# Automatic Recognition of Biological Particles in Microscopic Images

M. Ranzato [a,*] P.E. Taylor [b] J.M. House [b] R.C. Flagan [b]
Y. LeCun [a] P. Perona [b]

[a] *The Courant Institute - New York University, 719 Broadway 12th fl., New York, NY 10003 USA*

[b] *California Institute of Technology 136-93, Pasadena, CA 91125 USA*

**Abstract**

A simple and general-purpose system to recognize biological particles is presented. It is composed of four stages. First (if necessary) promising locations in the image are detected and small regions containing interesting samples are extracted using a feature finder. Second, differential invariants of the brightness are computed at multiple scales of resolution. Third, after point-wise nonlinear mappings to a higher dimensional feature space, this information is averaged over the whole region thus producing a vector of features for each sample that is invariant with respect to rotation and translation. Fourth, each sample is classified using a classifier obtained from a mixture-of-Gaussians generative model.

This system was developed to classify 12 categories of particles found in human urine; it achieves a 93.2% correct classification rate in this application. It was subsequently trained and tested on a challenging set of images of airborne pollen grains where it achieved an 83% correct classification rate for the 3 categories found during one month of observation. Pollen classification is challenging even for human experts and this performance is considered good.

*Key words:* biological particles, feature, non-linearity, recognition, cells, pollen, mixture of Gaussians (MoG)

## 1 Introduction

Microscopic image analysis is used in many fields of technology and physics. In a typical application the input image may have a resolution of 1024x1024 pixels, while objects of interest have a resolution of just 50x50 pixels. In all these applications, detection and classification are difficult because of poor resolution and strong variability of objects of interest, and because the background can be very noisy and highly variable. In this work, we aim to develop a general recognition system that is able to deal with the variability *between* the classes and also with the variability *within* each class. Moreover, our approach is segmentation free and allows easy addition of new classes.

Sec. 3 describes the two datasets used to train and test the system. The first is a collection of image patches centered around corpuscles found in microscopic urinalysis. These particles can be classified into 12 categories. The second dataset is a collection of airborne pollen images. In this dataset a recognition system has first to detect pollen grains, then, to classify them into their correct genus. We have 27 different pollen categories to detect, only the 8 most numerous classes were then used to test the classifier. We considered this second dataset not only because automatic pollen recognition is still an unsolved problem and interesting application, but also because we wanted to assess the ability of the classifier to adapt to other classes of particles. In sec. 4, we present a detector that is able to extract points of interest by looking at the differences of Gaussian images, and then to generate patches which might contain a particle of interest. In sec. 5, we describe the key point of the system: a new set of very informative and straightforward features based on the local invariants defined by Mohr and Schmid (1997) [4]. A simple processing step then allows us to extract from the input image most of the information contained in the particle in the foreground, so that no segmentation is needed. Sec. 6 gives an overview at the mixture of Gaussians classifier we have used in our experiments.

In both datasets, we have to classify images with poor resolution; in addition, we are considering many different classes of biological particles that sometimes look very similar to each other. In addition to improving and automating the microscopic analysis of urine particles and the recognition of airborne pollen grains, the system we have developed can be applied to many other kinds of biological particles found in microscopic analysis. Indeed, this system has shown very good performance on a large variety of corpuscles as reported by several experiments in sec. 7, and also by other recent experiments done on

* Corresponding author. Tel.: +1 212 998 3389; fax: +1 212 995 4122.
*Email addresses:* ranzato@cs.nyu.edu (M. Ranzato), taylor@caltech.edu (P.E. Taylor), jhouse@caltech.edu (J.M. House), flagan@caltech.edu (R.C. Flagan), yann@cs.nyu.edu (Y. LeCun), perona@caltech.edu (P. Perona).

another set of images: a collection of small underwater animals [1].

## 2 Previous Work

The approach formulated by Dahmen et al. (2000) [5] for the classification of red blood cells is general and segmentation-free, as is the system we are describing in this paper. They aim to classify three different kinds of red blood cells (namely, *stomatocyte, echinocyte, discocyte*) in a dataset of gray-scale images with resolution 128x128 pixels. Their idea of extracting features that are invariant with respect to shift, rotation and scale is powerful because it allows us to cluster cells that are found in different positions, orientations and scales. Given an image, they extract Fourier Mellin based features [7] and then model the distribution of the observed training data using Gaussian mixture densities. Using these models they are able to achieve a test error rate of about 15%.

Several research works about pollen recognition were done using scanning electron microscope (SEM) and some good results have been achieved (Vezey and Skvarla, 1990 [16]; Langford et al., (1990) [9]), but SEM analysis is expensive, slow and absolutely not suitable for a real-time application.
France et al. (1997) [6] describe two methods of pollen identification in images taken by optical microscopy. This work is one of the first attempts to automate the process of pollen identification using the optical microscope. Using a collection of about 900 patches containing pollen grains and 900 patches containing particles that are not pollen, they aim to identify the pollen grains. Their best approach uses the so-called "Paradise Network". Several small templates able to identify features in the object are generated, then linked together to form a certain number of patterns. This composition is repeated in testing, and the system looks for the best match between the assembled pattern and the pollen and non-pollen patterns. The network is able to recognize nearly 79.1% of pollen as pollen, 4% of the debris is also classified as pollen.

Ronneberger et al. (2000) [14] acquire 3D volume data for each pollen grain with a fluorescence microscope. They have a dataset of 350 samples of pollen taken directly from trees (i.e. samples with very little variation in appearance), and belonging to 26 categories. They extract features that are 3D invariant and use a Support Vector Machine (SVM) to classify them. Testing one image at a time they achieve an error rate of about 8%. Note that this dataset is too small to get a reliable result, that they are using "pure" samples, and that 3D

[1] Edgington, D.R., Kerkez, I., Cline, D.E., Ranzato, M., Perona, P. 2006. "Detecting, tracking and classifying animals in underwater video". IEEE International Conference on Computer Vision and Pattern Recognition, demonstration, New York

analysis is not suitable for a real-time application.

Luo et al. (2004) [12] have developed a plankton recognition system using a dataset of images which has many similarities to ours. Their particles do not have clear contours and might be partially occluded and there are also many unidentifiable objects that must be discarded. First, they extract features by considering invariant moments, granulometric measurements and domain specific features. Secondly, they apply a feature selection procedure that reduces the number of features and happens to keep almost all the domain specific features. Finally, they classify with an SVM to obtain an overall average accuracy of 76%, but with a high error rate on the unidentifiable particles (about 42%).

Uebele et al. (1995) [15] use a neural-network-based fuzzy classifier on a dataset of blood cells. After extracting 13 features from each sample, they train a three-layered neural network. Then, they used the trained weights to cluster the feature space into non-overlapping regions, which reduces the error rate by 1%. More recently, a multi-layer convolutional neural network (LeCun et al., 1998 [10]) has been trained successfully to detect faces and to recognize objects in natural and biological images (Ning et al., 2005 [13]). This technique is very attractive because it employs *end-to-end* learning which makes unnecessary any hand-made tuning and any feature selection activity, but it requires a very large dataset to estimate its internal parameters.

## 3 Datasets

Our first dataset is a collection of particles found in microscopic urinalysis. Because of the technique used to acquire images in urinalysis, and the low concentration of particles in the specimen, the only challenging task in this application is classification. Thus the images in this dataset are already small image patches centered around a corpuscle. Some of these samples are shown in fig. 1. They are (columns from left to right): bacteria, white blood cell clumps, yeasts, crystals, hyaline casts, pathological casts, non squamous epithelial cells, red blood cells, sperm, squamous epithelial cells, white blood cells and mucus. All categories have 1000 gray-level images with the exception of mucus (546 samples) and pathological casts (860 samples). The size is variable, and goes from 36x36 to 250x250 pixels at $0.68 \mu m/pix$. As shown in fig. 1, there can be a marked variability in shape and texture among items belonging to the same category, but also a similarity among cells belonging to different classes.
During the tuning stage we took 470 images per class to estimate the parameters of the system and validated on 30 images. The final test was performed taking 90% of the images in each class as training images and the remaining

ones as test images with random extraction. In each of these training sets, we included the images used in the tuning stage.

We developed our classifier using the urine dataset and then tested our system on a completely different dataset of particles: a collection of airborne pollen images. An example of 4 images belonging to this dataset is shown in fig. 2. This dataset has 1429 images containing 3686 pollen grains belonging to 27 species. Images have a resolution of 1024x1280 pixels with $0.5\mu m/pix$. As you can see in fig. 2, there can be images with a high density of particles (upper left corner), with a low density of particles (upper right corner), with big objects inside (lower left corner) and with bubbles of glue (lower right corner). These images are acquired using an optical microscope to analyze a tape collecting samples from the most standard and widely used air sampler [2]. In order to build a labeled dataset, we developed a GUI to enable human operators to manually collect a large training set. Using this software, we built a reference list which stores for each image like those in fig. 2 the position and the genus of each pollen particle identified by some expert. Given this information, we derived a dataset of patches centered on pollen particles. Fig. 3 shows some samples belonging to the most numerous pollen classes. They are (columns from left to right): alder (416 samples), ash (329 samples), elm (522 samples), mulberry (111 samples), oak (413 samples), olive (326 samples), jacaranda (282 samples) and pine (705 samples). Only these 8 most numerous categories will be considered in the classification experiments reported in sec. 7.2. On the pollen dataset we have tested the whole recognition system, detector and classifier. Detection on this dataset (see fig. 2) is very challenging because pollen grains are rare in images and the background is highly variable. A segmentation-based approach has never given good results. In addition, classification is difficult because of the natural variability of pollen grains belonging to the same category and the smooth transition in appearance across items belonging to different classes (see fig. 3).
Because of the shortage of available samples in this dataset of patches, we did not consider a validation set. In the classification experiments described in sec. 7.2, tuning is done by randomly taking 10% of the images for testing. The final test is an average of 100 experiments in each of which 10% of the images are randomly extracted to build the test dataset.

## 4 Detection

Detection is the first stage of a recognition system. Given an input image with a lot of particles, it aims to find key points which have at least a little probability of being a particle of interest. Once one of these points is detected,

[2] This air sampler is called a Burkard 7-day volumetric spore trap.

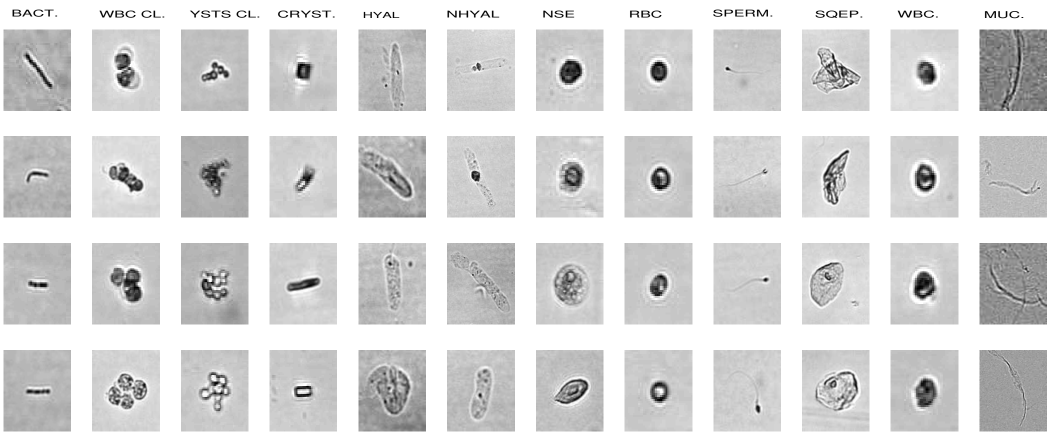


Fig. 1. Urine dataset: samples belonging to the twelve categories. The resolution is variable (from 36x36 up to 250x250) but in this figure, all the images have been rescaled to fit the grid.

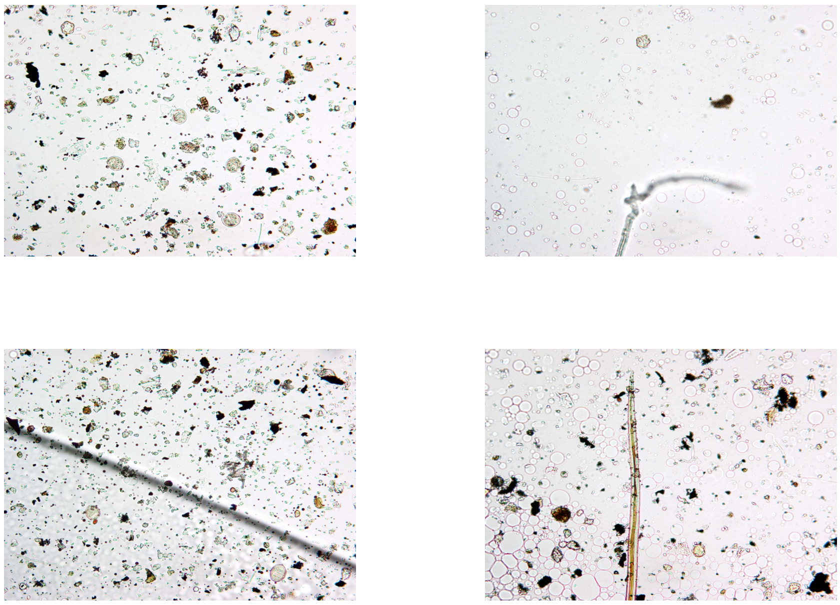

Fig. 2. Examples of images in the pollen dataset; this dataset has 1429 images with 3686 pollen grains belonging to 27 species. In a typical image, only two or three pollen grains can be found while all the other particles are dust. The resolution is 1024x1280 pixels.

a patch is centered on it, and then it is passed to the classifier in the next stage. It is extremely important to detect all the objects of interest especially when they are rarely found in images and their number is low, as typically happens in pollen detection. Though we only need to detect corpuscles in images of the pollen dataset, these are so variable in terms of particle density and shape that a general-purpose system has to be developed.

Our detector is based on a filtering approach (Lowe 1999 [11]). The input image is converted in gray level values. It is blurred and sampled at two

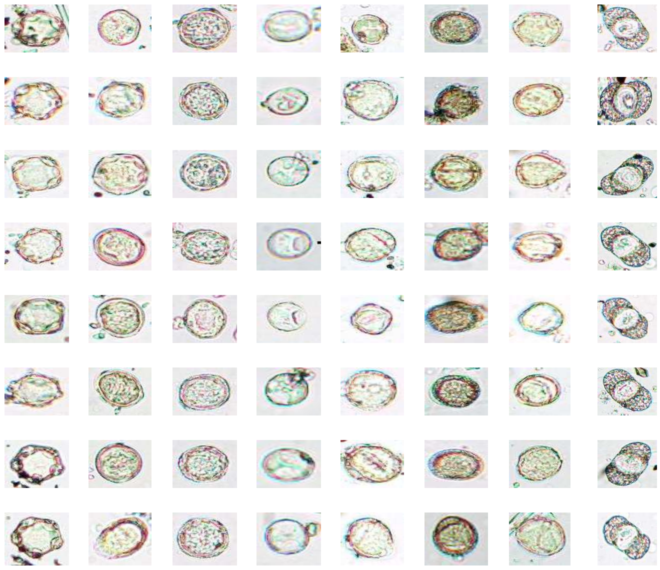

Fig. 3. Examples of pollen patches extracted by some expert from images like those of fig. 2. Each column shows examples belonging to the same category, (columns from left to right): alder, ash, elm, mulberry, oak, olive, jacaranda and pine. These samples are taken from the 8 most numerous categories and have an average resolution of 50x50 pixels; these images are rescaled to fit the grid.

different frequencies, $F_1$ and $F_2$. In order to avoid artifacts near the image border, we normalize the background brightness to zero [3]. We apply to both sampled images a filtering stage, and for both images we compute a difference of Gaussians (Lowe 1999 [11]). A set of interest points is extracted looking at the extrema of the two previously found images. In this way, we focus on particles that are approximately round and within a certain size.
Next, we remove points that are too close to the image border, and for the remaining ones we create a proper bounding box (see fig. 4). In order to do this, we estimate 8 points on the edge of the particle near the interest point; then, we take that circle passing through 3 of these points and minimizing a fitting error. This is the sum of the distances of the 3 closest points (among the remaining 5) to the circumference. This circle yields a square bounding box whose side and centroid are the diameter and center of the circumference, respectively. The fitting error in this circumference estimation is then used to choose which patch to discard in the case of two partially overlapping bounding boxes. After removing those patches that are too small or that overlap too much with other patches, and those patches whose average brightness is too high (because of bubbles of glue, see bottom right picture in fig. 2), we have

[3] The estimate of the background brightness is given by the peak of the histogram of all the image pixel values.

a set of patches that might contain a particle of interest.

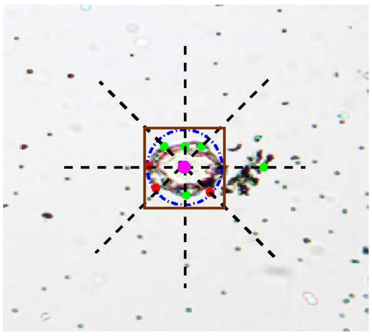

Fig. 4. During detection (1) an interest point is found inside each particle (magenta dot), (2) 8 points are estimated on the edge of the particle (red and green dots), (3) a circle passing through 3 of these points (red dots) and minimizing a fitting error is computed, and (4) a bounding box is determined by taking diameter and center of the circumference as side and centroid of the box.

## 5 Feature extraction

In order to recognize an object in an image, it must be represented by some kind of features that should express the characteristics and the information contained in the image. These features should be invariant with respect to shift and rotation of the particle, and should be robust against small changes in scale as well. In this way, we are able to cluster images belonging to a certain class because we acquire the same features independent of the position and orientation of the cell. We derived our features from the "local jets" originally studied by Koenderink and Van Doorn (1987) [8], and the invariants as defined by Mohr and Schmid (1997) [4].

Let $I(x, y)$ be an image with spatial coordinates $x$ and $y$. Given $I(x, y)$ we can compute a set of local invariant descriptors with respect to shift and rotation by applying kernels $K_j$ as described by Koenderink and Van Doorn (1987) [8]. For example, the first invariant is the local average luminance, the second one is the square of the gradient magnitude, and the fourth one is the Laplacian. In this work, the set of invariant features we consider is limited to the first 9, which are combinations up to third order derivatives of the image. Moreover, these 9 invariants are computed at different scales. At each stage a blurred image with a Gaussian kernel $G_{\sigma i}$ is considered; the width of the kernel is $\sigma_i^2 \in \{0, 1/2, 1, 2\}$.
Each pixel in the image has an associated vector with $9 \times 4 = 36$ components that are given by

$$F_{i,j}(x, y) = I * G_{\sigma_i} * K_j(x, y), \text{ with } i \in \{0, 1, 2, 3\}, j \in \{0 \cdots 8\} \tag{1}$$

Let $m_{bg}$ be the mean of $F_{i,j}(x,y)$ over all the $(x,y)$ that fall in the 10 pixel-wide frame around the image-patch border (i.e. we are estimating the background brightness of component $i,j$). We define the non-linear functions $f_k$ as follows:

$$\begin{aligned}
f_1(z) &= \begin{cases} 0 & \text{if } z < m_{bg} \\ z - m_{bg} & \text{if } z \geq m_{bg} \end{cases} && \textit{positive part} \\
f_2(z) &= \begin{cases} z - m_{bg} & \text{if } z < m_{bg} \\ 0 & \text{if } z \geq m_{bg} \end{cases} && \textit{negative part} \\
f_3(z) &= |z - m_{bg}| + m_{bg} && \textit{absolute value}
\end{aligned} \tag{2}$$

Thus, we define a new pixel feature as:

$$H_{i,j,k}(x,y) = f_k(I * G_{\sigma_i} * K_j(x,y)) \tag{3}$$

with $i \in \{0,1,2,3\}$, $j \in \{0 \cdots 8\}$ and $k \in \{1,2,3\}$. Each pixel has a vector with $9 \times 3 \times 4 = 108$ components.
Finally, for each $(i,j,k)$ in $H_{i,j,k}$ we thus average all the corresponding non-zero pixel values in the image, and we construct a single image feature vector with 108 components. This feature vector is invariant to shift and rotation of the cell, is scale-invariant because of the average operation, and is robust with respect to changes in the size of the patch centered around the particle because we exclude from the mean those values that are at zero. Most importantly, it is computed without any segmentation of the cell. To summarize, the detection algorithm described in sec. 4 extracts a square bounding box around a particle, and then an invariant feature vector is computed without segmenting the particle from the background of the patch.

We apply the non-linearity in eq. (2) (represented on the right panel of fig. 5) because we have found experimentally that this produces a dramatic decrease in the error rate when compared with a system using just an average of the features defined in eq. (1); the error rate is almost halved. Other non-linearities have been tested (see fig. 5), but the non-linearity we have introduced in eq. (2) achieves the best performance while keeping the system very simple.

## 6 Classification

Given a patch with one centered object, a feature vector is extracted from the image. In training, the system learns how to distinguish among features of images belonging to different categories. In the test phase, a decision is made

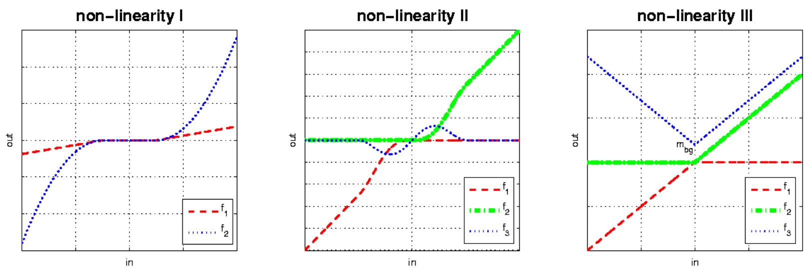


Fig. 5. Non-linearities tested: non-linearity I performs a linear and quadratic mapping resetting to zero those values around the background mean; non-linearity II aims to separate the low, medium and high ranges of values; non-linearity III is a simplification of non-linearity II. The second and the third non-linearities give the best results; we have chosen the last one.

according to the test image feature. If the object in the image is considered a particle of interest, then its class is also identified; otherwise, it is discarded because the feature does not match any training model.

We use a Bayesian classifier that models the training data as a mixture of Gaussians (MoG) (Bishop 2000 [3]). If the number of categories is $K$, then we have to estimate $K$ MoGs. At the end of training, we get for each category the *class-conditional distribution* $f(\boldsymbol{x} \mid C_j)$, for $j = 1...K$. The estimation is done by applying the expectation maximization (EM) algorithm (Bishop 2000 [3]). An important property of mixture models is that they can approximate any continuous probability density function to arbitrary accuracy with a proper choice of number of components and parameters of the model.

Given a test image, we compute its feature vector $\boldsymbol{x}$. Then, we apply Bayes' rule: we want to pick the class that maximizes the probability $P[C_k \mid \boldsymbol{x}]$ of the class $C_k$ given the test data $\boldsymbol{x}$. Applying the extension of Bayes' rule to continuous conditions and assuming equal distribution for the class probabilities, we need only to find the class that maximizes $f(\boldsymbol{x} \mid C_k)$, where $f$ is the probability density function learned during the training stage.

Note that we need to decrease the feature space dimensionality before estimating the MoGs, since the number of parameters in the model grows very quickly with the dimension of the feature space and we have available only a limited number of training samples. Particularly, given the whole set of training-feature vectors we compute Fisher's Linear Discriminants (Bishop 2000 [3]). We project both the training and the test data along these directions in order to decrease the dimensionality of the feature space from 108 to a certain $D$. This parameter is found by a cross-validation procedure during the training phase.

Note that we are presenting the simplest classifier that is able to give a good performance. Other classifiers have been tried. Slightly better performance can be attained using this classifier as basic weak learner in the AdaBoost algorithm and by classifying with SVM. However, the improvement amounts to only a fraction of percent in the error rate, while the simple Bayesian classifier allows easy addition of new categories by estimating a MoG for the new class independently from the others. Thus, we will refer only to the mixture of Gaussians classifier in the next section.

## 7 Experiments

### *7.1 Detection*

Detection is performed only on images of the pollen dataset (see fig. 2). We evaluate the performance by measuring how well our automatic detector agrees with the set of reference labels. A *detection* occurs if the algorithm indicates the presence of an object at a location where a pollen exists according to the reference list. Similarly, a *false alarm* occurs if the algorithm indicates the presence of an object at a location where no pollen exists according to the reference list. We say that there is a match between the detector box and the expert's one, if

$$\sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \leq \alpha \sqrt{\frac{A_1 + A_2}{2}} \tag{4}$$

where $x_{ci}, y_{ci}$ are the coordinates of the box centroid, $A_i$ is the area of the box and $\alpha$ is a parameter to be chosen. In our experiments, we have chosen $\alpha$ equal to $1/\sqrt{2}$. The overall performance can be evaluated computing the *recall* and the *precision*. The former is the percentage of detected pollen grains when compared to the pollen particles labeled by the experts. The latter is the ratio of detected pollen grains over the total number of detected particles. After tuning the system in order to find good values for the sub-sampling frequencies (8 and $8\sqrt{2}$) and for the variance of the Gaussian kernel ($\sqrt{2}$), we ran the detector on all the images of the dataset. We achieved a recall equal to 93.9% and a precision equal to 8.6%, that is, we are able to find a pollen with a probability of 0.93 and for each pollen grain another 10 particles that are not pollen are detected. This is a good performance because we are dealing with a highly variable background and with many other particles in each image. Even an expert has to focus his attention on about 10 points before finding a pollen grain in an image. Such a high number of false alarms is not a problem as long as we are able to make a good classification of these particles (see next section).

Since we aim to assess the performance of the whole pollen recognition system, that is, detection followed by classification, we have built a dataset with these automatically detected patches in order to train and test the classifier. For this reason, we have gone through this dataset to remove those patches having an incorrect bounding box. It is possible to have a good overlap with the expert's patch even if the particle is not properly centered in the region of interest because of dust or other particles around it. Fig. 6 shows examples of "good" and "bad" patches belonging to the *alder* category. If we consider only the "good" patches, the overall percentage of detection decreases to 83%, but then we are able to perform a meaningful classification on those extracted images.
Note that this technique is quite fast: the analysis of one image of size 1024x1280 pixels takes 6 seconds on a 2.5GHz Pentium processor computer (code in Matlab).

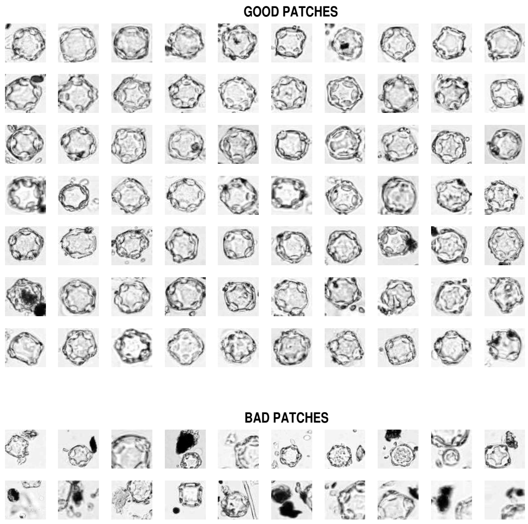


Fig. 6. Examples of 90 patches of alder pollen extracted by our detector. The first 7 rows show examples of good bounding boxes. The last 2 rows show samples that were removed from the set because they were incorrectly centered in their bounding box. Problems usually arise when there is dust attached to the pollen or when pollen grains are grouped in a single cluster. 93% of pollen grains are detected, and 83% of pollen particles are detected and correctly assigned to a region of interest. Note that images have been rescaled to a common resolution in order to fit the grid.

### *7.2 Classification*

The classifier needs to be tuned with respect to the number of feature-space dimensions, the number of components in the mixture and the structure of the covariance matrix of the Gaussians in the mixture. Thus, we ran many experiments varying one parameter at a time, and kept that combination of values that gave the lowest error rate. In order to estimate parameters of the

MoGs by EM algorithm, we used some functions of the NETLAB package for MATLAB [1]. Since the EM algorithm needs a random initialization, we also ran many experiments varying only the state of the random number generator in order to find the best values.

We first considered the urine dataset and took 500 images per class for the tuning stage. The best results were achieved using 4 full covariance Gaussians in a 12 dimensional feature space in each mixture. The test was performed taking 90% of the images in each class as training images and the remaining ones as test images with random extraction. The average results of 10 experiments is shown in fig. 7 in the form of a confusion matrix; the error rate on these 12 categories is equal to 6.8%.

In order to assess whether the system is able to classify particles with a different statistics, we applied the classifier to the dataset of pollen grain patches. Let us consider first the dataset of pollen patches extracted by experts (see fig. 3). Tuning is done by randomly taking 10% of the images for testing. The result of this stage is shown in tab. 1. To assess the classification performance, in fig. 8 we show the average confusion matrix of 100 experiments; the error rate is 21.8%. Each time we have randomly extracted from each class 10% of the images for test purposes. When the same experiment was done using the patches containing pollen grains that were found by the detection system we achieved an error rate of 23.2%.

In order to assess the overall performance of the pollen recognition system, we will need to combine the detector and the classifier. Since we train the classifier separately from the detector, however, we feed the classifier not only with the 8 most numerous pollen categories, but also with the same 10:1 ratio of false alarms to genuine particles that the detector will provide. We have found experimentally that it is better to model the false alarm class and, given its huge number of available samples, we can use a more complex model for it. Moreover, we put a threshold on the estimated probability of the class given the input data. If there is low confidence, we do not make a decision and instead assign the sample to an "unknown" category, *UNK* in the graphical representation of the confusion matrices shown in fig. 9 and fig. 10. Of course, this unknown class can be merged with the false alarm class; for this reason, we assume the system is correct when it assigns a false alarm patch either to the false alarm category or to the unknown class (in the following experiments the error rate is computed differently in order to deal with these considerations). The tuning stage proceeds as in the previous experiments and it is summarized in tab. 1. The average performance on 100 experiments is shown in fig. 9.

Note that in any practical application of airborne pollen recognition, it is useful to know what types occur together at various times of the year in the

geographic area of interest. For example, since we know that only ash, alder and pine pollen grains are commonly found in January in this area [2], we can restrict the pollen categories to these three. Taking 10 false alarm patches for each of pollen grain, and proceeding with an analogous tuning (see tab. 1), we have achieved an error rate of about 17% as shown in fig. 10.
Note that 20ms are required to analyze a particle on a 2.5GHz Pentium processor computer (code in Matlab).

We conclude this section reporting classification results on the urine dataset using some methods referenced in sec. 2. The same mixture of Gaussians classifier has been trained on a different set of features: the Fourier-Mellin based features [7] used by Dahmen et al. (2000) [5]. These features are the amplitude spectrum of the image Fourier transform computed in a log-polar coordinate system. With these features we have achieved an error rate of about 21% for the classification of the urine particles into the 12 categories.
Finally, we report a result on the same dataset using a different classifier. We trained different architectures of a convolutional neural network on raw images [10]. We reduced the images to a common resolution of 52x52 pixels by bilinear interpolation or sampling since the original resolution is variable. The network giving the best performance has in the first convolutional layer 6 feature maps. This is followed by a sub-sampling layer with a vertical and horizontal stride equal to 2. The next convolutional layer outputs 16 feature maps which are sub-sampled by a factor of 4 in the next sampling layer. The last convolutional layer has 120 feature maps which will be reduced to 12 (as many as the number of classes) by a 1 layer neural network. With this system we have achieved an error rate of 16%. The network seems to overfit the data and needs more training samples overall for the classes characterized by a very low contrast; few thousands of samples seem to be insufficient to properly estimate the more than 50,000 internal parameters.

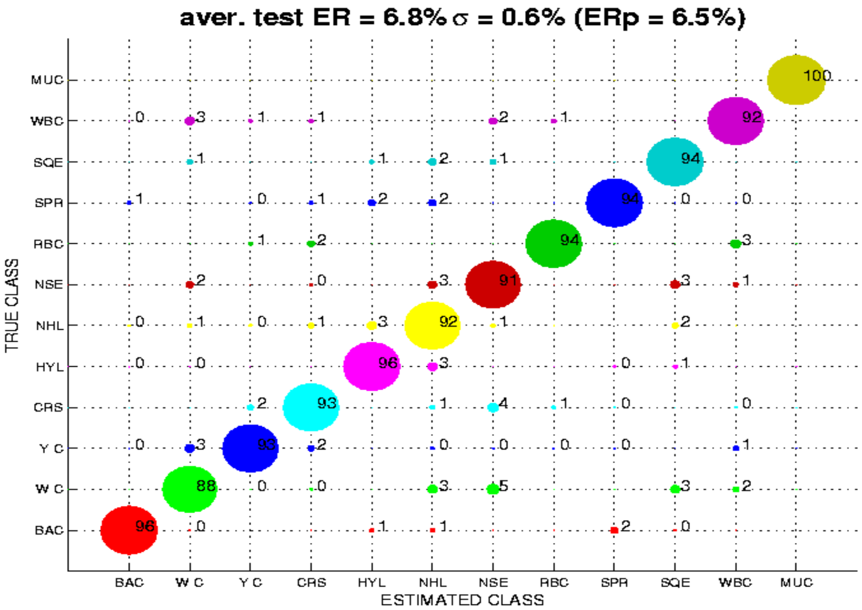


Fig. 7. Classification of urine particles: averaged confusion matrix for experiments in which 90% of images in the full data set are used for training and 10% are randomly extracted for testing. The proportional error rate (*ERp* in parenthesis) is the average of the error rates of each single class; the main error rate shown is the ratio between the total number of misclassified images and the total number of tested images. We refer also to the proportional error rate because we run experiments using a different number of test samples for each class, namely, classes *NHL* and *MUC* have fewer test images. In the intersection of row $i$ with column $j$, we have the percentage of items belonging to class $i$ that have been assigned to class $j$. On the diagonal, we show the percentage of correct classification of each class.

## 8 Summary and Conclusions

Microscopic analysis of biological particles aims to find and to classify particles. These objects often have variable shape and texture, and very low contrast and resolution. Moreover, the background can be highly variable as well. Recognition on these images is very challenging.

In this work, we have defined a new kind of feature based on "local jets" (Mohr and Schmid 1997 [4]). These are able to extract information from a patch centered on the object of interest without any segmentation. Classification performed with a simple mixture of Gaussians classifier has achieved a very low error rate in considering the 12 classes of cells that are found in urinalysis. Even though these images are characterized by very low contrast and poor resolution, we reached an error rate equal to 6.8%. The same classifier is able to achieve an error rate of 22% on 8 species of airborne pollen grains. This classifier outperforms the previous systems dealing with analogous pollen images and makes feasible the development of a real-time instrument

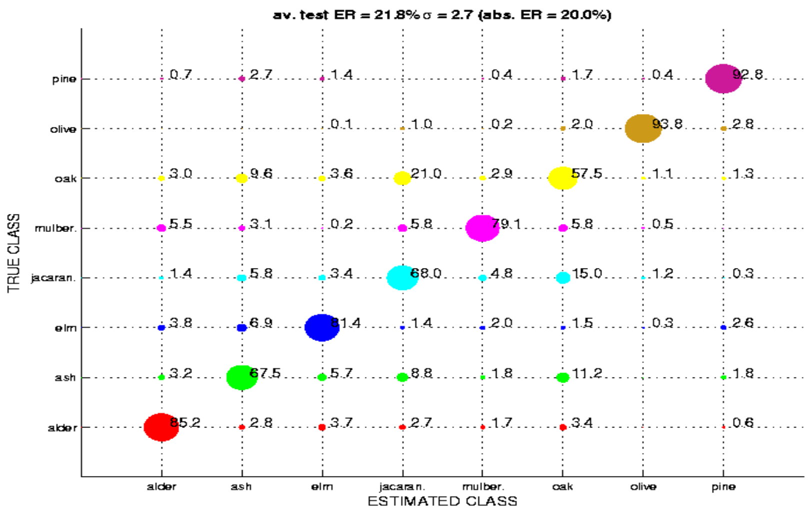


Fig. 8. Classification of 8 pollen categories (expert's patches): average test confusion matrix of 100 experiments. Every time 10% of the images are randomly extracted for testing. The proportional error rate (average of the error rates of each single class) is first shown; the error rate in parentheses is the ratio between the total number of misclassified images and the total number of tested images. Since we are testing a different number of samples for each class, we mainly refer to the proportional error rate.

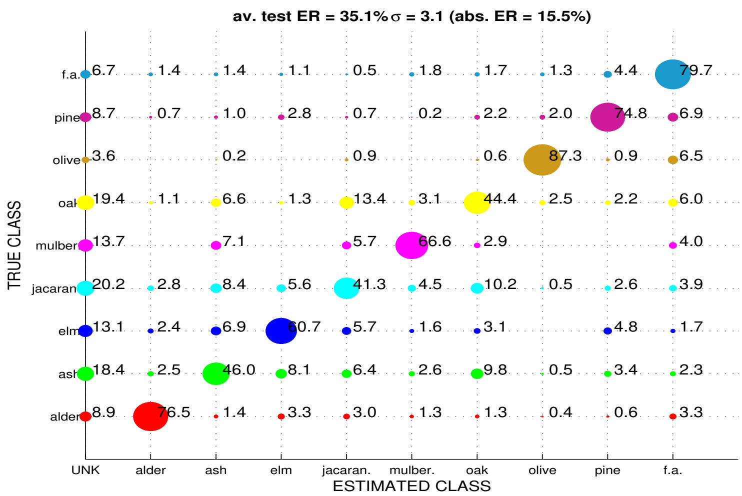


Fig. 9. Classification of 8 pollen species and false alarm class: average test confusion matrix of 100 experiments in which the detector patches were used. In these experiments, we have taken for each pollen particle 10 false alarms and we have randomly extracted 10% of the images for testing. In the plot *f.a.* collects the patches assigned to the false alarm class, *UNK* the patches having classification confidence below threshold. A false alarm patch is correctly classified when it is assigned either to *f.a.* or to *UNK*.

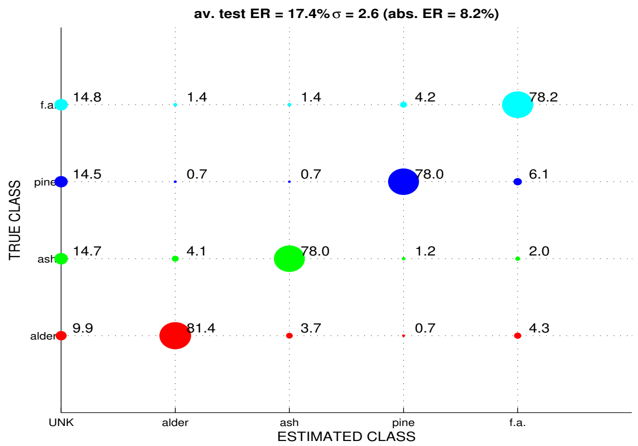


Fig. 10. Classification of those pollen grains that can be found in January: average test confusion matrix of 100 experiments in which the detector patches were used. Even if for each pollen grain we also detect 10 false alarms, most of these particles are subsequently rejected by the classifier: 14.8+78.2=93% of these objects are correctly classified.

for counting pollen. In this work, we did not aim to address a specific application and we have proved by many experiments on different datasets that this system is actually able to achieve very good performance on many different kinds of "small" object categories found in microscopic images.
We believe that our features could be improved by considering color information and by applying more complex non-linearities.
We have also presented a detection system based on a filtering approach. This gives very good results on a pollen dataset because it is able to detect a pollen grain with a probability of 9/10. For each pollen grain another 10 particles that are not pollen are detected, though most of these extraneous corpuscles are subsequently rejected by the classifier. Moreover, even an expert has to focus his attention on nearly 10 points in each image to establish the presence of one grain of pollen. This result is reliable because we have used 1429 images with 3686 pollen grains. Improvements could be obtained by introducing other kinds of tests to decrease the number of false alarms, by considering a more sophisticated estimation of regions of interest, and by taking into account color information as well.

## Acknowledgements


The authors are very grateful to C. Poultney and H.M. Kim for their precious comments.
Research described in this article was supported by Philip Morris USA Inc., Philip Morris International and the Southern California Environmental Health

| **parameters** | **value** |
| --- | --- |
| *8 pollen classes: expert's patches* fig. 8 | |
| dimension D | 14 |
| number components in MoG | 2 |
| covariance in MoG | diagonal |
| *8 pollen classes and false alarms detector patches* fig. 9 | |
| dimension D | 14 |
| number components in MoG (pollen) | 2 |
| covariance in MoG (pollen) | spherical |
| number components in MoG (f.a.) | 4 |
| covariance in MoG (f.a.) | full |
| confidence threshold | 0.5 |
| *3 pollen classes and false alarms (January) detector patches* fig.10 | |
| dimension D | 8 |
| number components in MoG (pollen) | 1 |
| covariance in MoG (pollen) | full |
| number components in MoG (f.a.) | 10 |
| covariance in MoG (f.a.) | full |
| confidence threshold | 0.7 |

Table 1
Pollen classification: parameters found during the tuning stage.

Sciences Center (NIEHS grant number 5P30 ES07048). P. Taylor gratefully acknowledges support from a Boswell Fellowship, which is a joint position intended to foster cooperation between the California Institute of Technology and the Huntington Medical Research Institute.

## References

[1] http://www.ncrg.aston.ac.uk/netlab/down.php

[2] http://www.che.caltech.edu/groups/rcf/DataFrameset.html

[3] Bishop, C.M., 2000. Neural Networks for Pattern Recognition, Oxford University Press.

[4] Mohr, R., Schmid, C., 1997. Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5), 530-535.

[5] Dahmen, J., Hektor, J., Perrey, R., Ney, H., 2000. Automatic classification of red blood cells using Gaussian mixture densities. Proc. Bildverarbeitung für die Medizin 331-335.

[6] France, I., Duller, A.W.G., Lamb, H.F., Duller, G.A.T., 1997. A comparative study of approaches to automatic pollen identification. Proceedings of British Machine Vision Conference.

[7] Grace, A.E., Spann, M., A comparison between Fourier-Mellin descriptors and moment based features for invariant object recognition using neural networks. 1991 Pattern Recognition Letters, 12:635-643

[8] Koenderink, J.J., van Doorn, J., 1987. Representation of local geometry in the visual system. Biological Cybernetics 55, 367-375.

[9] Langford, M., Taylor, G.E., Flenley, J.R., 1990. Computerised identification of pollen grains by texture. Review of Paleibotany and Palynology 1990.

[10] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86 (11), 2278-2324.

[11] Lowe, D.G., 1999. Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision.

[12] Luo, T., Kramer, K., Goldgof, D.B., Hall, L.O., Samson, S., Remsen, A., Hopkins, T., 2004. Recognizing plankton images from the shadow image particle profiling evaluation recorder. IEEE Transactions on System, Man and Cybernetics 34 (4) 1753-1762.

[13] Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., Barbano, P., 2005. Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing 14 (9), 1360-1371.

[14] Ronneberger, O., Heimann, U., Schultz, E., Dietze, V., Burkhardt, H., Gehrig, R., 2000. Automated pollen recognition using gray scale invariants on 3D volume image data. Second European Symposium on Aerobiology.

[15] Uebele, V., Abe, S., Lan, M.S., 1995. A neural-network-based fuzzy classifier. IEEE Transactions on System, Man and Cybernetics 25 (2), 353-361.

[16] Vezey, E.L., Skvarla, J.J., 1990. Computerised feature analysis of exine sculpture patterns. Review of Paleibotany and Palynology.