(19) World Intellectual Property Organization International Bureau



(43) International Publication Date 26 May 2006 (26.05.2006)

- (51) International Patent Classification: *G06K 9/00* (2006.01) *G06K 9/62* (2006.01)
- (21) International Application Number: PCT/US2005/040905
- (22) International Filing Date: 10 November 2005 (10.11.2005)

(25) Filing Language: English

- (26) Publication Language: English
- (30) Priority Data: 60/627,465 11 November 2004 (11.11.2004) US
- (71) Applicant (for all designated States except US): THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK [US/US]; 110 Low Memorial Library, 535 West 114th Street, New York, New York 10027 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): LONG, Xi [US/US];
55 Tiemann Place, Apartment 35, New York, New York 10027 (US). CLEVELAND, W. Louis [US/US]; 605 West 113th Street, Apartment 75, New York, New York 10027 (US). YAO, Y. Lawrence [US/US]; 560 Riverside Drive, Apartment 7J, New York, New York 10027 (US).

(10) International Publication Number WO 2006/055413 A2

- (74) Agents: WALPERT, Gary, A. et al.; Wilmer Cutler Pickering Hale and Dorr LLP, 399 Park Avenue, New York, New York 10022 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

 without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHODS AND SYSTEMS FOR IDENTIFYING AND LOCALIZING OBJECTS BASED ON FEATURES OF THE OBJECTS THAT ARE MAPPED TO A VECTOR

(57) Abstract: A method of identifying and localizing objects belonging to one of three or more classes, includes deriving vectors, each being mapped to one of the objects, where each of the vectors is an element of an N-dimensional space. The method includes training an ensemble of binary classifiers with a CISS technique, using training sets generated with an ECOC technique. For each object corresponding to a class, the method includes calculating a probability that the associated vector belongs to a particular class, using an ECOC probability estimation technique. The method includes generating a confidence map for each object type using the probability calculated for the vector as a confidence value, comparing peaks in the map for the object type with corresponding peaks in maps for other classes, using a highest peak to assign class membership, and localizing the object corresponding to the highest peak.

METHODS AND SYSTEMS FOR IDENTIFYING AND LOCALIZING OBJECTS BASED ON FEATURES OF THE OBJECTS THAT ARE MAPPED TO A VECTOR

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit of the following Patent Applications: U.S. Provisional Patent Application Serial No. 60/627,465, filed November 11, 2004, entitled "Methods And Systems For Identifying And Localizing Objects In A Digital Image," which is hereby incorporated by reference in its entirety.

BACKGROUND OF THE INVENTION

[0002] The present invention relates to pattern recognition systems. Particularly, the invention relates to systems for analyzing vector data to identify and/or localize objects that are represented by the vector data.

[0003] Cell detection in bright field microscopy is an inherently difficult task due to the immense variability of cell appearance. Even more difficult is the recognition of the subtle differences in appearance that distinguish unstained viable from non-viable cells in bright field images. Although an experienced observer can sometimes recognize these differences, viability stains are commonly used for reliable determination of viability. The requirement of a human observer represents a severe impediment to the development of high throughput systems that require recognition of viable cells. Therefore, there is a great need for effective algorithms that automatically recognize viable cells.

[0004] Currently, a typical approach for cell detection is to use fluorescent probes that have a chemical specificity for cell organelles. However, this approach can consume one or more of a very limited number of available fluorescence channels just for the purpose of cell identification. Currently for commercially available microscopes there are typically only four channels for simultaneous monitoring and eight channels for sequential observation, while there are many cellular characteristics for which the fluorescence

WO 2006/055413

PCT/US2005/040905

channels may be used to detect. It is therefore highly desirable to identify cells with a method that uses transmitted light illumination, thereby permitting all available fluorescence channels to be used to obtain cellular and subcellular information for further cell analysis.

[0005] Classical image analysis approaches require end-users to have programming skills and require independent optimizations for different cell types. An alternative is to use machine-learning techniques, which avoid end-user programming since classifiers only need to be trained. For example, Artificial Neural Networks (ANNs) have been successfully used to identify cells in bright field images. These algorithms are able to capture complex, nonlinear, relationships in high dimensional feature spaces. However, ANNs are based on the Empirical Risk Minimization (ERM) principle. Therefore, they are prone to false optimizations due to local minima in the optimization function and are susceptible to training problems such as "overfitting." This makes ANN-training a complex procedure that may be daunting for biologists and others who are not immersed in the complexities of ANNs.

[0006] In recent years, Support Vector Machines (SVMs) have been found to be remarkably effective in many real-world applications (see, for example, V. Vapnik, *Statistical Learning Theory*, Wiley, 1998; E. Osuna, R. Freund, and F. Girosi., *Support Vector Machines: Training and Applications*. *A.I. Memo* 1602, MIT A.I. Lab., 1997; P. Papageorgiou and T. Poggio, *A trainable object detection system: car detection in static images*, *A.I. Memo* 1673, MIT A.I. Lab., 1999; K. Veropoulos, N. Cristianini and C. Campbell, *The application of Support Vector Machines to Medical Decision Support: A Case Study*, *ACAI* 99, *Workshop on "Support vector machines theory and applications,"* Crete, Greece, July 14, 1999; and, Michael P. S. Brown, W. N. Grundy, et al., *Knowledge-based analysis of microarray gene expression data by using support vector machines*, *PNAS*, vol. 97, no. 1, 2000, pp. 262-267.

[0007] Unlike ANNs, SVMs follow the Structural Risk Minimization (SRM) principle, which aims at minimizing an upper bound of the generalization error. As a

WO 2006/055413

PCT/US2005/040905

result, an SVM tends to perform well when applied to data outside the training set. SVMs also have many desirable properties such as flexibility in choice of kernel function and implicit mapping into high dimensional feature spaces. But what makes SVMs most attractive is that they avoid several major problems associated with ANNs. For example, SVMs control overfitting by restricting the capacity of the classifier. They also depend on the solution of a convex Quadratic Programming (QP) problem which has no local extrema. The unique optimal solution can therefore be efficiently obtained.

SUMMARY OF THE INVENTION

In general, the invention identifies and/or localizes one or more objects in a [0008] sample, where the objects include a physical material or substance. Alternatively, the object may include an "informational" object, such as a DNA sequence or a textual data packet. In one aspect of the invention, a machine vision component is provided which is capable of identifying the at least one object in a sample, based on features of the object that are mapped to a vector. The machine vision component may use various types of illumination for identifying the at least one substance, such as light illumination. In one embodiment, the machine vision component includes machine learning software that, when executed, identifies the object in the sample based on features of the object that are mapped to a vector. The machine vision component may also include software for preprocessing the image of the sample to reduce the size of the image data used by the machine learning software to identify the at least one substance in the sample. In some embodiments, the preprocessing software preprocesses may use techniques such as Principle Component Analysis, Independent Component Analysis, Self-Organization Maps, Fisher's Linear Discriminant, and kernel PCA.

[0009] In one aspect, the invention is a method of identifying one or more objects, wherein each of the objects belongs to a first class or a second class. The first class is heterogeneous and has C subclasses, and the second class is less heterogenous than the

first class. The method includes deriving a plurality of vectors each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an Ndimensional space. The method further includes preprocessing each of the plurality of vectors using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the plurality of vectors to M dimensions, wherein M is less than or equal to C. The method also includes classifying the preprocessed vectors by (i) grouping the preprocessed vectors belonging to any of the C subclasses of the first class into a first set of vectors, and (ii) grouping the preprocessed vectors belonging to the second class into a second set of vectors.

[0010] In one embodiment, each of the plurality of vectors includes information mapped from a digital image. In another embodiment, the information mapped from a digital image includes a pixel patch. In one embodiment, the preprocessed vectors are classified with an artificial neural network. In another embodiment, the preprocessed vectors are classified with a support vector machine. Another embodiment includes training the support vector machine with training sets generated with a compensatory iterative sample selection technique.

[0011] In one embodiment, the compensatory iterative sample selection technique includes (a) selecting a first working set of pre-classified objects from a set of training sets, (b) training the support vector machine with the first working set, (c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects, (d) selecting a replacement set of pre-classified objects from the set of the working set with the replacement set, and (e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

[0012] In another aspect, the invention is a method of identifying and localizing one or more objects, where each of the objects belongs to either a first class or a second class.

The method includes deriving a plurality of vectors each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an N-dimensional space. The method further includes training a support vector machine with a compensatory iterative sample selection technique, and processing the plurality of vectors with the support vector machine, so as to classify each of the plurality of vectors into either the first class or the second class.

[0013] In one embodiment, each of the plurality of vectors includes information mapped from a digital image. In another embodiment, the information mapped from a digital image includes a pixel patch.

[0014] In another aspect, the invention is a method of identifying and localizing one or more objects in a digital image, where each of the one or more objects belongs to either a first class or a second class. The method includes deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects. Each of the plurality of pixel patches is an element of an Ndimensional space. The method also includes training a support vector machine with a compensatory iterative sample selection technique, and processing the plurality of pixel patches with the support vector machine, so as to classify each of the plurality of pixel patches into either the first class or the second class.

[0015] In another aspect, the invention is a method of identifying one or more cells in a digital image, where each of the one or more cells belongs to one of two or more classes. The method includes deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects. Each of the plurality of pixel patches is an element of an N-dimensional space. The method also includes training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique, and processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the two or more classes.

[0016] In one embodiment, each of the ensemble of binary classifiers is a support

WO 2006/055413

PCT/US2005/040905

vector machine. In another embodiment, the method further includes, for each pixel patch, calculating a probability that the pixel patch belongs to a particular one of the two or more classes, using an Error Correcting Output Coding probability estimation technique. In another embodiment, the method further includes localizing a cell in the digital image by identifying a pixel patch having a cell that is centered within the pixel patch.

[0017] In another aspect, the invention is a method of identifying one or more objects, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of vectors, each being mapped to one of the one or more objects. Each of the plurality of vectors is an element of an N-dimensional space. The method further includes training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The method also includes processing the plurality of vectors with the ensemble of binary classifiers, so as to classify each of the plurality of vectors into one of the three or more classes.

[0018] In another aspect, the invention is a method of identifying one or more objects in a digital image, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an N-dimensional space. The method also includes training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The method also includes processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the three or more classes. One embodiment further includes localizing an object in the digital image by identifying a pixel patch having an object that is centered within the pixel patch.

[0019] In another aspect, the invention is a method of identifying and localizing one

or more objects, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of vectors, each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an N-dimensional space. The method further includes training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. For each object, the method includes calculating a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The method also includes generating a confidence map for each object type using the probability calculated for the vector as a confidence with corresponding peaks in confidence map for other classes, and using a highest peak to assign class membership. The method also includes determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0020] In another aspect, the invention is a method of identifying and localizing one or more objects in a digital image, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an Ndimensional space. The method also includes training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. For each object, the method includes calculating a probability that the associated pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The method also includes generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map, comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership.

The method further includes determining localization of the cell corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0021] In another aspect, the invention is a method of identifying and localizing one or more objects, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of vectors, being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an Ndimensional space. The method also includes training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique. For each object, the method includes calculating a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The method further includes generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map, comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership. The method also includes determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0022] In another aspect, the invention is a method of identifying and localizing one or more objects in a digital image, where each of the one or more objects belongs to one of three or more classes. The method includes deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an Ndimensional space. The method also includes training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique. For each object, the method includes calculating a probability that the pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The method also includes generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value

within the confidence map, comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other class, and using a highest peak to assign class membership. The method also includes determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0023] In another aspect, the invention is a method of generating a training set of pre-classified objects for training a classifier. The method includes applying one or more fluorescent markers to a sample containing objects to be classified, generating one or more fluorescence images of the sample containing objects to be classified, and generating a transmitted light illumination image of the sample containing objects to be classified. For each of the one or more fluorescence images, the method includes superimposing at least a portion of the fluorescence image with a corresponding portion of the transmitted light illumination image. The method also includes using information from the one or more fluorescence images to identify characteristics of corresponding objects in the transmitted light illumination image, thereby producing a transmitted light illumination image are pre-classified objects. One embodiment further includes using information from the transmitted light illumination image having one or more pre-classified objects to identify characteristics of corresponding one or more pre-classified objects to identify characteristics of corresponding one or more pre-classified objects. One embodiment further includes using information from the transmitted light illumination image having one or more pre-classified objects to identify characteristics of corresponding elements in one or more pre-classified objects to identify characteristics of corresponding elements in one or more subsequently generated fluorescent images.

[0024] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of vectors each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an N-dimensional space. The computer readable medium also includes instructions for preprocessing each of the plurality of vectors using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the plurality of vectors to M dimensions, wherein M is less than or equal to C. The computer readable medium further includes instructions for classifying the preprocessed vectors by (i) grouping the preprocessed vectors belonging to any of the C subclasses of the first class into a first set

of vectors, and (ii) grouping the preprocessed vectors belonging to the second class into a second set of vectors.

[0025] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of pixel patches from the digital image, each being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space. The computer readable medium also includes instructions for preprocessing each of the plurality of pixel patches using a Fisher Linear Discriminant, where the preprocessing reduces the dimensionality of each of the pixel patches to M dimensions, wherein M is less than or equal to C. The computer readable medium also includes instructions for classifying the preprocessed pixel patches by (i) grouping the preprocessed pixel patches, and (ii) grouping the preprocessed pixel patches belonging to the second class into a second set of pixel patches.

[0026] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of vectors each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an N-dimensional space. The computer readable medium also includes instructions for training a support vector machine with a compensatory iterative sample selection technique. The computer readable medium also includes instructions for processing the plurality of vectors with the support vector machine, so as to classify each of the plurality of vectors into either the first class or the second class.

[0027] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an N-dimensional space. The

computer readable medium also includes instructions for training a support vector machine with a compensatory iterative sample selection technique. The computer readable medium also includes instructions for processing the plurality of pixel patches with the support vector machine, so as to classify each of the plurality of pixel patches into either the first class or the second class.

[0028] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an N-dimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding decomposition technique. The computer readable medium also includes instructions for processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the two or more classes.

[0029] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of vectors, each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an N-dimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for processing the plurality of vectors with the ensemble of binary classifiers, so as to classify each of the plurality of vectors into one of the three or more classes.

[0030] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of pixel patches from the digital image, each

WO 2006/055413

PCT/US2005/040905

of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an N-dimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the three or more classes.

[0031] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of vectors, each being mapped to one of the one or more objects, where each of the plurality of vectors is an element of an Ndimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for calculating, for each object, a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The computer readable medium also includes instructions for generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map. The computer readable medium also includes instructions for comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership. The computer readable medium also includes instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0032] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium

includes instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for calculating, for each object, a probability that the pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The computer readable medium also includes instructions for generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map. The computer readable medium also includes instructions for comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other class, and using a highest peak to assign class membership. The computer readable medium also includes instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0033] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for deriving a plurality of vectors, being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for each object corresponding to a class, calculating a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The computer readable medium also includes instructions for generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map. The computer

readable medium also includes instructions for comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership. The computer readable medium also includes instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

In another aspect, the invention is a computer readable medium including [0034] stored instructions adapted for execution on a processor, comprising. The computer readable medium includes instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, where each of the plurality of pixel patches is an element of an Ndimensional space. The computer readable medium also includes instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique. The computer readable medium also includes instructions for calculating, for each cell, a probability that the pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique. The computer readable medium also includes instructions for generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map. The computer readable medium also includes instructions for comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership. The computer readable medium also includes instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

[0035] In another aspect, the invention is a computer readable medium including stored instructions adapted for execution on a processor. The computer readable medium includes instructions for applying one or more fluorescent markers to a sample containing objects to be classified. The computer readable medium also includes instructions for generating one or more fluorescence images of the sample containing objects to be

classified. The computer readable medium also includes generating a transmitted light illumination image of the sample containing objects to be classified. The computer readable medium also includes instructions for superimposing, for each of the one or more fluorescence images, at least a portion of the fluorescence image with a corresponding portion of the transmitted light illumination image. The computer readable medium also includes instructions for using information from the one or more fluorescence images to identify characteristics of corresponding objects in the transmitted light illumination image, thereby producing a transmitted light illumination image having one or more pre-classified objects. In one embodiment, the computer readable medium further includes instructions for using information from the transmitted light illumination image having one or more pre-classified objects to identify characteristics of corresponding elements in one or more subsequently generated fluorescent images.

BRIEF DESCRIPTION OF DRAWINGS

[0036] The foregoing and other objects of this invention, the various features thereof, as well as the invention itself, may be more fully understood from the following description, when read together with the accompanying drawings in which:

[0037] FIG. 1 is a flow diagram of a method for identifying at least one cell or other identifiable substance in a sample according to one embodiment of the invention.

[0038] FIG. 2 depicts images of a sample set of cell patches for use in a learning set according to one embodiment of the invention.

[0039] FIGs. 3a-3b are.graphical representations comparing PCA and FLD where (a) is projected with PCA and (b) is projected with FLD.

[0040] FIG. 4 is a graphical representation showing the effect of neuron number in the first hidden layer on the generalization properties of the ANN.

[0041] FIGs. 5a-d depict sample images for focus variation experiments where (a)

the focal plane is at the equator of the microsphere, (b) the focal plane is at the supporting surface, (c) the focal plane is $2\mu m$ below the equator, and (d) the focal plane is $37.5 \mu m$ below the equator.

[0042] FIGs. 6a-6b are graphical representations showing misclassification rates with different focus conditions and preprocessing methods where the ANN in (a) was trained with only focused samples and applied to all samples, and in (b) was trained with focused and 25µm focus variation samples and applied to all samples.

[0043] FIGs. 7a-e depict sample images for an illumination variation experiment varying between extremely week extremely strong illumination: (a) Intensity level 3: representing extremely weak illumination; (b) Intensity level 4: representing weak illumination; (c) Intensity level 5: representing normal illumination; (d) Intensity level 6: representing strong illumination and (e) Intensity level 7: representing extremely strong illumination.

[0044] FIGs. 8a-b are graphical representations showing misclassification rates with different illumination conditions and preprocessing methods where the ANN in (a) was trained with only level 5 illumination and applied to all levels, and in (b) was trained with level 4, 5 and 6 and applied to all levels.

[0045] FIGs. 9a-b depict sample images for a size variation experiment where in (a) the microspheres had no size variation and in (b) the microspheres varied between 0% and 20% in size.

[0046] FIGs. 10a-b are graphical representations showing misclassification rates with different size variations and preprocessing methods where the ANN in (a) was trained with only 0% variation samples and applied to all samples, and in (b) was trained with 0% and 15% variation samples and applied to all samples.

[0047] FIGs. 11a-e depict sample images for a noise variation experiment where the zero-mean Gaussian noise varied between standard deviations: (a) STD=0, (b) STD=15,
(c) STD=30, (d) STD=45, (e) STD=60.

[0048] FIGs. 12a-b are graphical representations showing misclassification rates

with different noise conditions and preprocessing methods where the ANN is (a) was trained with STD=0 samples and applied to all samples, and in (b) was trained with STD=0, 45 samples and tested on all samples.

[0049] FIGs. 13a-c depict sample images for a living cell experiment where in (a) cells are almost completely separate and the background is clean, (b) most cells are attached to each other and there are trash and debris in the background, and (c) most cells are clumped together and the background is full of trash and debris.

[0050] FIGs. 14a-b depict images showing detection results of an ANN classifier with the detected cell positions denoted by white crosses in the images, where identification occurred in (a) using an ANN with PCA preprocessing (Sensitivity: 82.5%, Positive predictive value: 83.02%) and (b) using an ANN with FLD preprocessing (Sensitivity: 94.38%, Positive predictive value: 91.52%).

[0051] FIG. 15 is a graphical representation showing sensitivity (SE) and positive predictive value (PPV) results for the sample images shown in FIGs.13a-c using different preprocessing methods.

[0052] FIG. 16 is a block diagram depicting one embodiment of the present invention. This embodiment is a computer-based system for identifying and/or localizing objects.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0053] The present invention generally provides systems for the identifying and/or localizing one or more identifiable objects within a sample. The objects may include physical objects or substances, such as a living cell or cells, or portions thereof, in a digitized image of a sample. Alternatively, the objects may include informational objects such as data blocks or a DNA code sequence stored in a suitable storage medium. The objects that may be identified are generally all items that may be represented by feature

vectors. In general, the invention identifies objects by analyzing such vectors. In some embodiments, the vectors are derived from images. The term "substance" is used herein to denote items that have a mass and occupy space, including solid as well as liquid and gaseous materials. Although the described embodiments are set forth herein by way of example with respect to certain types of substances, such as cells, and certain analytical techniques, it is understood that the present invention is generally applicable to various other types of living or non-living materials and techniques, and with or without all of the components or subcomponents described herein, and is thus not limited thereto.

Machine Vision Component

[0054] At least one embodiment includes a machine vision component, which reduces or eliminates the need for intensive microscopic observation by a human operator. The machine vision component is generally a computing device with associated software that, when executed, identifies and/or recognizes objects (*e.g.*, individual cells or portions thereof, or other identifiable substances) within an image. The machine vision component also generates information related to the objects, such as the position of the object (*e.g.*, coordinates with respect to a reference position) or particular characteristics of the object (*e.g.*, whether the cells are viable or non-viable). The machine vision component operates on image data produced by a microscopy component. In one embodiment, the machine vision component is capable of identifying cells or portions thereof in a culture having a plurality of cells that vary or are non-uniform with respect to their features, and/or are partially aggregated.

[0055] Automatic cell recognition or identification may be facilitated with fluorescent probes that have a chemical specificity for cell organelles. For example, DNA intercalators may be used to stain nuclear DNA for cell identification. Fluorescent probes used in this respect, however, can consume one or more fluorescence channels. In one embodiment of the invention, the machine vision component is capable of identifying cells based on their feature set using transmitted light illumination, thereby permitting all

of the available fluorescence channels to be used for other purposes, such as to provide additional cellular and sub-cellular information that can greatly enhance the value of the information obtained by molecular genetic analysis.

[0056] In one embodiment, pixel patches are used as the primary input data elements. The described embodiment uses a square pixel patch, but other embodiments may use pixel patches having different shapes, such as rectangular, circular, elliptical, or a shape matched to the objects to be detected. In a typical pixel patch of 25×25 pixels (which is enough to enclose a complete cell), there may be 625 pixels, each of which is characterized by a grayscale intensity. In the described embodiment, the value of the grayscale intensity can range from 0 to 255, inclusive, although other grayscale ranges may be used. The ordered set of these 625 grayscale intensities generates a 625dimensional input vector for the software. Essentially, the software is taught to sort pixel patches into two or more classes. Some classes contain desired objects (such as cells or viable cells). Other classes contain all undesired objects (e.g., non-cells or non-viablecells, fragments of cells, trash). Classes of desired objects are usually much smaller (regarding the number of elements of the class) than the class of undesired objects. This embodiment scans the overall image by moving a pixel box one pixel per step, until the image has been covered. This technique of scanning the images neglects a band of cells at the periphery of the image that is one half of the width of the pixel box, although other scanning techniques may be used instead of or in addition to the described technique to address the neglected band of cells.

[0057] Referring to FIG. 1a, in one embodiment, the machine vision component identifies cells in a culture using algorithms derived based on machine learning techniques, such as Artificial Neural Networks ("ANN") or Support Vector Machines (SVM). ANNs are able to capture complex, even nonlinear, relationships in high dimensional feature spaces that are not easily handled by algorithms based on heuristic rules. The ANN-based algorithms may use pixel values directly from primary grayscale images (bright field) for cell recognition; however, the image data processed by these

types of algorithms is very large, which results in networks that may be very complex and exhibit slow performance. Accordingly, in one embodiment of the invention, the image data is preprocessed, as shown in FIG. 1b, using statistical data processing techniques, such as Principle Component Analysis ("PCA"), Independent Component Analysis ("ICA"), Self-Organization Maps ("SOM"), or preferably Fisher's Linear Discriminant ("FLD") techniques. Such preprocessing generates abstract representations of cells from the image data that are dimensionally smaller, which makes subsequent classification computationally effective. One embodiment achieves computational effectiveness by processing image data to provide a cell classifier, e.g., the ANN or the SVM, with only the information that is essential for cell recognition.

PCA may be used for dimensionality reduction by using a set of orthogonal [0058] vectors, referred to as principal components, that point in the directions of maximum covariance in the image data as the basis of the new subspace. If the dimension of the subspace is given, PCA minimizes the mean square reconstruction (projection) error, and provides a measure of importance for each axis. More formally, suppose the learning set x is composed of n sample images ($x = \{x_1, x_2, ..., x_n\}$), where each image is represented by a vector x_i in an N-dimensional image space. The goal is to find a linear transformation that maps the input set to an *M*-dimensional subspace, where M < N. After defining $x' = \{x'_1, x'_2, \dots, x'_n\} = \{(x_1 - \mu), (x_2 - \mu), \dots, (x_n - \mu)\}$, where μ is the sample mean, the new feature vector set $y = \{y_i\}$ can be defined by the linear transformation:

$$y = W^T x', \quad y_i = W^T x'_i \tag{1}$$

and
$$\widetilde{x} = Wy = WW^T x'$$

$$\widetilde{x}_{i} = Wy_{i} = WW^{T}x'_{i}, \quad i = 1, 2, ..., n,$$
 (2)

where W is an $N \times M$ matrix whose columns form a orthonormal basis of the subspace and \tilde{x} is the reconstruction from y. PCA seeks to minimize the mean square reconstruction

error:

$$J_e = E\{ \| \boldsymbol{x}' - \tilde{\boldsymbol{x}} \| \} = E\{ tr[(\boldsymbol{x}' - \tilde{\boldsymbol{x}})(\boldsymbol{x}' - \tilde{\boldsymbol{x}})^T] \}$$
$$= tr(S_r) - tr(\boldsymbol{W}^T S_T \boldsymbol{W})$$
(3)

where $S_T = x' x'^T = \sum_{i=1}^{n} (x_i - \mu) (x_i - \mu)^T$ is an N×N matrix called total scatter. Note that the last term of (3) is equal to the variance of y:

$$tr(W^{T}S_{T}W) = tr(W^{T}x'x'^{T}W) = E\{tr(yy^{T})\}$$
(4)

Therefore, minimizing the mean square reconstruction error is equivalent to maximizing the projection variance. The optimal transformation matrix W_{opt} in PCA then can be defined as:

$$W_{opt} = \arg \max_{W} \left| W^{T} S_{T} W \right| \tag{5}$$

[0059] A larger eigenvalue means more variance in the data captured by the corresponding eigenvector. Therefore, by eliminating all eigenvectors except those corresponding to the highest *M* eigenvalues, the feature space for recognition is reduced from the original *N*-dimensional image space to the subspace spanned by the top M eigenvectors. The eigenvectors have the same dimension as the original vectors.

[0060] Note that although the PCA projection is optimal for reconstruction from a low dimensional basis, it is not optimal from a classification point of view, for it only considers the total scatter of the whole sample set and makes no discrimination within the sample points. In PCA, the total scatter is maximized. Therefore, there is not only maximization of the between-class scatter, which is useful for classification, but there is also maximization of the within-class scatter, which should be minimized. Consequently,

PCA may retain or even exaggerate unwanted information. Points from individual classes in the low dimensional feature space may therefore not be well clustered, and points from different classes could be mixed together.

[0061] As described above, the PCA method treats the learning set as a whole. Since

the learning set is labeled in different classes, it should be possible to use this information to build a more reliable representation for classification in the lower dimensional feature space. The key to achieving this goal is to use class specific linear methods, such as the FLD technique, which considers not only between-class variation but also within-class variation, and optimizes the solution by maximizing the ratio of between-class scatter to within-class scatter. This can be expressed in mathematical terms as follows. Assume that each image in the learning set belongs to one of *c* classes { $C_1, C_2, ..., C_c$ }. The between-class scatter matrix S_B and within-class scatter matrix S_w can be defined as:

$$S_{B} = \sum_{i=1}^{c} m_{i} (\mu_{i} - \mu) (\mu_{i} - \mu)^{T}$$
(6)

and $S_W = \sum_{i=1}^{c} \sum_{x_k \in C_i} (x_k - \mu_i) (x_k - \mu_i)^T$

where μ is the grand mean, μ_i is the mean of class C_i and m_i denotes the number of images in class C_i . The objective of FLD is to find the W_{opt} maximizing the ratio of the determinants of the above scatter matrices:

(7)

$$W_{opt} = \arg\max_{W} \frac{|W^{T}S_{B}W|}{|W^{T}S_{W}W|}$$
(8)

 W_{opt} is known to be the solution of the following generalized eigenvalue

problem:

$$S_B W - S_W W \Lambda = 0 \tag{9}$$

where Λ is a diagonal matrix whose elements are the eigenvalues. The column vectors w_i (i = 1,...,m) of matrix W are eigenvectors corresponding to the eigenvalues in Λ ..

[0062] Compared to the PCA method, the representation yielded by FLD tries to reshape the scatter instead of conserving its original details. FLD emphasizes the discriminatory content of the image. To illustrate the benefit of FLD projection, the learning set described above was projected to a three-dimensional subspace using PCA and FLD, results of which are shown in FIGs. 3a and 3b, respectively. One can see that although the point distribution range of PCA projection is greater in all three directions, (*i.e.*, the total scatter is larger), the points from different classes are somewhat mixed together. On the other hand, the points from different classes in FLD projection are better separated and, therefore, more suitable for classification.

[0063] When FLD is used, the dimension of the resulting subspace must be reduced to no more than n-1, where n is the number of recognition classes. Instances for which there are only two recognized classes (*e.g.*, cell and non-cell) result in a one-dimensional subspace, which may be inadequate for cell representation. In one embodiment, the non-cell classes are divided into two or more subclasses. For example, as shown in FIG. 2, the non-cell classes may be divided into 10 subclasses, which result in a total class number of 11, making FLD use practical. Thus, the difficulty with FLD can be overcome by treating the cell detection as a multi-class problem in the preprocessing stage, and as a binary classification problem in subsequent stages. The division of the "non-cell" class into multiple subclasses allows one to generate processed input vectors with sufficient dimensionality for effective cell recognition. This maneuver is possible because the "non-cell" class is highly heterogeneous in relation to the "cell" class.

[0064] In one embodiment, the ANN is trained to recognize whether an image patch contains a centered cell body or a viable/non-viable cell, as the case may be. This is

generally accomplished with image patches represented by feature vectors derived in preprocessing. The ANN uses the input-output mapping learned from a set of training samples to generalize to data not "seen" before. FIG. 2 shows a learning set of cell image patches used to train the ANN, which is manually selected from microscopic images for testing. A similar set is also used to train the ANN to detect microspheres in a sample. The learning set Ω is composed of two subsets ($\Omega = \Omega^{\text{pos}} + \Omega^{\text{neg}}$). Ω^{pos} contains patches of centered cells and is labeled "cell". All images in Ω^{pos} belong to a single class. Ω^{neg} is labeled "non-cell" and is divided into 10 sub-classes according to the similarity of the images. For example, subclasses 1-8 contain a specific fraction of a cell. Images in subclass 9 are almost blank and subclass 10 includes images with multiple fragments of different cells.

[0065] The training of an ANN involves gradually modifying the synaptic weights according to the back-propagated output error for each sample, until the desired average responses are obtained on the entire set. In one embodiment, the network structure is designed to be flexible by allowing the size of the network to be adjusted as a result of easily changing only a few parameters. In one embodiment, three layers in addition to the input layer are used: two hidden layers, and one output layer. The size of the neural network can be adjusted according to the size of training set and dimension of input vectors. In order to be able to establish all possible representations, a bias weight in addition to the inputs is included in each neuron. The hyperbolical tangent sigmoid function, TANSIG(x), may be used as the transfer function throughout the network.

[0066] After a network is trained with examples, the most important issue is how well the network generalizes to new data. The capacity for generalization greatly depends on the structure of the neural network. Generally, more neurons in the hidden layers give the system more capacity to partition the data. However, if the network has too many neurons, it will learn insignificant aspects of the training set and lose its generalization ability, a phenomenon termed "overfitting." Although there is no universal simple rule to determine how many hidden units are required for a given task, one rule of thumb is to

obtain a network with the fewest possible neurons in the hidden layer. Using the smallest possible size not only helps improve generalization, it also increases the computational speed of the system, for there is roughly a linear relationship between network size and speed.

[0067] One method to train an ANN with good generalization ability is to start with a small initial network and gradually add new hidden units in each layer until efficient learning is achieved. To avoid drawbacks associated with this training method, such as slow learning and difficult-to-avoid local minima, a pruning strategy may be used, which starts with a large network and excises unnecessary weights and units. The training results of the ANNs are discussed below in the examples section in connection with FIGs. 5-15.

[0068] As an alternative classifier to Artificial Neural Networks (ANN), Support Vector Machines (SVM) for cell recognition may also be used. SVMs like ANNs, are statistical learning machines that use supervised learning techniques and therefore eliminate the need for end user programming. SVMs have been successfully applied to distinguish between unstained viable and non-viable cells in bright field images, as discussed in X. Long, W. L. Cleveland and Y. L. Yao, *Automatic Detection of Unstained Viable cells in Bright Field Images Using a Support Vector Machine with an Improved Training Procedure*, *Computers in Biology and Medicine*, 2004, which poses different obstacles in comparison to cell and non-cell determinations.

[0069] In cell recognition, the training sample set is extremely large and highly unbalanced, in comparison since the number of pixel patches in the "viable-cell" (VC) class is much smaller than the number in the "not-a-viable-cell" (NAVC) class. The nature of a SVM is such that it requires a memory space that grows quadratically with the training sample number. Therefore, in practice, an SVM is typically trained with a small subset of the pixel patches available for training, raising the possibility that the chosen subset is not representative. The above problem has been solved where the two classes are balanced, *i.e.*, where both are represented by comparable numbers of class elements,

however, the problem has not been solved when classes are unbalanced and the number of samples is large relative to the available computational resources.

[0070] To handle instances where there is relatively high degree of imbalance, iterative SVM training, e.g., a "compensatory iterative sample selection" (CISS) technique, may be applied, which involves selecting a working set of pre-classified objects from the set of possible training objects. In one embodiment, the initial working set contains approximately equal numbers of samples from both classes, such as cell and non-cell, even though the classes are unbalanced. The SVM may then be trained with the initial working set and tested on the remainder of the pre-classified objects. From the set of incorrectly classified objects, a replacement set of a fixed size may then be selected, randomly or otherwise, and used to replace an equal number of objects in the working set. In one embodiment, the relative contributions of the classes are constrained to the replacement set. That is, in cases where there is an imbalance in the classes, the replacement set contains a larger proportion of the misclassified objects of the larger class. The contribution of the smaller class may also be negated or replaced with samples from the larger class. The replacement set is not therefore randomly chosen from the misclassified objects of the combined classes; rather, the replacement set is derived only from misclassified objects of the much larger class, which compensates for the imbalance. The SVM may then be trained with the new working set and retested. This procedure can be repeated until satisfactory accuracy is achieved, or until no further improvement occurs.

[0071] In many cases with highly imbalanced classes, a useful SVM can also be trained with a working set that includes most or all of the smaller class. However, in other cases, the size of the smaller class may be too large for this to be done. In these instances, the SVM may be trained with a working set with less extreme weighting. For example, a 10%-90% weighting might be better than the 0%-100% weighting used. The same solution may be applied to the selection of the replacement set. The weighting used for the may generally be optimized empirically to provide the best results.

WO 2006/055413

PCT/US2005/040905

[0072] SVMs represent an approximate implementation of the SRM principle and were first introduced by Vapnik *et al.* to solve pattern recognition and regression estimation problems. In what follows, we denote the training data as $(x_i, y_i=f(x_i))$: $i = 1, 2, ..., l, y_i \in \{-1, +1\}, x_i \in \mathbb{R}^n$.

[0073] In a linearly separable case, the SVM classifier follows the intuitive choice and selects the hyperplane (among many that can separate the two classes) that maximizes the margin, where the margin is defined as the sum of the distances of the hyperplane to the closest points of the two classes.

[0074] If the two classes are non-separable, positive slack variables are introduced to allow some training samples to fall on the wrong side of the separating hyperplane. The SVM then finds the hyperplane that maximizes the margin and, at the same time, minimizes a quantity proportional to the number of classification errors. The trade-off between maximizing the margin and minimizing the error is controlled by a user-adjusted regularization parameter C>0. A large C corresponds to a high penalty for classification errors.

[0075] In many practical cases, however, nonlinear decision surfaces are needed. Nonlinear SVMs can be generalized from linear SVMs by using a nonlinear operator $\Phi(\cdot)$ to map the input pattern x into a higher (even infinite) dimensional Euclidean space H. It has the form:

$$f(x) = w^T \Phi(x) + b$$

Mathematically, it can be shown that the solution of the nonlinear case is:

$$f(x) = \sum_{i=1}^{l} \alpha_{i} y_{i} \Phi^{T}(x_{i}) \Phi(x) + b = \sum_{i=1}^{l} \alpha_{i} y_{i} K(x_{i}, x) + b,$$

where the coefficients α_i are the solution of the following convex QP problem:

$$\max L_D(\alpha_i) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) ,$$

subject to
$$0 \le \alpha_i \le C$$
, and $\sum_{i=1}^{l} \alpha_i y_i = 0$,

where the function $K(\cdot, \cdot)$ is called kernel function and defined as

$$K(x,z) \equiv \Phi^T(x)\Phi(z)$$

[0076] It turns out that, in a typical problem, only the coefficients α_i of a few training samples will be nonzero. These samples are called "Support Vectors" (SVs). Let $s_j, \alpha^*_{j,j}=1,2,...,m, (m < l)$ denote these SVs and their corresponding nonzero coefficients. The decision function in (2) can be rewritten in the "sparse" form of the support vectors as:

$$f(x) = \sum_{j=1}^{m} \alpha^{*}{}_{j} y_{j} \Phi^{T}(s_{i}) \Phi(x) + b = \sum_{j=1}^{m} \alpha^{*}{}_{j} y_{j} K(s_{j}, x) + b$$

This form shows that the decision functions are determined only by the SVs, which typically represent a very small fraction of the whole training set.

[0077] The impractically large size of the "NAVC" class and the extremely unbalanced size ratio of "NAVC" to "VC" class make this application a very difficult case. In many applications, random selections of training sets with a controllable size will typically be made. However, since only a very small portion of the total set is used, the randomly selected training sets may not be representative. [0078] Osuna *et al.* proposed a "decomposition" algorithm to make use of all the available training samples. The algorithm exploits the sparseness of SVM solution and iteratively selects the "most representative" examples for the classes. In the context of SVMs, "most representative" refers to samples that are close to the boundary and are difficult to classify. Osuna's algorithm has been modified to produce an algorithm referred to herein as "Compensatory Iterative Sample Selection" (CISS) to handle the extremely unbalanced training sample set. The new algorithm includes four steps:

- 1) Construct an initial working set S (with a user-selected size l) from the whole training set S_T , such that S consists of all samples from the "VC" class S_V , and a comparable number of samples randomly selected from the very large "NAVC" sample set S_N .
- 2) Train a SVM classifier f(x) with S and use f(x) to classify the remainder of the preclassified training set S_T . Put misclassified "NAVC" samples into a set S_M .
- 3) From S_M , a "replacement set" of size *n* is randomly selected and is used to replace an equal number of "NAVC" samples in the working set which were correctly classified in step 2.
- 4) Repeat steps 2)-3) until no further improvement occurs.

This algorithm differs from that of Osuna et al. in the following ways:

- In Step 1, when generating the initial working set, Osuna *et al.* arbitrarily chose samples from the whole training set. Since only balanced classes were used in their case, the resulting working set was also approximately balanced. In our algorithm, since the training set is extremely unbalanced, arbitrary choices of samples from the whole training set would result in a working set that is likewise extremely unbalanced. To solve this problem, we constrained the initial working set to contain all the samples from the "VC" class and a comparable number of samples randomly selected from the very large "NAVC" class.
- 2) In the strategy of Osuna *et al.*, there is no deliberate control of the relative contributions of the two classes to the replacement set. This has been shown to work well for balanced cases. We have tested this approach with highly unbalanced classes and found it to be unusable (data not shown). In our algorithm, we deliberately constrain the relative contributions of the two classes to the replacement set. In the detailed studies described below, we make the contribution of the "VC" class zero and replace only samples from the much

larger "NAVC" class.

[0079] CISS leads to a reduced classification error that converges to a stable value. Essentially, we show that substitution of correctly classified "NAVC" samples in the working set *S* with misclassified "NAVC" samples from S_N , an improvement of the function for max $L_D(\alpha_i)$ above can be achieved.

[0080] Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$ be the working set used to train the SVM classifier, and let $S_N = \{(x_{l+1}, y_{l+1}), (x_{l+2}, y_{l+2}), ..., (x_L, y_L)\}$ be the remainder of the training samples, where $y_i = -1$ i = l+1, l+2,..., L, i.e. total training set $S_T = S \cup S_N$.

[0081] Let $\wedge_1 = \{\alpha_1, \alpha_2, ..., \alpha_l\}$ be an optimal solution to (3) when training with the working set *S*. We can extend it to the entire training set as $\wedge = \{\alpha_1, \alpha_2, ..., \alpha_l, \alpha_{l+1}, \alpha_{l+2}, ..., \alpha_L\}$ with $\alpha_i = 0$, i = l+1, l+2,..., *L*. Note that although the solution of \wedge is optimal over working set *S*, it may not be optimal over S_T . Since the SVM guarantees to find the optimal solution, what we need to prove here is that the solution \wedge is not necessarily optimal when we replace a correctly classified sample in *S* with one that is misclassified in S_N .

[0082] We now replace a correctly classified (randomly chosen) sample $\alpha_i=0$, $i\in[1, l]$ (note that for SVMs, points which are correctly classified during the training will have a coefficient of $\alpha=0$) with $\alpha_m=0$, $m\in[l+1, L]$, $y_mf(x_m)<1$. After replacement, the new subproblem is optimal if and only if $y_mf(x_m)\geq 1$.

[0083] Assume that there exists a margin support vector a_p , with corresponding label y_p =-1. We have $0 < \alpha_p < C$ since it is a support vector. (We can also pick out an error support vector; the proof is analogous). Then there exists a positive constant δ such that $0 < \alpha_p - \delta < C$. Consider a vector $\Lambda' = \{\alpha'_1, \alpha'_2, ..., \alpha'_l, \alpha'_{l+1}, \alpha'_{l+2}, ..., \alpha'_L\}$ which has the elements : $\alpha'_m = \delta$, $\alpha'_p = \alpha_p - \delta$, and $\alpha'_i = \alpha_i$ for all other elements. Then we have:

$$L_{D}(\Lambda') = \sum_{i=1}^{L} \alpha'_{i} - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha'_{i} \alpha'_{j} y_{i} y_{j} \times K(x_{i}, x_{j})$$

Note that:
$$\sum_{i=1}^{L} \alpha_i = \sum_{i=1}^{L} \alpha'_i, \ \alpha'_m = \delta = \delta + \alpha_m \text{ and } \alpha'_p = \alpha_p - \delta. \text{ It can be shown that:}$$
$$L_D(\Lambda') = L_D(\Lambda) - \delta[(y_m f(x_m) - 1] - \frac{\delta^2}{2} [K(x_p, x_p) - 2y_p y_m \times K(x_p, x_m) + K(x_m, x_m)]$$

If the δ is chosen small enough, the third term of above can be neglected. Then, above become:

$$L_D(\Lambda') = L_D(\Lambda) - \delta[(y_m f(x_m) - 1]]$$

Since $y_m f(x_m) < 1$, it can be easily shown from the equation above that $L_D(\Lambda') > L_D(\Lambda)$.

[0084] An extension of the CISS algorithm is to combine it with multiclass classification techniques as discussed above. For example, in a cell recognition context, the multiclass classification problem arises when there is more than one type of cell in the image. Currently, a popular strategy to handle a multiclass classification problem is to decompose it into a series of binary classifications. An example of one implementation of this strategy is "Error Correcting Output Coding" (ECOC). A binary classifier is trained on each binary problem. Binary classification results are then combined together to give the overall classification. Appendix A of this application (X. Long, W. L. Cleveland, and Y. L. Yao, *Multiclass Cell Detection in Bright Field Images of Cell Mixtures with ECOC Probability Estimation*) provides an ECOC-based probability estimation algorithm to enable the pixel patch decomposition technique described herein to be used in multiclass classification, in particular as applied to bright field images of living cells.

[0085] In many cases, the resulting binary classification problems will be also unbalanced, especially in cell detection and localization applications. Specifically, the "non-cell" class will be much larger than the class for any one cell type and often much larger than all of the "cell" classes combined. In these cases, the CISS algorithm is very useful in dealing with the multiclass classification problem since the classification accuracy of a multiclass classifier depends largely on the classification accuracy of each binary classifier and CISS can improve the accuracy of the individual binary classifiers.

[0086] The graph below is a graphical representation of the results obtained with the CISS algorithm in combination with the ECOC decomposition strategy.



The test set is a randomly generated artificial 2D data set that includes four classes. One class is much larger than the other three. As the graph indicates, the CISS algorithm clearly shows a trend of convergence, and the overall classification error is lowered. The fact that the CISS algorithm converges with randomly generated data emphasizes the generality of this approach, *i.e.*, the application to cell recognition is only one of many possible applications of this method.

[0087] An ECOC-based cell detection framework for bright field images of cultured cells is presented in FIG. 3c. The framework employs the multiclass classification and probability estimation ability of our proposed algorithm to analyze bright field images of cell mixtures. It permits not only the identification of the desired cells but also gives their locations relative to the pixel coordinates of the primary image. It also uses pixel patches as the primary input data elements. Essentially, the software is WO 2006/055413

PCT/US2005/040905

taught to classify pixel patches into different classes. Each class corresponds to a single cell type, except for the larger class containing all undesired objects (e.g. background, fragments of cells, trash), denoted as "non-cell."

[0088] Initially, ECOC is used to train an ensemble of SVM classifiers. This is done with input vectors that are derived from manually extracted training patches and are represented as linear combinations of feature vectors derived in Principal Component Analysis (PCA) preprocessing.

[0089] For each pixel p in the testing image (excluding pixels in the margin around the edges), a pixel patch centered at that pixel is extracted and represented in the same way as that in training process. The probability that this extracted patch belongs to each class is calculated by ECOC probability estimation. For each class corresponding to a cell type, this probability is then used as a "confidence value" $C[p] \in [0,1]$ in a "confidence map" for that cell type. Pixels in each confidence map are the confidence values of their corresponding patches in the original image and form "mountains" with large peaks representing a high probability of presence of the corresponding cell type. A given peak in a confidence map is compared with the corresponding peaks in the other confidence maps. The confidence map with the highest peak at that location gives the assignment of class membership. The pixel coordinates of the highest peak provide localization. It should be pointed out that generating a confidence map for the "non-cell" class is unnecessary in this case, since localization of the non-cell objects is not important.

[0090] For the ECOC approach, binary classifiers have to be trained as the base classifiers. The choice of base classifier can be arbitrary. In one embodiment, Support Vector Machines (SVMs) are used with the RBF kernel $K(x, y) = e^{-\gamma ||x-y||^2}$. The SVM classifier in this embodiment is implemented by modifying LibSVM (*see* http://www.csie.ntu.edu.tw/~cjlin/libsvm/). The regularization parameter *C* and the kernel parameter γ are optimized using a two-step "grid-search" method for each classifier. In the first step, a coarse grid-search with a grid size of 1 was used to localize a Region of

Interest (ROI) containing the optimal values (shown in FIG. 3d). In the second step, a fine grid-search over the ROI with a grid size of 0.25 is used to give more precise values for C and γ . The result is shown in FIG. 3e.

[0091] In one embodiment, the standard ECOC method is modified to enable probability estimation. Our new algorithm is an extension of the pairwise coupling method introduced by Hastie and Tibshirani (T. Hastie and R. Tibshirani, *Classification by pairwise coupling, Advances in Neural Information Processing Systems*, vol. 10, MIT Press, 1998.)

[0092] The Hastie and Tibshirani's pairwise coupling method can be briefly described as follows. Assume that after training a classifier using the samples from class *i* (labeled +1) and samples from class *j* (labeled -1), the pairwise probability estimation for every class *i* and *j* ($i \neq j$) is $r_{ij}(x)$. According to the Bradley-Terry (BT) model, $r_{ij}(x)$ is related to the class posterior probabilities $p_i = P(class=i | X=x)$ (i=1,2,...k):

$$r_{ij}(x) = P(class = i \mid class = i \cup class = j, X = x) = p_i(x)/(p_i(x) + p_j(x))$$

Note that p_i is also constrained by $\sum_{i=1}^{k} p_i(x) = 1$. There are k-1 variables but k(k-1)/2 constraints. Further, when k>2, k(k-1)/2 > k-1. This means that there may not exist p_i exactly satisfying all constraints. In this case, one must use the estimation

$$\hat{r}_{ii}(x) = \hat{p}_i(x) / (\hat{p}_i(x) + \hat{p}_i(x))$$

In order to get a good estimation, Hasti and Tibshirani use the average Kullback-Leibler distance between $r_{ij}(x)$ and $\hat{r}_{ij}(x)$ as the closeness criterion, and find the *P* that maximizes the criterion.

$$l(P) = \sum_{i < j} n_{ij} [r_{ij} \log \frac{r_{ij}}{\hat{r}_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \hat{r}_{ij}}]$$

this is equivalent to minimizing the negative log-likelihood:

$$l(\hat{P}) = -\sum_{i < j} n_{ij} [r_{ij} \log \frac{\hat{p}_i}{\hat{p}_i + \hat{p}_j} + (1 - r_{ij}) \log \frac{\hat{p}_j}{\hat{p}_i + \hat{p}_j}]$$

where n_{ij} is the number of training samples used to train the binary classifier that predicts r_{ij} .

This can be solved by a simple iterative algorithm:

- 1. Initialize $P = [p_1, p_2, ..., p_k]$ with random $p_i(x) > 0, i = 1, 2, ..., k$.
- 2. Repeat (j=1, 2, ..., k, 1, 2,...) until convergence:
 - (a) Calculate corresponding $\hat{r}_{ij}(x) = p_i(x)/(p_i(x) + p_j(x))$.

(b) Calculate
$$\hat{P} = [p_1, ..., p_{j-1}, \frac{\sum_{i:i \neq j} n_{ji} r_{ji}}{\sum_{i:i \neq j} n_{ji} \hat{r}_{ji}} p_j, p_{j+1}, ..., p_k]^T$$
.

(c) Update
$$P = \hat{P} / \sum \hat{p}_i$$
.

Pairwise coupling is a special case of ECOC. With some generalization, Hastie and Tibshirani's pairwise strategy can be extended to ECOC with any arbitrary code matrix *C*. A close look at the ECOC code matrix reveals that it actually divides the samples from different classes into two groups for each binary classifier: the ones labeled "+1" and the ones labeled "-1". In this sense, ECOC with any arbitrary code matrix is equivalent to pairwise group coupling. Therefore Hastie and Tibshirani's results can be generalized to cases where each binary problem involves data in two "teams" (two disjoint subsets of samples), *i.e.*, instead of comparing two individuals, we can compare two groups that are generated by ECOC and estimate the individual probabilities through the group comparisons.

Assuming an arbitrary code matrix C, for each column i of C results in:
$$r_i(x) = P(class \in I_i^+ \mid class \in I_i^+ \cup I_i^-, X = x) = \frac{\sum_{class \in I^+} p_{class}(x)}{\sum_{class \in I^+ \cup I^-} p_{class}(x)}$$

where I_i^+ and I_i^- are the set of classes for which the entries in the code matrix C(*, i) = +1 and C(*, i) = -1. If we define:

$$q_i = \sum_{j \in I_i^+ \cup I_i^-} p_j \qquad q_i^+ = \sum_{j \in I_i^+} p_j \qquad q_i^- = \sum_{j \in I_i^-} p_j$$

Similar to pairwise comparison, the negative log-likelihood must be minimized:

$$\min_{P} \quad l(P) = -\sum_{i=1}^{l} n_i [r_i \log \frac{q_i^+}{q_i^+} + (1 - r_i) \log \frac{q_i^-}{q_i^-}]$$

where *n_i* is the number of training samples of the binary classifier that corresponds to the *i*th column of the code matrix. Above equation can be solved by a slightly more complex iterative algorithm listed below. This algorithm is equivalent to a special case on probability estimation of Huang *et al.*'s Generalized Bradley-Terry Model (T. K. Huang, R. C. Weng, and C. J. Lin, *A Generalized Bradley-Terry Model: From Group Competition to Individual Skill*, http://www.csie.ntu.edu.tw/~cjlin/papers/generalBT.pdf, 2004). Since the convergence of Generalized Bradley-Terry Model has been proven, the algorithm is also guaranteed to converge.

- 1. Initialize $P = [p_1, p_2, ..., p_k]$ with random $p_i(x) > 0, i = 1, 2, ..., k$.
- 2. Repeat (j=1, 2, ..., k, 1, 2,...) until $\partial l(P) / \partial p_i = 0, i = 1,..., k$ are satisfied.
 - a) Calculate corresponding q_i^+, q_i^-, q_i , *i*=1,2,...*l*.

b) Calculate
$$\hat{P} = [p_1, ..., p_{j-1}, \frac{\sum_{i: j \in I_i^+} \frac{n_i r_i}{q_i^+} + \sum_{i: j \in I_i^-} \frac{n_i (1 - r_i)}{q_i^-}}{\sum_{i: j \in I_i^+ \cup I_i^-} \frac{n_i}{q_i}} p_j, p_{j+1}, ..., p_k]^T$$

c) Update
$$P = \hat{P} / \sum \hat{p}_i$$
.

[0093] To get effective cell recognition with an ANNs it may be necessary to use an improved preprocessing strategy (FLD, multiple "non-cell" subclasses). Although effective, this procedure requires human effort in the selection of "non-cell" subclasses. With SVMs, the above strategy is not necessary. After only PCA preprocessing (which is completely automatic), the SVM has sufficient discrimination without the FLD strategy. A feature of SVMs is that the input vectors are implicitly (and automatically) projected into a high (approaching infinite) dimensional hyperspace, which substantially increases the separability of the classes. This has an effect that is equivalent to FLD preprocessing, making the latter redundant and therefore unnecessary. SVMs and ANNs are highly complicated tools that have diverse requirements. In some circumstances, SVMs will be the tool of choice; in others, ANNs will be optimal.

[0094] In one embodiment, cells in a digitized microscopic image are detected or classified by extracting, for each pixel p in the microscopic image, a sub-image, which consists of the pixel's $m \times m$ neighborhood. The size of m can be adjusted to accommodate cell size. The sub-image is then mapped to a confidence value $C[p] \in [-1,1]$ by the classifier. After all the pixels are processed, a new image (referred to herein as a "confidence map") is created. Pixels in the confidence map are the confidence values of their corresponding sub-images in the original microscope image and form "mountains" with large peaks that represent cell positions. The cell positions/coordinates can then be found by identifying local maxima in mountains. To increase speed, only patches with average pixel intensities above a user-defined value are analyzed further.

EXAMPLES

[0095] In one embodiment the ANN is optimized using an empirical method to determine an upper bound for each layer of the network. Then the optimal number of neurons in the two hidden layers are estimated by independently decreasing the number of

hidden neurons in each layer from the upper bound to 1, and evaluating the generalization properties of the ANN on the test set at each step. To avoid entrapment in a local error minimum, every training session is repeated five times and the best weights were used for each number of hidden neurons.

[0096] FIG. 4 illustrates the generalization properties of the ANN for different numbers of neurons in the first layer, while keeping the size of the second hidden layer constant at five neurons. The mean squared error (the difference between the actual output and the desired output for the samples in the test set) is plotted versus the number of neurons. The error rate improved as the number of hidden neurons was increased, but leveled out at around 40 neurons when preprocessed by PCA and 37 neurons by FLD. This experiment was repeated with the number of neurons in the second layer changed from 1 to 10 and similar but worse results were obtained (not shown). Based on above results, one embodiment uses 40 neurons for PCA preprocessing and 37 for FLD preprocessing in the first hidden layer and 5 neurons in the second hidden layer.

Microsphere Experiments

[0097] In order to study systematically the factors that affect recognition accuracy and to compare the relative efficiencies of PCA and FLD preprocessing, microspheres were used as model cells. The microspheres are very uniform in size, shape and color and are stable over time. This facilitates experimental reproducibility and makes it possible to create ideal scenes in which critical factors can be individually isolated and well controlled. Furthermore, the ability to create scenes with very small within-class variation by using microspheres permits a test of the hypothesis that FLD gives better performance because it controls within-class variation.

[0098] Many experimental factors can affect bright field images of living cells. Among these are variations in focus, illumination, and image noise. These factors could in turn affect cell recognition accuracy. For example, variation in focus is especially

important, since it is often the case that there is no single focal plane that is optimal for all the cells in a microscope field. Another factor that could affect the recognition efficiency is the variation in size. The effects of these factors on recognition accuracy were systematically studied. For all microsphere experiments, recognition was performed as described above. For FLD preprocessing, the dimensionality was reduced to 10. For PCA preprocessing, results are shown when both 10 and 20 principal components were used to improve performance.

[0099] Referring to FIGs. 5a-d, four image groups were created at different focal planes relative to the microsphere equatorial plane to quantify the effects of focus variation, with all other conditions unchanged: (a) focused: the focal plane is at the equator of the microsphere (i.e. 12.5 μ m above the supporting surface); (b) 12.5 μ m: the focal plane is at the supporting surface; (c) 25 μ m: the focal plane is 25 μ m below the equator and is within the plastic bottom of the microplate well and (d) 37.5 μ m: the focal plane is 37.5 μ m below the equator. Two experimental schemes were performed on these images, which are shown in FIGs. 6a-b. In Scheme 1, each method was trained on the first group and then tested on all groups. In Scheme 2, each method was trained on the second group was an interpolation test and on the fourth group was an extrapolation test.

[00100] Referring to FIGs. 7a-e, images were taken under five light intensity levels of the microscope: (a) Intensity level 3: representing extremely weak illumination; (b) Intensity level 4: representing weak illumination; (c) Intensity level 5: representing normal illumination; (d) Intensity level 6: representing strong illumination and (e) Intensity level 7: representing extremely strong illumination. Two experimental schemes were performed using these images the results of which as shown in FIGs. 8a-b. To create the situation of small within-class variation, ANNs based on both PCA and FLD were trained with images only in Intensity level 3 and then tested with all levels in Scheme 1. In Scheme 2, within-class variation was purposely introduced by training the neural network with Intensity levels 4, 5, and 6 together and then tested again with all

levels.

[00101] Referring to FIGs. 9a-b, in the size variation experiment, computer generated images of microspheres with 0%, 5%, 10%, 15% and 20% variations in size were used. Again, two schemes were used to examine the effect of size variation on both PC A and FLD methods, results of which are shown in FIGs. 10 a-b. In Scheme 1, ANNs were trained with only microspheres having 0% size variation and tested to all sizes. In Scheme 2, they were trained using images with both 0% and 15% variation. The patch size used in both schemes was fixed to a value that was big enough to contain the biggest microspheres.

[00102] Referring to FIGs. 11a-e, noise used in noise variation experiments was zero-mean Gaussian noise with different standard deviations. An image set with five groups of images, each have different noise levels was created by adding computer generated noise to original images. The original images (standard deviation equals zero) belonged to the first group. Groups 2, 3, 4 and 5 contained images in which the standard deviations equaled 15, 30, 45 and 60 respectively. The two experimental schemes were: first, both PCA and FLD were applied to only Group 1 and then tested on all groups. Second, the training set was expanded to include both Groups 1 and 4. FIGs. 12a-b show the result of the experiments.

[00103] It can be seen from the results that both PCA and FLD preprocessing performed well if presented with images in the test set, which were selected, from the group(s) used for training. This is reasonable because the classifiers have learned very similar data during the training. Increasing the number of principal components in PCA preprocessing did improve the performance, but it was still no better than that of FLD. Furthermore, both preprocessing methods performed similarly in Scheme 1 for each of the factors studied, but very differently in Scheme 2, with the error rate of FLD being much less than that of PCA in both interpolation and extrapolation tests. The reason lies in that, for Scheme 1, all images in the training set came exclusively from a single group, in which all microspheres had very homogeneous appearance. Therefore, when we extracted

patches from these images and classified them into classes similar to those in FIG. 2, the within-class variations were very small. As expected, FLD was not superior to PCA in this case, since the variation was almost entirely between-class variation. Scheme 2, on the contrary, purposely introduced within-class variation into the training set by using images from different groups. In this case, the FLD method could learn the variation trend from the training set and choose projection directions that were nearly orthogonal to the within-class scatter, projecting away variations in focus, illumination, size and noise; the PCA method could not. Consequently, the generalization ability of the neural network with FLD preprocessing was greatly improved and substantially better than a similar neural network with PCA preprocessing in Scheme 2-type experiments.

Living cell experiments

[00104] Recognition of living cells in digitized microscope images was also studied. The testing images were divided into three groups denoting three different scenarios. Scenario 1 represents the case where cells are almost completely separate, *i.e.*, not aggregated, and the background is clean. Scenario 2 is more complex where most cells are attached to each other and there are trash and debris in the background. Scenario 3 represents the most complex case where most cells are aggregated together and there is more trash and debris in the background. The three microscope images used in the test are shown in FIGs. 13a-c. These images show considerable out of focus blur, cells in clumps occupying multiple focal planes, as well as size variations.

[00105] To obtain a standard for evaluation of our classifiers, three human experts independently evaluated pre-selected microscope images. The experts were asked to identify objects with the normal appearance of a viable cell and to exclude ghosts of cells, *i.e.*, objects having shape and size similar to viable cells but with lower contrast. The three lists generated by the human experts were merged to form one list, called "Human Standard." To be included in the Human Standard list, an object had to be identified as a cell by at least two of the experts.

[00106] In experiments with living cells, images were reduced to 10-dimensional subspaces for both PCA and FLD methods. Results obtained the ANN classifiers were compared to the Human Standard by evaluating sensitivity ("SE") and positive predictive value ("PPV"). The SE of a classifier is defined as the percentage of cells in the reference standard, which are identified by the classifier and the PPV is the percentage of classifier detected cells which are also listed in the reference standard.

[00107] The cell positions detected by the classifier are denoted by white crosses in the images (see FIGs. 14a-b for Scenario 3 result - Scenarios 1 and 2 are not shown). The detected cells were carefully compared with human standard. SE and PPV results of the classifiers are shown in TABLE 1 below and in FIG. 15.

		Scenario 1	Scenario 2	Scenario 3
PCA	SE	97.73%	87.76%	82.5%
	PPV	100%	89.58%	83.02%
FLD	SE	97.73%	95.92%	94.38%
	PPV	100%	95.92%	91.52%

TABLE 1

[00108] The results show that for Scenario 1, both PCA and FLD produced very good results. For example, they both achieved SE values of 97.7% and PPV values of 100%. For Scenario 2, where the image is more complex, the SEs of PCA and FLD dropped to 87.7% and 95.9%, respectively, and PPVs dropped to 89.5% and 95.9%, respectively. These results indicate that the FLD is superior to PCA when the image becomes more complex. This can be seen even more clearly in the very complex case represented by Scenario 3. Here, the SE percentage for FLD is 11.9 greater than that for PCA and the PPV percentage is 8.5 greater.

[00109] As noted previously, the results with microspheres suggest that FLD can better generalize from training sets with a single type of confounding factor. The experiments with living cells described in this section clearly show that FLD gives

superior generalization even when multiple types of confounding factors are present simultaneously. It should also be noted that a close inspection of results yielded by our algorithm suggests that it can distinguish between cell ghosts and viable cells, similar to a human observer.

[00110] The microspheres used in the experiments were 25µm-diameter, dry-red Fluorescent Polymer Microspheres from Duke Scientific (Cat. No. 36-5). The cells used were K562 chronic myelogenous leukemic cells (ATCC; Cat. No. CCL-243) grown at 37.0° C in BM+1/2 TE1+TE2 +10% fetal calf serum (FCS). For microscope observation, cells and microsperes in culture medium were dispensed into polystyrene 96-well microplates, which have well bottoms that are 1 mm thick. An Olympus Model-CK inverted microscope equipped with a 20× planachromat objective and a SONY DSC-F717 digital camera was used to obtain digitized images. The image processing, ANN training and classification programs were written in MATLAB code and implemented in MATLAB Version 6.5.0.180913a (R13) supplemented with Image Processing Toolbox Version 3.2 and Neural Network Toolbox Version 4.0.2. A standard PC equipped with an Intel Pentium 4/1.6G processor with 256-MB RAM was used.

[00111] In order to train and optimize the neural classifier, a set ϕ of 1700 inputoutput pairs ($\phi = \{(I_i, O_i)\}, i=1,2, ..., 1700$) was created by projecting the learning set Ω (containing patches of 25×25 pixels) to linear subspaces using both PCA and FLD methods. Accordingly, the set was also composed of two subsets $\phi = \phi^{\text{pos}} + \phi^{\text{neg}}$. The positive subset $\phi^{\text{pos}} = \{(I_i^{\text{pos}}, 1)\}$ consisted of feature vectors I_i^{pos} computed from the image patches in Ω^{pos} , together with the target output classification value $O_i^{\text{pos}}=1$. The other subset $\phi^{\text{neg}} = \{(I_i^{\text{neg}}, -1)\}$ consisted of feature vectors I_i^{neg} computed from image patches in Ω^{neg} and the target output value $O_i^{\text{neg}}=-1$ of the classifier. This set was further split into a training set of 1400 samples and a test set of 300 samples. The training set was used to modify the weights. The test set was used to estimate the generalization ability.

WO 2006/055413

PCT/US2005/040905

[00112] With the above system using a 25×25 pixel patch, a 640×480 sized image requires a processing time of 1 to 8 minutes, depending on the number of cells present. This is judged to be acceptable for some applications. Substantial speed improvements can be obtained by replacing the MATLAB environment with dedicated neural network software. Further improvement of speed is readily available with more powerful or specialized hardware, such as cluster computing systems.

[00113] One embodiment of the invention uses transmitted light illumination images in conjunction with one or more fluorescence images to automatically generate training sets for training the classifiers. For example, cell viability may be objectively determined using three images of the same microscope field. One image is the transmitted light image which will be analyzed by the pattern recognition algorithm (ANN or SVM). The other two images are images obtained with fluorescent probes, where one probe is specific to viable cells, and the other probe is specific to non-viable cells. A human observer can examine the three images and determine whether a particular cell in the transmitted light image is alive or dead. This process provides a pre-classified sample for use in training and testing. Alternatively, a more automated procedure can be used. Specifically, in one embodiment an image analysis algorithm is used to replace or partially replace the human observer, thereby evaluating the two fluorescence images automatically or semiautomatically and applying information from that evaluation to the transmitted light image. This embodiment therefore provides an automated or semi-automated system for creating the training set. Although this example deals specifically with cell viability, other characteristics identified with fluorescence images can similarly be used to identify and classify objects in a transmitted light image. Further, the concepts described herein can also be employed in the "reverse direction," i.e., images obtained with transmitted light illumination may be used to pre-classify the characteristics of objects in images obtained using fluorescence images. In other words, once the objects in a transmitted light illumination image have been classified and/or localized using the techniques described above, that information may be used to identify, localize or classify information

____ _ _ ...

PCT/US2005/040905

acquired via subsequent fluorescence imaging. For example, such a reverse direction technique can be used to monitor gene expression in real time.

[00114] Each of the embodiments describe herein may be implemented by instructions stored on a computer readable medium and executed on a processor, as depicted in FIG. 16. The computer readable medium 300 may be any medium known in the art for storing instructions, such as a magnetic disk drive, an optical disk drive, magnetic tape, FLASH memory or PROM, among others. In some embodiments, the processor 302 may include a personal computer, a workstation, or any other device known in the art with processing capabilities. The processor 302 reads instructions stored on the computer readable medium 300 and executes those instructions to perform any or all of the functions of the embodiments described herein. In some embodiments, the processor 302 is connected to a machine vision component 304 that generates a digital image as described herein and provides that image to the processor 302. In some embodiments, the processor is connected to an output device 306 such as a CRT or flat panel display that provides results information generated by the processor 302 to the user. In some embodiments, the processor is connected to an input device 308 such as a keyboard, a mouse or other input device known in the art for allowing the user to provide data and control to the processor.

[00115] The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiments are therefore to be considered in respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of the equivalency of the claims are therefore intended to be embraced therein.

APPENDIX A

Multiclass Cell Detection in Bright Field Images of Cell Mixtures with ECOC Probability Estimation

Xi Long^{*,a}, W. Louis Cleveland^b, and Y. Lawrence Yao^a

 ^aMechanical Engineering Department, Columbia University, 220 Mudd., MC4703, New York, NY 10027 USA.
 ^bThe Department of Medicine at St. Luke's Roosevelt Hospital Center and Columbia University New York, NY 10019 USA.
 *Corresponding author. Tel:+01-212-666-2393, Fax: :+01-212-666-2393. Email-address: xl2002@columbia.edu.

Abstract: To achieve high throughput with robotic systems based on optical microscopy, it is necessary to replace the human observer with computer vision algorithms that can identify and localize individual cells as well as carry out additional studies on these cells in relation to biochemical parameters. The latter task is best accomplished with the use of fluorescent probes. Since the number of fluorescence channels is limited, it is highly desirable to accomplish the cell identification and localization task with transmitted light microscopy. In previous work, we developed algorithms for automatic detection of unstained cells of a single type in bright field images [1], [2]. Here we extend this technology to facilitate identification and localization of multiple cell types. We formulate the detection of multiple cell types in mixtures as a supervised, multiclass pattern recognition problem and solve it by extension of the Error Correcting Output Coding (ECOC) method to enable probability estimation. The use of probability estimation provides both cell type identification as well as cell localization.

relative to pixel coordinates. Our approach has been systematically studied under different overlap conditions and outperforms several commonly used methods, primarily due to the reduction of inconsistent labeling by introducing redundancy. Its speed and accuracy are sufficient for use in some practical systems.

Key words: Cell detection, Error Correcting Output Coding (ECOC), Multiclass classification, Support Vector Machines.

1. INTRODUCTION

In high-throughput robotic systems that use optical microscopy, it is essential to replace the human observer with automatic cell-recognition algorithms. A first step towards this goal is to develop algorithms that can distinguish between "Cell" and "Non-cell" objects. This has recently become possible for bright field images of unstained cells in cultures using statistical learning techniques [1]. Even the distinction between viable and non-viable cells in these images can be done with sufficient accuracy for practical applications [2]. To proceed further towards the goal of fully automated microscopy, it is of critical importance to develop algorithms that can sort cell objects into subtypes.

Recognition of cell subtypes is a multiclass classification problem. Although binary classification has been well developed, the problem of multiclass classification is still an ongoing research issue and is not straightforward [3], [4]. Some binary classification methods, such as decision trees, Bayes classifiers, and neural networks, can easily be generalized to monolithic k-way classifiers to handle multiclass classification tasks. However, in cases

where these classifiers are required to learn a very complex decision boundary, they often produce unacceptable accuracy due to the limited representational capability of the learning algorithms and the limited availability of training samples [4], [5]. This has led to a search for alternatives.

Since binary classification has been well developed, a natural alternative to monolithic *k*-way classifiers is to reduce the multiclass problem to a set of binary classification problems. Intuitively, there are two straightforward ways to accomplish this. The first possibility is to apply a classifier between one class and the remaining *k*-1 classes (called "1 vs. all" or "1 vs. rest" method). In the second approach, a classifier is trained between each pair of classes (called the 1 vs. 1 approach). In both cases, we are faced with the possibility of indecisive or contradictory results [3]. Furthermore, error analysis also shows that, in both 1 vs. all and 1 vs. 1 cases, poor results can be produced by the ensemble of classifiers, even though the error rates for individual classifiers are acceptable. For example, in the 1 vs. all case, suppose that *n* binary classifiers are used to output *n* hypotheses h_1 , h_2 ,..., h_n , each with (fractional) training error e_1 , e_2 ,..., e_n , respectively, Guruswami and Sahai have proved that the worst-case training error for the ensemble is $\min\{\sum_{i=1}^{n} e_i, 1\}$; and for randomized situations, the error is $\min\{\frac{n-1}{n}\sum_{i=1}^{n} e_i, 1\}$. In many practical cases, these errors are unacceptably high [6].

This problem has led to more sophisticated strategies that use a high degree of classifier redundancy. In these strategies, a large number of independently constructed classifiers "vote" on the correct class for a test sample. The "bagging" technique, for instance, first generates multiple training sets by sampling with replacement, and then trains a classifier on each generated set [7]. "Boosting" can be viewed as a special case of bagging where the sampling is adaptive, concentrating on misclassified training instances [8]. These approaches have been proven to greatly reduce classification errors in practice. However, available evidence suggests that they can only reduce the variance errors that result from random variation and noise in the learning sample and from random behavior in the learning algorithm. Bias errors, which result from systematic errors of the learning algorithm, can not be reduced by these techniques [9]. In 1995, Dietterich and Bakiri developed the Error Correcting Output Coding (ECOC) method [10], which has been shown to reduce both variance and bias errors [9], [11].

Recent work has shown that ECOC offers further improvement in applications ranging from face verification [12], text classification [13], [14], and cloud classification [15] to speech synthesis [16]. These promising results have led us to explore the use of ECOC in automatic cell recognition algorithms for high throughput robotic systems.

In these systems, it is often necessary not only to identify the class of a cell but also to determine its position relative to pixel coordinates, since tracking and manipulation of cells are often needed. In previous applications of the ECOC method, only classification was considered. Prior ECOC methods therefore must be extended in order to achieve both classification and localization.

WO 2006/055413

In our previous work, which considered binary classification problems, we successfully achieved both classification and localization by a pixel patch decomposition method. In this method, pixel patches from the original images are mapped to "confidence values" that reflect the estimated class probability. Patches containing centered cells give the highest probability and thereby provide the localization (see below) [1],[2]. Here, we develop an ECOC-based probability estimation algorithm to enable the pixel patch decomposition technique to be used in multiclass classification problems.

Currently, a popular approach for multiclass probability estimation is proposed by Hastie and Tibshirani [17], In this method, the multiclass probability estimation is obtained by coupling results from pairwise (1 vs. 1) comparisons. In this paper, we generalize their approach to cases where each binary problem involves comparison of data from two "teams" (of classes) that are generated by ECOC. The class probability of each individual class is estimated through team comparisons. In one implementation using this new algorithm with Support Vector Machines (SVMs) [5], [18] as base binary classifiers, we are able to subtype and localize cells in bright field images of cell mixtures prepared by mixing cells from three different cell lines. The experimental results suggest that our algorithm can reduce classification errors to the point where some practical applications are possible.

2. MATERIALS AND EXPERIMENTAL CONDITIONS

Both microspheres and living cells were used for training and testing classifiers. The

microspheres were 25µm-diameter, red and 40µm-diameter, green fluorescent polymer microspheres from Duke Scientific (Cat. No.36-5, 36-7). The cell lines were K562 (human chronic myelogenous leukemic cells, ATCC; Cat. No.CCL-243), CR10.PF.G cells (obtained from D. J. Volsky) and EAT cells (Ehrlich Ascites Tumor cells, ATCC; Cat. No.CCL-77). All cells were grown at 37.0 °C in BM+1/2 TE1+TE2 +10% fetal calf serum (FCS) [19]. For microscope observation, cells in culture medium were dispensed into polystyrene 96-well microplates, which have glass bottoms that are 0.175mm thick. Cell viability was determined by nigrosine staining [20] before and after microscope observation and was greater than 95%.

To obtain an accurate training and testing standard, the fluorescent probes for living cells (CellTracker[™] CAT. No. C2925 and C34552, Molecular Probes) were used to label CR10 (green, fluorescein bandpass) and EAT (red, propidium iodide) cells. K562 cells were unlabelled. Under bright field illumination, these labels are invisible.



(a)

(b)



(c)

Fig. 1 Typical sample images: (a) bright field image; (b) superposition of the bright field and the red fluorescence image; (c) superposition of the bright field and the green fluorescence image.

An Arcturus Pixcell II inverted microscope equipped with a 20× planachromat objective (Numerical Aperture: 0.4) and a Hitachi model KP-D580-S1 CCD color camera was used to obtain digitized images. For each microscope field, a set of three images was acquired (Fig. 1). One image was acquired with bright field illumination and was used for SVM training or testing. Two auxiliary fluorescence images were also acquired to distinguish different cell lines, which were either unlabelled or labeled red or green.

Sixty sets of microscope images were acquired and used in our cell detection experiments. In each experiment, two subsets were extracted: one exclusively for training and another exclusively for testing. Ambiguous objects showing both red and green fluorescence were manually deleted. The deleted objects were a very small percentage of the total number of cells.

The computer programs were written in MATLAB and C++. Our algorithm was implemented with the LIBSVM version 2.5 [21], which was compiled as a dynamic link

library for MATLAB. All experiments were implemented in the environment of MATLAB Version 6.5.0.180913a (R13) supplemented with Image Processing Toolbox Version 3.2. A standard PC equipped with an Intel Pentium 4/2.8G processor and 256-MB RAM was used.

3. OVERALL FRAMEWORK FOR CELL DETECTION

In this section, an ECOC-based cell detection framework for bright field images of cultured cells is presented. The framework employs the multiclass classification and probability estimation ability of our proposed algorithm to analyze bright field images of cell mixtures. It permits not only the identification of the desired cells but also gives their locations relative to the pixel coordinates of the primary image. It also uses pixel patches as the primary input data elements. Essentially, the software is taught to classify pixel patches into different classes. Each class corresponds to a single cell type, except for the larger class containing all undesired objects (e.g. background, fragments of cells, trash), denoted as "Non-cell".



Fig. 2 Illustration of the overall multiclass cell detection process with ECOC probability estimation. The essential aspects of this framework are illustrated in Fig. 2. Basically, we first train

an ensemble of SVM classifiers with ECOC. This is done with input vectors that are derived from manually-extracted training patches and are represented as linear combinations of feature vectors derived in Principal Component Analysis (PCA) preprocessing [1],[22].

For each pixel p in the testing image (excluding pixels in the margin around the edges), a pixel patch centered at that pixel is extracted and represented in the same way as that in training process. The probability that this extracted patch belongs to each class is calculated by ECOC probability estimation. For each class corresponding to a cell type, this probability is then used as a "confidence value" $C[p] \square [0,1]$ in a "confidence map" for that cell type. Pixels in each confidence map are the confidence values of their corresponding patches in the original image and form "mountains" with large peaks representing a high probability of presence of the corresponding cell type. A given peak in a confidence map with the highest peak at that location gives the assignment of class membership. Localization is provided by the pixel coordinates of the highest peak. It should be pointed out that generating a confidence map for the "Non-cell" class is unnecessary in our case since localization of the non-cell objects is not important for us.



Fig.3 Coarse grid search, grid size = 1.



Fig. 4 Fine grid search, grid size = 0.25.

As has been mentioned above, in the ECOC approach, binary classifiers have to be trained as the base classifiers. The choice of base classifier can be arbitrary. In this work, we used Support Vector Machines (SVM) [5], [18] with the RBF kernel $K(x, y) = e^{-\gamma ||x-y||^2}$. The SVM classifier in our experiment is implemented by modifying LibSVM [21]. The regularization parameter C and the kernel parameter γ are optimized using a two-step "grid-search" method for each classifier [21]. In the first step, a coarse grid-search with a grid size of 1 was used to localize a Region of Interest (ROI) containing the optimal values

(shown in Fig. 3). In the second step, a fine grid-search over the ROI with a grid size of 0.25 was used to give more precise values for C and γ . The result is shown in Fig. 4.

4. EXTENDING ECOC FOR PROBABILITY ESTIMATION

4.1 Brief summary of ECOC

As noted above, our cell identification and localization algorithm requires mapping each pixel patch in the image into a set of three "confidence values", which reflect the estimated class probabilities. For this mapping, multiclass probability estimation is needed. Since the standard ECOC method simply assigns a class label to each sample (i.e., they do not output the conditional probability of each class P(class = c | X = x) given a sample *x*), we need to extend it to enable probability estimation. Our development of the probability estimation algorithm requires a consideration of some of the fundamental aspects of ECOC, which are briefly described in this section. A detailed introduction to ECOC can be found in [10] and [23].

The ECOC approach essentially proceeds in two steps: training and classification. In the first step, the multiclass classification problem is decomposed into training l binary classifiers on l dichotomies of the instance space. Assuming k classes and l classifiers, each such decomposition can be represented by a coding matrix $C \in \{-1,0,+1\}^{k \times l}$ which specifies a relation between classes and dichotomies. If C(i, j) = +1 (or C(i, j) = -1) then the samples belonging to class i ($1 \le i \le k$) are considered to be positive (or negative) samples

for training the *j*th $(1 \le j \le l)$ binary classifier, f_j . If C(i, j) = 0, then the samples belonging to class *i* are not used in training f_j . Thus a binary learning problem is built for each column of the matrix. Each class *i* is encoded by the *i*th row of the matrix *C*. This codeword is denoted by C_i . To classify a new instance *x*, the vector formed by the output of the classifiers $F(x) = (f_1(x), f_2(2)..., f_l(x))$ is computed and is assigned to the class whose codeword C_i is closest to F(x). In this sense, the classification can be seen as a decoding operation:

Class of input
$$x = \arg \min_{i=1}^{k} d(C_i, F(x))$$
(1)

where d() is the decoding function.

Different decoding functions have been reported in the literature. For example, Dietterich and Bakiri initially used a simple Hamming distance [10]. In the case where margin-based classifiers are used, Allwein *et al.* showed the advantage of using a loss-based decoding function [23]. The loss-based function is typically a non-decreasing function of the margin and thus weights the confidence of each classifier according to the margin. However, no formal results exist that suggest the optimal choice of the decoding function. In this paper, we tried two of the most commonly used loss-based functions:

L1 norm based function
$$d(C_i, F(x)) = \sum_{j=1}^{l} |C_{ij} - F_j(x)|$$
(2)

and L2 norm based function
$$d(C_i, F(x)) = \sum_{j=1}^{l} (C_{ij} - F_j(x))^2$$

(3)

Because the codewords come from an error-correcting code, the ECOC method introduces redundancy into the system by training decision boundaries multiple times. Even if some of the individual classifiers were wrong for a specific instance x, the ECOC method can still classify x in the right class. ECOC, therefore, can greatly increase the classification accuracy. It is worth noting that both 1 vs. all and 1 vs. 1 are special cases of the ECOC framework. 1 vs. all is equivalent to linear decoding with a coding matrix whose entries are always -1 except diagonal +1 entries. 1 vs. 1 is also equivalent to Hamming decoding with the appropriate coding matrix.

4.2 Extension of standard ECOC for probability estimation

In this section, we modify the standard ECOC method to enable probability estimation. Our new algorithm is an extension of the pairwise coupling method introduced by Hastie and Tibshirani [17]. It should also be noted that, while this work was in progress, Huang *et al*. independently developed a very similar algorithm based on the same strategy and formulated it as "Generalized Bradley-Terry Model" [24]. To our knowledge, they have not applied their algorithm to practical applications.

4.2.1 Hastie – Tibshirani method for pairwise coupling

The Hastie and Tibshirani's pairwise coupling method can be briefly described as

follows. Assume that after training a classifier using the samples from class *i* (labeled +1) and samples from class *j* (labeled -1), the pairwise probability estimation for every class *i* and j ($i \neq j$) is $r_{ij}(x)$. According to the Bradley-Terry (BT) model [24], $r_{ij}(x)$ is related to the class posterior probabilities $p_i = P(class=i | X=x)$ (i=1,2,...k):

$$r_{ij}(x) = P(class = i \mid class = i \cup class = j, X = x) = p_i(x) / (p_i(x) + p_j(x))$$
(4)

Note that p_i is also constrained by $\sum_{i=1}^{k} p_i(x) = 1$. There are k-1 variables but k(k-1)/2

constraints. When k>2, k(k-1)/2>k-1. This means that there may not exist p_i exactly satisfying all constraints. In this case, one must use the estimation

$$\hat{r}_{ij}(x) = \hat{p}_i(x) / (\hat{p}_i(x) + \hat{p}_j(x))$$
(5)

In order to get a good estimation, Hasti and Tibshirani use the average Kullback-Leibler distance between $r_{ij}(x)$ and $\hat{r}_{ij}(x)$ as the closeness criterion, and find the *P* that maximizes the criterion.

$$l(P) = \sum_{i < j} n_{ij} [r_{ij} \log \frac{r_{ij}}{\hat{r}_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \hat{r}_{ij}}]$$
(6)

this is equivalent to minimizing the negative log-likelihood:

$$l(\hat{P}) = -\sum_{i < j} n_{ij} [r_{ij} \log \frac{\hat{p}_i}{\hat{p}_i + \hat{p}_j} + (1 - r_{ij}) \log \frac{\hat{p}_j}{\hat{p}_i + \hat{p}_j}]$$
(7)

where n_{ij} is the number of training samples used to train the binary classifier that predicts r_{ij} . This can be solved by a simple iterative algorithm:

- 1. Initialize $P = [p_1, p_2, ..., p_k]$ with random $p_i(x) > 0, i = 1, 2, ..., k$.
- 2. Repeat (*j*=1, 2, ..., *k*, 1, 2,...) until convergence:

(a) Calculate corresponding $\hat{r}_{ij}(x) = p_i(x)/(p_i(x) + p_j(x))$.

(b) Calculate
$$\hat{P} = [p_1, ..., p_{j-1}, \frac{\sum_{i:i \neq j} n_{ji} r_{ji}}{\sum_{i:i \neq j} n_{ji} \hat{r}_{ji}} p_j, p_{j+1}, ..., p_k]^T$$
.

(c) Update $P = \hat{P} / \sum \hat{p}_i$.

4.2.2 Generalization of Hastie – Tibshirani method

It has been mentioned above that pairwise coupling is a special case of ECOC. With some generalization, Hastie and Tibshirani's pairwise strategy can be extended to ECOC with any arbitrary code matrix *C*. A close look at the ECOC code matrix reveals that it actually divides the samples from different classes into two groups for each binary classifier: the ones labeled "+1" and the ones labeled "-1". In this sense, ECOC with any arbitrary code matrix is equivalent to pairwise group coupling. Therefore we can generalize Hastie and Tibshirani's results to cases where each binary problem involves data in two "teams" (two disjoint subsets of samples), i.e. instead of comparing two individuals, we can compare two groups that are generated by ECOC and estimate the individual probabilities through the group comparisons.

Assuming an arbitrary code matrix C, for each column i of C, we have

$$r_i(x) = P(class \in I_i^+ \mid class \in I_i^+ \cup I_i^-, X = x) = \frac{\sum_{class \in I^+} p_{class}(x)}{\sum_{class \in I^+ \cup I^-} p_{class}(x)}$$
(8)

where I_i^+ and I_i^- are the set of classes for which the entries in the code matrix C(*, i)=+1and C(*, i)=-1. If we define

$$q_{i} = \sum_{j \in I_{i}^{+} \cup I_{i}^{-}} p_{j} \qquad q_{i}^{+} = \sum_{j \in I_{i}^{+}} p_{j} \qquad q_{i}^{-} = \sum_{j \in I_{i}^{-}} p_{j}$$
(9)

Similar to pairwise comparison, we need to minimize the negative log-likelihood

$$\min_{P} \quad l(P) = -\sum_{i=1}^{l} n_i [r_i \log \frac{q_i^+}{q_i^+} + (1 - r_i) \log \frac{q_i^-}{q_i^-}]$$

(10)

where n_i is the number of training samples of the binary classifier that corresponds to the *i*th column of the code matrix. Above equation can be solved by a slightly more complex iterative algorithm listed below. This algorithm is equivalent to a special case on probability estimation of Huang *et al.*'s Generalized Bradley-Terry Model in [24]. Since the convergence of Generalized Bradley-Terry Model has been proven [24], the algorithm is also guaranteed to converge.

- 1. Initialize $P = [p_1, p_2, ..., p_k]$ with random $p_i(x) > 0, i = 1, 2, ..., k$.
- 2. Repeat (j=1, 2, ..., k, 1, 2, ...) until $\partial l(P) / \partial p_i = 0, i = 1, ..., k$ are satisfied.
 - a) Calculate corresponding $q_i^+, q_i^-, q_i, i=1,2,...l$.

b) Calculate
$$\hat{P} = [p_1, ..., p_{j-1}, \frac{\sum_{i:j \in I_i^+} \frac{n_i r_i}{q_i^+} + \sum_{i:j \in I_i^-} \frac{n_i (1-r_i)}{q_i^-}}{\sum_{i:j \in I_i^+ \cup I_i^-} \frac{n_i}{q_i}} p_j, p_{j+1}, ..., p_k]^T.$$

c) Update $P = \hat{P} / \sum \hat{p}_i$.

5. EXPERIMENTS WITH ARTIFICIAL DATA

To gain insight into the factors that affect the classification accuracy of our algorithm, we have carried out experiments with artificial 2D data generated by the Matlab random

functions. Unlike actual data vectors that have high dimensionality (e.g. 39*39, see below), the artificial 2D data vectors generate results that can be graphically represented and intuitively interpreted.

Four different sets of artificial data have been used (Fig. 5 (a),(b),(c),(d)). Data set 1 represents a simple scenario, where the classes are well separated. Data sets 2-4 represent progressively more difficult scenarios, with data set 4 having a very large class overlap.



(a)

(b)



Fig. 5 Data sets used in the simulation experiment. Class number: 4; Sample number in each class: 300; Class distribution: Normal distribution. The four data sets have same covariance but different mean for each class.

(d)

(c)

	μ_1	μ_2	μ_3	μ_4
Data set 1	$\begin{bmatrix} -3 & -3 \end{bmatrix}^{T}$	[3 3] ^T	$[3 -3]^{T}$	$\begin{bmatrix} -3 & 3 \end{bmatrix}^{T}$
Data set 2	$\begin{bmatrix} -2.5 & -2.5 \end{bmatrix}^{\mathrm{T}}$	$[2.5 \ 2.5]^{\mathrm{T}}$	$[2.5 - 2.5]^{\mathrm{T}}$	$\begin{bmatrix} -2.5 & 2.5 \end{bmatrix}^{\mathrm{T}}$
Data set 3	$[-2 - 2]^{T}$	$[2 \ 2]^{T}$	$[2 - 2]^{T}$	$\begin{bmatrix} -2 & 2 \end{bmatrix}^{T}$

Data set 4	$\begin{bmatrix} -1.5 & -1.5 \end{bmatrix}^{\mathrm{T}}$	$[1.5 1.5]^{\mathrm{T}}$	$[1.5 - 1.5]^{\mathrm{T}}$	$\begin{bmatrix} -1.5 & 1.5 \end{bmatrix}^{\mathrm{T}}$
	~ _	-	_	

Table 1 Mean vectors used to generate artificial data for Datasets 1, 2, 3 and 4.

The artificial data sets used in this section were constructed as follows. We first constructed a 2D data set which consists of four different multivariate normal distribution classes. After creating the first data set, three different data sets, each with the same covariance and sample numbers but different mean vectors were also constructed. Each class was given 300 samples. The covariance matrices of the four classes were (same for all data sets): $\sigma_1 = \begin{bmatrix} 1.5 & 0.4 \\ 0.4 & 1.5 \end{bmatrix}$, $\sigma_2 = \begin{bmatrix} 3 & 0.1 \\ 0.1 & 4 \end{bmatrix}$, $\sigma_3 = \begin{bmatrix} 2 & 0.2 \\ 0.2 & 3 \end{bmatrix}$ and $\sigma_4 = \begin{bmatrix} 3 & 0.8 \\ 0.8 & 8 \end{bmatrix}$. The mean vectors of the four classes for each data set are summarized in Table 1.

5.1 Reconstruction of probability distribution from ECOC probability estimation

To evaluate directly our proposed ECOC probability estimation algorithm, we used it to estimate the known probability distributions of the above artificial 2D data sets. In this experiment, ECOC was implemented with a sparse matrix that was selected from 10000 randomly generated 4×10 matrices. To select the optimum matrix in the set of 10000, we calculated the minimum Hamming distance between all pairs of the rows for each matrix. The matrix with the biggest minimum distance was chosen [21]. Since the four artificial 2D data sets have known distributions, the ideal probability distributions of the classes can be easily calculated. Fig. 6 plots the ideal class probability of the samples in Data set 2 against their coordinates. The ECOC-reconstructed class probability distribution is shown in Fig. 7. As one can see from the figures, the reconstructed probability distribution matches the ideal distribution very well.

Fig. 8 gives a quantitative evaluation of the mean square error (MSE) of the ECOC probability estimation. This result is shown in comparison with the result provided by the pairwise coupling method proposed of Hastie and Tibshirani [17]. As indicated in the figure, our ECOC algorithm is superior to the pairwise coupling method for three of the four test classes. Therefore, ECOC probability estimation has a higher overall accuracy.



Fig.6 Ideal class probability distribution of Data set 2.



Fig.7 Class probability distribution of Data set 2, estimated by our ECOC probability estimation method.



Fig. 8 Comparison of probability estimation errors of our ECOC probability estimation and the pairwize (1 vs. 1) probability estimation by Hastie and Tibshirani. The ideal probability distribution was used as reference.

5.2 Comparison of extended ECOC with other methods

WO 2006/055413

PCT/US2005/040905

Using the above artificial 2D data sets, we systematically compared the proposed ECOC probability estimation method with other widely used approaches: 1) 1 vs.all; 2) 1 vs. 1 (pairwise coupling by Hastie and Tibshirani); 3) ECOC with Hamming decoding; 4) ECOC with L1-Norm based decoding and 5) ECOC with L2-Norm based decoding. We used randomly generated sparse code matrices as described in Section 5.1 for all ECOC-based methods. For inconsistent labels (ties and contradictory votes), we adapted the strategy described in [3] and randomly chose labels for them. Results are shown in Fig. 9. As expected, ECOC-based methods are generally superior to non-ECOC approaches, i.e. 1 vs. all and 1 vs. 1. Even within ECOC-based methods, the extended ECOC with probability estimation method produces the highest classification accuracy. Finally and most interestingly, all candidate methods perform very well on Data set 1, which represent a very simple case. However, as the scenario gets more and more complex, ECOC-based methods show a greater advantage over other approaches.



Fig. 9 Classification accuracy of different methods on the artificial data sets. The methods used are 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with Hamming decoding; 4) ECOC with L1-Norm based decoding; 5) ECOC with L2-Norm based decoding; 6) ECOC with probability estimation.

We hypothesized that the superiority of ECOC methods was largely due to the fact that these methods generated more decision boundaries, which can greatly reduce inconsistent labeling areas, i.e. areas in which sample points can not be consistently labeled using the majority voting strategy. To verify this hypothesis, the decision boundaries of different candidate methods were plotted and compared (Fig. 10). Figs. 10 (a) and (b) show decision boundaries of Data set 1 that are generated by the 1 vs. all and ECOC probability estimation method, respectively. Figs. 10 (c) and (d) show those of Data set 4. One can see that although there exist many areas with inconsistent labeling in Fig. 10 (a), most areas are very close to class interfaces. Since there is little overlap in Data set 1, few sample points fall into these areas. Therefore, 1 vs. all method works almost as well as ECOC probability estimation, which dramatically eliminates the inconsistent labeling areas (Fig. 10 (b)). On the other hand, since there is a large overlap in Data set 4, a great proportion of the sample points fall into the inconsistent labeling areas when the 1 vs. all method is used. In this case, ECOC probability estimation outperformed the 1 vs. all method by a very large margin. Our hypothesis is very consistent with the experimental results shown in Fig. 9.



(a)

(b)



Fig. 10 Examples of decision boundaries generated by different methods on Data sets 1 and 4. (a) 1 vs. all on Data set 1; (b) ECOC probability estimation on Data set 1; (c) 1 vs. all on Data set 4; (d) ECOC probability estimation on Data set 4.

(d)

6. EXPERIMENTS WITH BRIGHT FIELD IMAGES OF LIVING CELLS

(c)

In this section, we evaluate quantitatively the extended ECOC-based cell detection method for bright field images of cell mixture prepared by mixing cells from three different cell lines. The overall framework of this approach has been described in Section 3. In what follows, the detailed experiment is described in steps. The experimental result is also quantitatively analyzed.

6.1 Pixel patch extraction and construction of preclassified training set

Since individual cells typically occupy only a small percentage of total image area, it is advantageous to decompose an image using pixel patches that just large enough to contain the largest cells in the image. In actual experiments, 39×39 pixel patches centered at all possible locations in the 640×480 microscope image were extracted (except in the 20-pixel margin around the edges). Our experiments indicate that performance is not very sensitive to small variations in patch size, e.g. a patch size of 37×37 produced similar results (data not shown). Since many locations in the image are uniform background, a "mask" was created to exclude these patches. Essentially, the "mask" eliminated all pixel patches whose average pixel intensities were below a user-chosen threshold.

A training set was created with the aid of an interactive program that displays the digitized microscope images and allows a user to select the locations of cell centers with a mouse cursor after manual comparison of bright field and fluorescence images. For each cell type, the pixel patches extracted from the selected cell locations were preprocessed by PCA [1], [22] and used as input vectors of that class. The pixel patches in the "Non-cell" class were then generated automatically by extracting all the pixel patches whose centers were $r \ge 8$ pixels away from any of the manually selected cell locations. The value of r was empirically chosen in relation to the sizes of cells and pixel patches. PCA preprocessing was used to reduce dimensionality to 10 for all input vectors. After all input vectors are preprocessed, each attribute of the PCA-preprocessed vectors was linearly scaled to the range [-1, +1]. The
main advantage of scaling is to avoid computational difficulties and to avoid the dominance of attributes with greater numeric ranges over those with smaller numeric ranges [21]. Finally, the classes were labeled with ordinal numbers.

6.2 ECOC training

We followed the procedure described in the simulation experiment and used randomly generated sparse code matrices for all ECOC-based methods in this section. For each binary SVM classifier, the parameters are independently optimized following the aforementioned two-step grid search procedure. During the process of binary classifier training, the Compensatory Iterative Sample Selection (CISS) algorithm [2], a new SVM training procedure which we developed previously, was employed to address the imbalance problem caused by the large "Non-cell" sample set. This algorithm maintains a fixed-size "working set", in which the training samples are kept balanced by iteratively choosing the most representative training samples for the SVM. These samples are close to the boundary and are therefore more difficult to classify. This scheme can make the decision boundary more accurate, especially when applied to difficult scenarios.

6.3 Identification and localization of living cells in bright field images



(c)

Fig. 11 Sample images for living cell experiment. (a) Scenario 1: mixture of 2 types of microspheres and 1 type of cells; (b) Scenario 2: mixture of 1 type of microspheres and 2 types of cells; (c) Scenario 3:

mixture of 3 types of cells.

In order to examine the effect of our algorithm on images with different levels of complexity, three different scenarios were created. In Scenario 1, both red and green fluorescent microspheres were used as two types of model cells and mixed with the K562 cells. Since the microspheres have obviously different size, color and texture from living cells, this scenario represents a very simple case. Scenario 2 is more complex since it is the mixture of only one type of microspheres (red) and cells from two cell lines (K562 and CR10.PF.G). Scenario 3 represents the most complex case where three kinds of living cells (K562, CR10.PF.G and EAT) were mixed without the addition of any microspheres. Typical images from these three scenarios are shown in Fig. 11. For each scenario, there is a total of 4 classes: one for each of the desired objects (microspheres or cells) and one for all objects that are neither cells nor microspheres (the "non-cell" class).

An ensemble of SVM classifiers was trained and tested on each scenario. For each ensemble, testing samples were from the same scenario as the training samples. However, none of the samples used for training were used for testing.



Fig. 12 Classification accuracy of different methods on living cell testing sets. The methods used are 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with Hamming decoding; 4) ECOC with L1-Norm based decoding; 5) ECOC with L2-Norm based decoding; 6) ECOC with probability estimation.

After training, we first tested the classifier ensembles on manually extracted pixel patches. This is done with 3 testing sets. Each set has 2000 manually extracted pixel patches from one scenario. Testing set 1 consisted of pixel patches of 500 K562 cells, 500 green fluorescent microspheres, 500 red fluorescent microspheres and 500 background from Scenario 1. Testing set 2 consisted of pixel patches of 500 K562 cells, 500 CR10 cells, 500 red fluorescent microspheres and 500 background from Scenario 2. Testing set 3 consisted of pixel patches of 500 K562 cells, 500 cR10 cells, 500 red fluorescent microspheres and 500 background from Scenario 2. Testing set 3 consisted of pixel patches of 500 K562 cells, 500 CR10 cells, 500 cR10 cells, 500 EAT cells and 500 background from Scenario 3. The classification accuracy is shown in comparison with other candidate methods in Fig. 12.

The classifier ensembles were also applied to pixel patches obtained by automatic pixel

patch decomposition of entire microscope images (640×480) described in Section 3. Fig. 13 shows the confidence maps for Fig. 11 (c). The range of the confidence value ([0,1]) in the confidence maps has been linearly scaled to [0,255] for grayscale representation. In Figs. 14, 15 and 16, the cell and microsphere positions detected are denoted by different symbols (diamond, square and cross, one for each class) in the image.



(a)

(b)



(c)

Fig. 13 Confidence maps for Fig. 11 (c). (a) confidence map for CR10 cells; (b) confidence map for EAT cells; (c) confidence map for K562 cells. The confidence values are linearly scaled to 0~255 for display.

Statistical cell detection results for whole microscope images in Scenarios 1, 2 and 3 are summarized in Figs. 17, 18 and 19, respectively. For each scenario, ten testing images (640×480) were used. We employed a "Free-response Receiver Operating Characteristics" method (FROC) [25] with the average false positive (FP) number of all cell types in each image and the average sensitivity (true positive percentage, i.e., the percentage of cells that are identified correctly) of all cell types as performance indexes. As described above, the cell positions are identified as "peaks" of the "mountains" in the confidence maps. This requires a user-defined threshold for the definition of "peak". The FROC curve plots the relationship of false positives and sensitivity as a function of the threshold (not explicitly represented in the plot). In a practical application, a suitable threshold can then be selected to achieve the required behavior. Generally speaking, the bigger the area under the curve, the better the

result is. A total of 3 methods were compared in the experiment: 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with probability estimation.



Fig. 14 Detecting result of the image in Scenario 1 using SVM with ECOC probability estimation. The positions detected are denoted by black symbols in the image. Diamond: green fluorescent microspheres; Square: red fluorescent microspheres; Cross: K562 cells.



Fig. 15 Detecting result of the image in Scenario 2 using SVM with ECOC probability estimation. The positions detected are denoted by black symbols in the image. Diamond: CR10 cells; Square: red fluorescent microspheres; Cross: K562 cells.



Fig. 16 Detecting result of the image in Scenario 3 using SVM with ECOC probability estimation. The cell positions detected are denoted by white symbols in the image. Diamond: CR10 cells; Square: EAT cells; Cross: K562 cells.



Fig. 17. FROC plots of different candidate methods when applied to Scenario 1: 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with probability estimation. The testing set includes 10 images.



Fig. 18. FROC plots of different candidate methods when applied to Scenario 2: 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with probability estimation. The testing set includes 10 images.



Fig. 19. FROC plots of different candidate methods when applied to Scenario 3: 1) 1 vs.all; 2) 1 vs. 1 by Hastie and Tibshirani; 3) ECOC with probability estimation. The testing set includes 10 images.

Results with both manually and automatically extracted pixel patches show that for Scenario 1, a very easy case, all methods produce very good results. For Scenario 2, where the images are more complex, our ECOC probability estimation method (and other ECOC-based methods) starts to show some advantage. A much greater advantage is seen in the very difficult case represented by Scenario 3. For example, in Scenario 3, if the average false positive acceptance number in each image is set at 1, ECOC probability estimation achieves a sensitivity of 84.5%, which is 4 percentage points greater than that of 1 vs. 1, and 15 points greater than that of the 1 vs. all approach. The result closely parallels that obtained in the simulation experiments with artificial data as shown in Fig. 9.

As noted previously, our results with the artificial data shown in Figs. 9 and 10 suggest that ECOC-based methods can greatly reduce the inconsistent labeling by introducing redundancy, and therefore can partition the sample space more accurately than other methods. The experimental results with living cells described in this section add further support to this claim.

It should also be noted that a close inspection of results yielded by our algorithm suggests that it can distinguish between different cell lines according to the subtle difference in the cell appearance, similar to a human observer. For example, to a human observer, the CR10 cells in the testing images are relatively small, and are rough-looking in texture. The K562 cells and the EAT cells are about the same size. However, the edge and texture of the K562 cells are slightly smoother than those of EAT cells. Our experimental results suggest that our algorithm can actually make these subtle distinctions, and thereby emulate a human expert quite well.

With regard to the processing speed, when our current method is used with a 39×39 pixel patch, a 640×480 image requires a processing time of 5-15 minutes, depending on the number of objects present in the image. However, as yet, optimization of speed has not been attempted.

7. CONCLUSION

An extended ECOC algorithm for multiclass classification has been described. Unlike prior ECOC methods, which only assign class labels, this algorithm also calculates class probabilities for each sample. This extension in conjunction with a strategy developed in our previous studies not only facilitates assignment of class membership but also permits localization of identified objects relative to pixel coordinates. Our algorithm therefore makes possible both subtyping and localization of unstained cells in bright field images of cell mixtures. Our extended ECOC strategy has been shown to be superior to many other currently existing approaches, especially for complex scenarios. The speed and accuracy of our multiclass cell detection framework suggest that it can be useful in some systems that require automatic subtyping and localization of cells in cell mixtures.

In this study, our goal has been simply to explore the efficacy of the ECOC approach using a conveniently available model system. The successful results described in this paper raise possibilities for further optimization. For example, we have considered images of unstained cells obtained with bright field illumination. With respect to microscopy, these

images represent a worst-case scenario. Images that contain much more discriminatory information can be obtained with other, commonly used transmitted light microscopy techniques such as phase contrast, differential interference contrast, and Hoffman modulation contrast. With these inherently more discriminatory images, we believe that our algorithm will achieve higher accuracy and will have wide applicability to systems with diverse combinations of cell types.

8. ACKNOWLEDGMENTS

This work is supported by NIH Grant CA89841.

9. REFERENCE

[1] X. Long, W. L. Cleveland and Y. L.Yao, A New Preprocessing Approach for Cell Recognition, IEEE Transactions on Information Technology in Biomedicine, vol. 9, no.3, 2005, 407-412.

[2] X. Long, W. L. Cleveland and Y. L.Yao, Automatic Detection of Unstained ViableCells in Bright Field Images Using a Support Vector Machine with an Improved TrainingProcedure, Computers in Biology and Medicine, 2006, in press.

[3] D.J.M. Tax and R.P.W. Duin, Using Two-Class Classifiers for Multiclass Classification, ICPR16: Proc. 16th Int. Conf. on Pattern Recognition, Quebec City, Canada, 2002, 124-127.

[4] G. Valentini and F. Masulli, Ensembles of Learning Machines, Neural Nets WIRN

Vietri-02, Series Lecture Notes in Computer Sciences, Springer-Verlag, Heidelberg, Germany, 2002.

[5] V. Vapnik, Statistical Learning Theory, Wiley, 1998.

[6] V. Guruswami and Amit Sahai, Multiclass learning, boosting, and error-correcting codes, Proceedings of the twelfth annual conference on Computational learning theory, Santa Cruz, CA, USA, 1999, 145-155.

[7] L. Breiman, Bagging predictors, Machine Learning, vol. 26, no. 2, 1996, 123-140.

[8] Y. Freund and R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, vol.55, no.1, 1997,119-139.

[9] E. Kong and T. Dietterich, Error-correcting output coding corrects bias and variance, Proceedings of the 12th International Conference on Machine Learning, 1995, 313-321.

[10] T. G. Dietterich and G. Bakiri, Solving Multiclass Learning Problems via Error-Correcting Output Codes, Journal of Artificial Intelligence Research, vol. 2, 1995, 263-286.

[11] G. James and T. Hastie, The error coding method and PiCTs, Journal of Computational and Graphical Statistics, vol. 7, no. 3, 1997, 377--387.

[12] J. Kittler, et al., Face Verification Using Error Correcting Output Codes, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01), 2001, 755-760.

[13] A. Berger, Error-Correcting Output Coding for Text Classification, IJCAI'99: Workshop

on machine learning for information filtering, Stockholm, Sweeden, 1999.

[14] R. Ghani, Using Error-Correcting Codes For Text Classification, Proceedings of ICML-00, 17th International Conference on Machine Learning, 2000, 303-310.

[15] D. Aha and R. Bankert, Cloud classification using error-correcting output codes, Artificial Intelligence Applications: Natural Resources, Agriculture, and Environmental Science, vol. 11, no.1, 1997, 13-28.

[16] G. Bakiri and T. Dietterich, Achieving high-accuracy text-to-speech with machine learning, Data mining in speech synthesis, Kluwer Academic Publishers, Boston, MA, 1999.

[17] T. Hastie and R. Tibshirani, classification by pairwise coupling, Advances in Neural Information Processing Systems, vol. 10, MIT Press, 1998.

[18] C. Burges, A tutorial on Support Vector Machines for pattern recognition, Data Mining and Knowledge Discovery, vol.2, 1998, 122-167.

[19] W. L. Cleveland, I. Wood and B.F. Erlanger, Routine large-scale production of monoclonal antibodies in a protein-free culture medium, Journal of Immunological Methods, vol. 56, 1983, 221-234.

[20] B.B. Mishell *et al.*, Preparation of Mouse Cell Suspensions, in B.B. Mishell and S.M. Shiigi eds: Selected methods in Cellular Immunology, W.H. Freeman and company, New York, 1980.

[21] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

 σ

[22] T.W. Nattkemper, H. Ritter and W. Schubert, A neural classifier enabling

high-throughput topological analysis of lymphocytes in tissue sections, IEEE trans Info. Tech. Biomedicine, vol. 5, no. 2, 2001, 138-149.

[23] E. Allwein, R. Schapire, and Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of Machine Learning Research, vol. 1, 2000, 113–141.

[24] T.-K. Huang, R. C. Weng, and C.-J. Lin, A Generalized Bradley-Terry Model: FromGroupCompetitiontoIndividualhttp://www.csie.ntu.edu.tw/~cjlin/papers/generalBT.pdf, 2004.

[25] D.P. Chakraborty, Maximum likelihood analysis of free-response receiver operating characteristic (FROC) data, Medical Physics, vol.16, 1989, 561-568.

CLAIMS

What is claimed is:

1. A method of identifying one or more objects, wherein each of the one or more objects belongs to a first class or to a second class, the first class being heterogeneous and having C subclasses, the second class being less heterogenous than the first class, comprising:

deriving a plurality of vectors each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

preprocessing each of the plurality of vectors using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the plurality of vectors to M dimensions, wherein M is less than or equal to C; and,

classifying the preprocessed vectors by (i) grouping the preprocessed vectors belonging to any of the C subclasses of the first class into a first set of vectors, and (ii) grouping the preprocessed vectors belonging to the second class into a second set of vectors.

2. The method of claim 1, wherein each of the plurality of vectors includes information mapped from a digital image.

3. The method of claim 2, wherein the information mapped from a digital image includes a pixel patch.

4. The method of claim 1, wherein the preprocessed vectors are classified with an artificial neural network.

5. The method of claim 1, wherein the preprocessed vectors are classified with a support vector machine.

6. The method of claim 5, further including training the support vector machine with

.

a compensatory iterative sample selection technique.

7. The method of claim 6, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

8. A method of identifying one or more objects in a digital image, wherein each of the one or more objects belongs to a first class or to a second class, the first class being heterogeneous and having C subclasses, and the second class being less heterogeneous than the first class, comprising:

deriving a plurality of pixel patches from the digital image, each being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

preprocessing each of the plurality of pixel patches using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the pixel patches to M dimensions, wherein M is less than or equal to C; and,

classifying the preprocessed pixel patches by (i) grouping the preprocessed pixel patches belonging to any of the C subclasses of the first class into a first set of pixel patches, and (ii) grouping the preprocessed pixel patches belonging to the second class into a second set of pixel patches.

9. The method of claim 8, wherein the preprocessed pixel patches are classified with an artificial neural network.

10. The method of claim 8, wherein the preprocessed pixel patches are classified with a support vector machine.

11. The method of claim 10, further including training the support vector machine with a compensatory iterative sample selection technique.

12. The method of claim 11, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

13. The method of claim 8, further including localizing an object in the digital image by identifying a pixel patch having an object that is centered within the pixel patch.

14. The method of claim 8, wherein the first class homogeneous class includes cells, and the second heterogeneous class includes non-cells.

15. A method of identifying and localizing one or more objects, wherein each of the one or more objects belongs to either a first class or a second class, comprising:

deriving a plurality of vectors each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

training a support vector machine with a compensatory iterative sample selection technique; and,

processing the plurality of vectors with the support vector machine, so as to classify each of the plurality of vectors into either the first class or the second class.

16. The method of claim 15, wherein each of the plurality of vectors includes information mapped from a digital image.

17. The method of claim 16, wherein the information mapped from a digital image includes a pixel patch.

18. The method of claim 15, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

19. A method of identifying and localizing one or more objects in a digital image,

wherein each of the one or more objects belongs to either a first class or a second class, comprising:

deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

training a support vector machine with a compensatory iterative sample selection technique; and,

processing the plurality of pixel patches with the support vector machine, so as to classify each of the plurality of pixel patches into either the first class or the second class.

20. The method of claim 19, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

21. The method of claim 19, further including localizing an object in the digital image by identifying a pixel patch having an object that is centered within the pixel patch.

22. A method of identifying one or more cells in a digital image, wherein each of the one or more cells belongs to one of three or more classes, comprising:

deriving a plurality of pixel patches from the digital image, each of the plurality of

pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the three or more classes.

23. The method of claim 22, wherein each of the ensemble of binary classifiers is a support vector machine.

24. The method of claim 22, wherein processing the plurality of pixel patches further includes, for each pixel patch:

calculating a probability that the pixel patch belongs to a particular one of the two or more classes, using an Error Correcting Output Coding probability estimation technique.

25. The method of claim 22, further including localizing a cell in the digital image by identifying a pixel patch having a cell that is centered within the pixel patch.

26. The method of claim 25, wherein localizing a cell further includes:

for each cell, calculating a probability that the pixel patch of that cell belongs to a particular one of the two or more classes, using the Error Correcting Output Coding probability estimation technique;

generating a confidence map for each cell type using the probability calculated for the pixel patch as a confidence value within the confidence map;

comparing peaks in the confidence map for the cell type with corresponding peaks in confidence maps for other cell types, and using a highest peak to assign class membership;

determining localization of the cell corresponding to the highest peak by

determining pixel coordinates of the highest peak.

27. A method of identifying one or more objects, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of vectors, each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique; and,

processing the plurality of vectors with the ensemble of binary classifiers, so as to classify each of the plurality of vectors into one of the three or more classes.

28. The method of claim 27, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

29. A method of identifying one or more objects in a digital image, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the

plurality of pixel patches is an element of an N-dimensional space;

training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique; and,

processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the three or more classes.

30. The method of claim 29, wherein the compensatory iterative sample selection technique comprises:

(a) selecting a first working set of pre-classified objects from a set of training objects;

(b) training the support vector machine with the first working set;

(c) testing the support vector machine with pre-classified objects from the set of training objects not included in the first working set so as to produce a set of correctly classified objects and a set of incorrectly classified objects;

(d) selecting a replacement set of pre-classified objects from the set of incorrectly classified objects, and replacing a subset of the working set with the replacement set;

(e) repeating steps (b), (c) and (d) until the set of incorrectly classified objects does not decrease in size for subsequent iterations of steps (b), (c) and (d).

31. The method of claim 29, further including localizing an object in the digital image by identifying a pixel patch having an object that is centered within the pixel patch.

32. A method of identifying and localizing one or more objects, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of vectors, each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space; training an ensemble of binary classifiers with a compensatory iterative sample

selection technique, using training sets generated with an Error Correcting Output Coding technique;

for each object, calculating a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

generating a confidence map for each class using the probability calculated for the vector as a confidence value within the confidence map;

comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

33. A method of identifying and localizing one or more objects in a digital image, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique;

for each object, calculating a probability that the pixel patch associated with the object belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

generating a confidence map for each cell type using the probability calculated for the pixel patch as a confidence value within the confidence map;

comparing peaks in the confidence map for the cell type with corresponding peaks

in confidence maps for other cell types, and using a highest peak to assign class membership; and,

determining localization of the cell corresponding to the highest peak by determining pixel coordinates of the highest peak.

34. A method of identifying and localizing one or more objects, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of vectors, being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

for each object, calculating a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map;

comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

35. A method of identifying and localizing one or more objects in a digital image, wherein each of the one or more objects belongs to one of three or more classes, comprising:

deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

for each object, calculating a probability that the pixel patch associated with the object belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map;

comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

36. A method of generating a training set of pre-classified objects for training a classifier, comprising:

applying one or more fluorescent markers to a sample containing objects to be classified;

generating one or more fluorescence images of the sample containing objects to be classified;

generating a transmitted light illumination image of the sample containing objects to be classified;

for each of the one or more fluorescence images, superimposing at least a portion of the fluorescence image with a corresponding portion of the transmitted light illumination image; and,

using information from the one or more fluorescence images to identify characteristics of corresponding objects in the transmitted light illumination image, thereby producing a transmitted light illumination image having one or more preclassified objects. 37. The method of claim 36, further including using information from the transmitted light illumination image having one or more pre-classified objects to identify characteristics of corresponding elements in one or more subsequently generated fluorescent images.

38. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of vectors each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an Ndimensional space;

instructions for preprocessing each of the plurality of vectors using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the plurality of vectors to M dimensions, wherein M is less than or equal to C; and,

instructions for classifying the preprocessed vectors by (i) grouping the preprocessed vectors belonging to any of the C subclasses of the first class into a first set of vectors, and (ii) grouping the preprocessed vectors belonging to the second class into a second set of vectors.

39. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for preprocessing each of the plurality of pixel patches using a Fisher Linear Discriminant, wherein the preprocessing reduces the dimensionality of each of the pixel patches to M dimensions, wherein M is less than or equal to C; and,

instructions for classifying the preprocessed pixel patches by (i) grouping the preprocessed pixel patches belonging to any of the C subclasses of the first class into a

first set of pixel patches, and (ii) grouping the preprocessed pixel patches belonging to the second class into a second set of pixel patches.

40. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of vectors each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an Ndimensional space;

instructions for training a support vector machine with a compensatory iterative sample selection technique; and,

instructions for processing the plurality of vectors with the support vector machine, so as to classify each of the plurality of vectors into either the first class or the second class.

41. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for training a support vector machine with a compensatory iterative sample selection technique; and,

instructions for processing the plurality of pixel patches with the support vector machine, so as to classify each of the plurality of pixel patches into either the first class or the second class.

42. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

instructions for processing the plurality of pixel patches with the ensemble of binary classifiers, so as to classify each of the plurality of pixel patches into one of the two or more classes.

43. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of vectors, each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique; and,

instructions for processing the plurality of vectors with the ensemble of binary classifiers, so as to classify each of the plurality of vectors into one of the three or more classes.

44. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique; and,

instructions for processing the plurality of pixel patches with the ensemble of

binary classifiers, so as to classify each of the plurality of pixel patches into one of the three or more classes.

45. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of vectors, each being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers with a compensatory iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique;

instructions for calculating for each object, a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

instructions for generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map;

instructions for comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

46. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers with a compensatory

iterative sample selection technique, using training sets generated with an Error Correcting Output Coding technique;

instructions for calculating for each object, a probability that the pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

instructions for generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map;

instructions for comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

47. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of vectors, being mapped to one of the one or more objects, wherein each of the plurality of vectors is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

instructions for calculating for each object, a probability that the associated vector belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

instructions for generating a confidence map for each object type using the probability calculated for the vector as a confidence value within the confidence map;

instructions for comparing peaks in the confidence map for the object type with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

48. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for deriving a plurality of pixel patches from the digital image, each of the plurality of pixel patches being mapped to one of the one or more objects, wherein each of the plurality of pixel patches is an element of an N-dimensional space;

instructions for training an ensemble of binary classifiers using training sets generated with an Error Correcting Output Coding technique; and,

instructions for calculating, for each cell, a probability that the pixel patch belongs to a particular one of the three or more classes, using the Error Correcting Output Coding probability estimation technique;

instructions for generating a confidence map for each class using the probability calculated for the pixel patch as a confidence value within the confidence map;

instructions for comparing peaks in the confidence map for the class with corresponding peaks in confidence maps for other classes, and using a highest peak to assign class membership; and,

instructions for determining localization of the object corresponding to the highest peak by determining pixel coordinates of the highest peak.

49. A computer readable medium including stored instructions adapted for execution on a processor, comprising:

instructions for applying one or more fluorescent markers to a sample containing objects to be classified;

instructions for generating one or more fluorescence images of the sample containing objects to be classified;

generating a transmitted light illumination image of the sample containing objects

to be classified;

instructions for superimposing, for each of the one or more fluorescence images, at least a portion of the fluorescence image with a corresponding portion of the transmitted light illumination image; and,

instructions for using information from the one or more fluorescence images to identify characteristics of corresponding objects in the transmitted light illumination image, thereby producing a transmitted light illumination image having one or more preclassified objects.

50. The computer readable medium of claim 49, further including:

instructions for using information from the transmitted light illumination image having one or more pre-classified objects to identify characteristics of corresponding elements in one or more subsequently generated fluorescent images. 1/19

Microscopic images



FIG. 1a



FIG. 1b





I










FIG. 3c







FIG. 4

















FIG. 5d





9/19

FIG. 6a









FIG. 7c



FIG. 7d





FIG. 7b

WO 2006/055413







FIG. 8b



FIG. 9b

FIG. 9a









FIG. 10b



FIG. 11a



FIG. 11b





FIG. 11c





FIG. 11e



FIG. 12a



FIG. 12b





FIG. 13b



FIG. 13c





FIG. 14a



FIG. 14b



FIG. 15

18

16

14

12

10

8

6

4

2

0

PERCENTAGE (%)

100-SE, by PCA 100-PPV, by PCA 100-SE, by FLD

100-PPV, by FLD



FIG. 16