

[0128] The connected component analysis system implements a very standard technique. In the preferred implementation the open source Image Processing Library 98 (IPL98) is used for connected component analysis.

[0129] The text line detection system implements a technique described by Okun et al. (reference: "Robust Text Detection from Binarized Document Images") to identify candidate text segments blocks of consistent heights. For a page from a book, this method may identify a whole line as a block, while for a form with many boxes this method will identify the text in each box.

[0130] The confetti generation module identifies all the coordinates of the blocks.

[0131] The confetti storage system stores the confetti information in the appropriate persistent storage.

[0132] The job database stores confetti information for each job.

[0133] The feature identification system looks for point and line features. The preferred implementation performs image layout analysis using two image properties, the point of intersection of lines and edge points, as shown in FIG. 8, of text paragraphs. Every unique representation of points is referred as a unique class in the system and represents a unique point

pattern in the system database. The preferred implementation uses a heuristically developed convolution method only on black pixels to perform a faster computation. The system identifies nine types of points: four T's, four L's, and one cross (X) using nine masks.

**[0134]** The preferred implementation of point pattern matching is performed by creating a string from the points detected in the image and then using the Levenshtein distance to measure the gap between the trained set with the input image. The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

**[0135]** The feature storage system saves the calculated features for each document.

**[0136]** FIG. 9 is a system diagram of the classification system. System 556 has a custom lexicon set 907, a feature retrieval system 923, a confetti presentation system 925, an OCR system 927, a text retrieval system 929, a key word identification system 941, a stop word removal system 943, a key word prioritization system 945, a vector space creation system 947, a ranking system 949 and a class identification system 965.

**[0137]** The preferred implementation performs classification using both image-level features and textual features. The key challenge in classification architecture is defining the classifier appropriate to the domain. Many forms and documents, such as business letters, tax forms, mortgage applications, health insurance forms, etc., have structural layouts and associated text, each of which have important domain information.

**[0138]** The feature retrieval system receives confetti images and point pattern image features (discussed above) for a document from the job database and presents them to the OCR system. Several optical character recognition (OCR) software programs are available in the market today. The preferred implementation uses Tesseract, an open source software which allows custom modifications. In the preferred implementation, a custom domain-specific lexicon set has been added to the system to improve the accuracy of the system. During training of the document for a particular domain, several key words are collected that are made part of the custom lexicon set of that domain classifier.

**[0139]** The OCR system converts each confetti image into text.

**[0140]** The text retrieval system presents the text to the key word identification system.

**[0141]** The key word identification system receives the confetti text and interfaces with the class database. The class database consists of the global dictionary and global priority words which are created by the training process and the point pattern signatures of all the trained forms.

**[0142]** Under the preferred embodiment, the stop word removal system removes stop words from the list of text that was received from the text retrieval system. Stop words are common words—for example: “a,” “the,” “it,” “not,” and, from the W-2 and 1099-INT example above, words including “omb,” “name,” “address,” etc. The stop words are provided by the class database and, in the preferred embodiment, are domain specific.

**[0143]** The key word prioritization system, in the preferred implementation, calculates the priority of each word as function of line height (LnHt) of the word, partial of full match

(PFM) with form name and total number of words in that form (N). The approximate value of priority is formulated as

$$Pr = (\sum LnHt * PFM) / N.$$

**[0144]** The summation is taken to give more priority to the word whose frequency is higher in a particular form. Partial or Full Match (PFM) increases the priority if the word partially or fully matches the form name. The calculation divides by the number of words in the form (N) to normalize the frequency if the form has a large numbers of words.

**[0145]** The vector space creation system stores in a table the priority of each word in the form. A vector is described as (a1, a2, . . . ak). Where a1, a2 . . . ak are the magnitude in the respective dimensions. For example, for input words and corresponding line heights of a W-2 tax form, the following are word-priority vectors are stored:

|              |    |
|--------------|----|
| omb          | 10 |
| employer     | 5  |
| employer     | 5  |
| wages        | 5  |
| compensation | 5  |
| compensation | 5  |
| dependent    | 5  |
| wages        | 10 |
| social       | 5  |
| security     | 5  |
| income       | 5  |
| tax          | 5  |
| federal      | 5  |
| name         | 5  |
| address      | 5  |

The normalized valued for the priorities are:

|              |          |
|--------------|----------|
| omb          | 0.666667 |
| employer     | 0.666667 |
| wages        | 1        |
| compensation | 0.666667 |
| dependent    | 0.333333 |
| social       | 0.333333 |
| security     | 0.333333 |
| income       | 0.333333 |
| tax          | 0.333333 |
| federal      | 0.333333 |
| name         | 0.333333 |
| address      | 0.333333 |

**[0146]** In such a vector space, the words with larger font size or higher frequency will have higher priority.

**[0147]** The ranking system calculates the cosine distance of two vectors V1 and V2 as:

$$\cos \theta = (V1.V2) / (|V1| * |V2|)$$

where V1.V2 is the dot product of two vectors and |V| represents the magnitude of the vector. When the cosine distance nears 0, that means the vectors are orthogonal and when it nears 1 it means the vectors are in the same direction or similar.

**[0148]** The class which has the maximum cosine distance with the form is the class to which the form should be classified, and is shown by module 965.

**[0149]** The class identification system performs point pattern matching based on the image features collected during image processing. As mentioned earlier, the point pattern matching of documents is performed by creating a string from

the points detected in the image and then using Levenshtein distance to measure the gap between the trained set with the input image.

[0150] In the preferred embodiment, the results of the ranking and the point pattern matching are used to determine the class matching values. If the system is not successful in finding a class match within a defined threshold, the document is marked as unclassified by the success evaluation module defined above.

[0151] FIG. 10 is a system diagram of the organized document development system. System 564 has a business rules database 1010, a business rules engine 1020, a bookmark and tab library 1040, a summary page system 1042 and a database update system 1044.

[0152] The business rules database stores rules that determine the ordering of documents for a given domain. For example, a simple business rule for organizing tax documents is to organize all wage related documents like W-2's first, followed by interest income documents, etc.

[0153] The business rules engine identifies and orders the information in the jobs database. The business rules engine applies the rules in the business rules database to the documents in the jobs database.

[0154] The FIG. 11 shows such an example of an automatically organized tax source document using the system.

[0155] The bookmark and tag library creates an organized electronic document based on the outputs of the business rules engine.

[0156] The summary page system creates a summary of key document data. In the preferred implementation, the summary includes a table of contents, the date and time the document was processed, the name of the job, etc.

[0157] In the preferred implementation, after the document is fully organized, the document is stored in the job database, the job is marked completed and appropriate message is sent to the processing server to make the final finished document available to the end user either through their portal.

[0158] FIG. 12 is a system diagram of the service control manager. System 526 has a main thread 1201, task queues 1202, database client thread controllers 1203, task queues 1204, slave controllers 1205 and SCM queue 1206.

[0159] The main thread controls the primary state machine for all the jobs in the system.

[0160] Task queues 1202 provide message queues for database communication.

[0161] Database client thread controllers manage the database server interface.

[0162] Task queues 1204 provide message queues for communication with slave controllers.

[0163] Slave controllers manage various slave processes via the slave controller interface.

[0164] The SCM queue provides a mechanism for the various controllers to communicate with the main thread.

[0165] In the preferred implementation, various threads communicate between each other using message queues. Whenever a new document is received for processing, the main thread is notified and it requests the DB client thread to retrieve the job for processing based on the states and the queue of other jobs in the system.

[0166] In the preferred implementation, once the job is loaded in memory, a finite state machine for that job is created and the job starts to be processed. The main thread puts the job on a particular task queue based on the state machine instructions. For example, if the job needs to be image processed,

then the job will be placed on the image processing task queue. If the slave controller for the image processing slave finds an idle image processing slave process, then the job is picked up from that queue and given to the slave process for processing. Once the slave finishes performing its assigned task, it returns the job to the slave controller which puts the job back on the SCM queue. The main thread sequentially picks up the job from the SCM queue and decides on the next state of the job based on the FSM states. Once a job is completed, the FSM for the job is closed and the organized document is returned to the repository and made available to the client's portal as a finished and processed document.

[0167] FIG. 13 is a diagram that depicts the various components of a computerized document analysis system according to certain embodiments of the invention. The method of parallel processing each job is performed by a host computer, 1301 that contains volatile memory, 1302, a persistent storage device such as a hard drive, 1308, a processor, 1303, and a network interface, 1304. Using the network interface, the system computer can interact with databases, 1305, 1306. Although FIG. 13 illustrates a system in which the system computer is separate from the various databases, some or all of the databases may be housed within the host computer, eliminating the need for a network interface. The programmatic processes may be executed on a single host, as shown in FIG. 13, or they may be distributed across multiple hosts.

[0168] The host computer shown in FIG. 13 may serve as a document analysis system. The host computer receives electronic documents from multiple users. Workstations may be connected to a graphical display device, 1307, and to input devices such as a mouse 1309, and a keyboard, 1310. Alternately, the active user's work station may comprise a handheld device.

[0169] In some embodiments, the flow charts included in this application describe the logical steps that are embodied as computer executable instructions that could be stored in computer readable medium, such as various memories and disks, that, when executed by a processor, such as a server or server cluster, cause the processor to perform the logical steps.

[0170] While text extraction and recognition may be performed using OCR or OCR-like techniques it is not limited to such. Other techniques could be used, including image recognition-like techniques.

[0171] Organizing electronic documents is not limited to bookmark and/or folder approaches. It includes any ways in which it can be made easier to find and use the documents, such as document tagging.

[0172] As described above, preferred embodiments extract image features from a document and use this to assist in classifying the document category. These image features include inherent image features, e.g., lines, line crossings, etc. that are put in place by the document authors (or authors of an original source or blank document) to organize the document or the like. They were typically not included as a means of identifying the document, even though the inventors have discovered that they can be used as such, especially with the use of machine learning techniques.

[0173] While many applications can benefit from extracting both image and text features so that the extracted features may be used to classify documents, for some applications, image features alone may suffice. Specifically, some problem domains may have document categories where the inherent

image features are sufficiently distinctive to classify a document with high enough confidence (even without processing text features).

[0174] Although the invention has been described and illustrated in the foregoing illustrative embodiments, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of implementation of the invention can be made without departing from the spirit and scope of the invention. Features of the disclosed embodiments can be combined and rearranged in various ways.

What is claimed is:

1. In a document analysis system that receives jobs from a plurality of users and which automatically classifies documents to organize each job according to the categories of documents the job contains, a method of parallel processing each job comprising:

- for each job, automatically separating the job into its constituent electronic documents;
- for each received electronic document, automatically separating the document into subsets of electronic pages;
- for each page of each subset, automatically extracting image features that are indicative of how the document is laid out or textually-organized and therefore indicative

of a corresponding document category and automatically extracting text features it contains, in which feature extraction for each subset is done independently and in parallel of such automatic extraction for the other subsets of the document;

for each subset, automatically comparing the extracted features with feature sets associated with each category of document to determine a comparison score for the subset;

using the comparison score for each of the subsets to automatically classify the electronic document as being one of the categories of documents; and

organizing the job according to the categories of documents the job contains.

2. The method of claim 1, wherein the constituent subset comprises an electronic page of the constituent electronic document.

3. The method of claim 1, wherein the extracted image features and the extracted text features for each page of each subset are processed independently and in parallel.

4. The method of claim 1, wherein the constituent subset comprises a part of an electronic page of the constituent electronic document.

\* \* \* \* \*