Machine Learning approach to discriminate *Saccharomyces cerevisiae* yeast cells using sophisticated image features

Mohamed Tleis\(^1\) and Fons J. Verbeek\(^1\) *

\(^1\)Imaging and Bioinformatics, Leiden Institute of Advanced Computer Science and Mathematics (LIACS), Leiden University, The Netherlands.

http://bio-imaging.liacs.nl/

Summary

In biological research, *Saccharomyces cerevisiae* yeast cells are used to study the behaviour of proteins. This is a time consuming and not completely objective process. Hence, Image analysis platforms are developed to address these problems and to offer analysis per cell as well. The robust segmentation algorithms implemented in such platforms enables us to apply a machine learning approach on the measured cells. Such approach is based on a set of relevant individual cell features extracted from the microscope images of the yeast cells. In this paper, we composed a set of features to represent the intensity and morphology characteristics in a more sophisticated way. These features are based on first and second order histograms and wavelet-based texture measurement. To show the discrimination power of these features, we built a classification model to discriminate between different groups. The building process involved evaluation of a set of classification systems, data sampling techniques, data normalization schemes and attribute selection algorithms. The results show a significant ability to discriminate different cell strains and conditions; subsequently it reveals the benefits of the classification model based on the introduced features. This model is promising in revealing subtle patterns in future high-throughput yeast studies.

1 Introduction

In biological research, Baker’s yeast is an excellent standard model organism since many processes that take place in both animal and plant cells occur in a similar way in yeast. Baker’s yeast, scientifically known as *Saccharomyces cerevisiae*, can be easily cultured and used to study the behaviour of genes by tagging them with fluorescent proteins.

The study of proteins is time consuming and not completely objective. Hence, automated analysis platforms are developed to address this problem. Such platforms are ideally composed of segmentation, measurement and data analysis modules. The segmentation module segments the cells located in the microscope images, while the measurement module measures various features of the segmented cells. The data analysis part analyzes the measurement and report relevant statistics about the different cell groups [1].

Subtle Patterns are not easy to be extracted from the measurement or basic statistical analysis of the data especially in high throughput screening (HTS) where thousands of images are analysed.

*To whom correspondence should be addressed. Email: m.tleis@liacs.leidenuniv.nl*
Moreover, when biologists construct and study different yeast cell strains cultivated in different media, it can be not possible to know if the different cell groups have different characteristics for the same expressed proteins. Thus, the need of an automatic system in order to extract those hidden features.

The idea in this work is to apply a machine learning approach to address the automatic analysis of data. The features (attributes) for training the machine learning system were designed in a way to offer a more sophisticated description of the intensity and morphology characteristics of the cells. Such features includes basic shape descriptors, intensity descriptors, texture measurements, first and second order histogram features, i.e. moment invariants and co-occurrence matrix derived features, in addition to wavelet-based texture measurement.

After creating our dataspace with the introduced features. A number of linear and non-linear classifiers were evaluated to select the best model that can classify the instances in the dataspace into cells belonging to a different group, i.e. different strain, or cultured in a different medium. Since our dataspace is imbalanced with a different ratio for different cell groups, we were careful in our evaluation to choose the classifier system that can classify well the majority as well as the minority classes. Therefore, we considered sampling and normalization techniques. Cross validation was considered as well, and feature selection algorithms were evaluated to choose the best algorithm that fits our dataset.

The result shows that many classifiers performed excellently after data preprocessing. Consequently a classification model is built. This Model can significantly discriminate between two different cell groups.

The next section discusses the features used to describe the cells. Subsequently, there is the section about building the classification model, then the results section to illustrate the advantages of machine learning and the designed features in our study. We finalize this paper with a conclusion section and points to consider for future work.

2 Features and Texture Measurement

We have successfully developed a platform to segment *S. cerevisiae* yeast cells and measure a range of features and textures for each individual cell obtained from two channel images acquired by a laser scanning confocal microscope (CLSM). Figure 1 shows a sample two-channels image of *S. cerevisiae* yeast cells. The first channel in Fig. 1a is a bright-field channel depicting yeast cell structures. The second overlaid channel in Fig. 1b is a fluorescent channel of the BMH1 gene expressed Bmh1 protein binded with GFP protein cultivated in low NaCl medium. The yellow contours surrounding the cells in Fig. 1 are the results of our segmentation algorithm [2]. In this study, more sophisticated features and texture measurement were introduced to describe the characteristics of cell morphology and intensity distribution to facilitate the analysis and discrimination of different yeast cells. An image feature is a representation or an attribute of an image describing certain special characteristics of the pattern of interest. While feature extraction is defined as locating those pixels in an image that have some distinctive characteristics [3]. Texture is defined as the visual effect which is produced by spatial distribution of total variations over relatively small areas [4]. Image texture is believed to be a rich source
of visual information. They are complex visual patterns composed of entities, or sub-patterns, that have characteristic brightness, colour, slope, size, etc. Thus texture can be regarded as a similarity grouping in an image. The most known feature extraction techniques in image analysis are considered in our research. These techniques are classified into histogram based features and the moment invariants derived from them, co-occurrence matrix based features, and multi-scale features [5]. In the following sub-section we start discussing the histogram based features then the moment invariants derived from them. The third sub-section is dedicated to explain the additional features derived from the co-occurrence matrix; and the last sub-section highlights on the multi-scale features and specifically wavelet-based texture features.

2.1 First order histogram based features

Assuming that our microscope image is a function \( f(x, y) \) of two space variables \( x \) and \( y \), \( x = 0, 1, \ldots, N - 1 \) and \( y = 0, 1, \ldots, M - 1 \). The function \( f(x, y) \) can take discrete values \( i = 0, 1, \ldots, L - 1 \), where \( L \) is the total number of intensity levels in the image. The intensity-level histogram is a function showing the number of pixels for each intensity level in the whole image. This function is depicted in Eq. 1, where \( \delta(j, i) \) is the Kronecker delta function, depicted in Eq. 2.

\[
h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x, y), i), \tag{1}
\]

\[
\delta(j, i) = \begin{cases} 
1, & j = i \\
0, & j \neq i 
\end{cases} \tag{2}
\]

The histogram of intensity levels is obviously a concise and simple summary of the statistical information contained in the image. Calculation of the grey-level histogram involves single pixel. Thus the histogram contain the first-order statistical information about the image, i.e. the region of interest (RoI) of cell objects. Different useful image features are worked out from

Figure 1: Sample image of segmented S.cerevisiae yeast cells in two overlaid channels
the histogram to quantitatively describe the first-order statistical properties of the cells. In this paper, we considered many basic shape descriptors based on the first order histogram, and the most relevant texture features among those originally proposed by Haralick et al. [6, 7]. A list of those features are listed in Table 1.

Another way to characterize the texture is by deriving moment invariants from the first order statistical information in the histogram [8]. The following sub-section discusses these moment invariant features.

2.2 Moment invariant features

Recognition of visual patterns independent of position, size, and orientation in the visual field has been a goal of much recent research. To achieve maximum utility and flexibility, it would be useful if the extraction technique is insensitive to variations in shape and provide improved performance with repeated trials. This property is known in moment invariant techniques. An image moment is defined as a certain particular weighted average, i.e. moment, of the pixel intensities in an image. Traditionally, moment invariants are computed based on the information provided by both the shape boundary and its interior region. Image moments are useful after segmentation to describe cell characteristics that uniquely describe the shape of that cell. Low order moments are used to derive simple properties including area, total intensity, centroid, skewness, kurtosis and information about the cell’s orientation. Moment invariant values are invariant to translation, scale and rotation of the cell [12]. Although wavelet transform (discussed in section 2.4) is scale invariant, it is not in all cases translation or rotation invariant [13], this is an additional advantage of moment invariants in our measurements. Moment Invariants have been frequently used as features for image processing, remote sensing, shape recognition and classification. They showed to be fairly reliable at distinguishing certain classes of topographic objects [12] as well as in many other applications [5]. In yeast studies, the first and second moment invariants were the top predictors to classify virulent from non virulent cells [14].

The set of seven moment invariants proposed by Hu are widely known, and hence we adopted them in our study [15, 7]. We define Hu’s set as in Eq. 9, where $\Phi_1$ and $\Phi_2$ are invariants based on second order moments, while $\Phi_3$ ... $\Phi_7$ are invariants based on third order moment.

$$hu = \{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7\}. \tag{9}$$

The effectiveness of moment invariants will increase when fused with the results of other techniques [12]. In this research we fuse them with second order statistical texture measurements obtained from the co-occurrence matrix and wavelet-based texture measurements to get the best from these approaches in the classification step required to discriminate between various cell conditions. The co-occurrence matrix based features and the wavelet-based texture features are discussed in the following sub-sections.
### Table 1: Features based on first order histogram

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>The number of pixels occupied by the cell.</td>
</tr>
<tr>
<td>Total Intensity</td>
<td>Sum of the intensity values of the pixels occupied by the cell.</td>
</tr>
<tr>
<td>Intensity Standard Deviation</td>
<td>The standard deviation from the mean (intensity/pixel) of the intensity values at each pixel.</td>
</tr>
<tr>
<td>Perimeter</td>
<td>The perimeter representation method used to estimate the perimeter of the cell is that of Vossepoel and Smeulders [9].</td>
</tr>
<tr>
<td>Circularity</td>
<td>The circularity of detected shapes [10].</td>
</tr>
<tr>
<td></td>
<td>[ Circularity = \frac{4\pi \cdot \text{Size}}{\text{perimeter}^2}. ]</td>
</tr>
<tr>
<td>Vacuole Size</td>
<td>If the fluorescent protein is expressed in the cytoplasm and nucleus, but not in the vacuole, the size of the central vacuole can be estimated. This is done by using a vacuole filter algorithm that looks in the fluorescent images for a region, inside the RoI (region of Interest) representing every cell, that forms the largest connected region with the lowest intensity values.</td>
</tr>
<tr>
<td>Membrane Features</td>
<td>Different features can be measured in the membrane pixels surrounding the cell border. Such features include size, total Intensity, Intensity standard deviation.</td>
</tr>
</tbody>
</table>
| Variance                | The variance (\( \mu_2 \) or \( \sigma^2 \)) is a measure of intensity contrast and can be computed from the second statistical moment. \[
\mu_2(z) = \sum_{i=0}^{L-1} (z_i - m)^2 \cdot P(z_i) \] \[ (4) \] \where \( z_i \) is the intensity value of the histogram at location \( i \), \( m \) the mean intensity value, \( L \) the total number of intensity levels (histogram range), and \( P(z_i) \) is the corresponding histogram with \( i \) between 0 and \( L-1 \). [11])
| Relative Smoothness     | The variance (\( \sigma^2 \)) is used to establish the descriptor of relative smoothness (R): \[
R(z) = 1 - \frac{1}{1 + \sigma^2(z)} \] \[ (5) \] This measure is zero for areas of constant intensities where the variance is zero there, and it approaches 1 for large values of the variance [11].
| Skewness                | The skewness (\( \mu_3 \)) of the intensity histogram which is the third statistical moment. \[
\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 \cdot P(z_i) \] \[ (6) \] A negative skewness means that most of the pixel values are high and thus concentrated at the right side of the histogram. A positive skewness means that most of the pixel values are low and thus concentrated at the left side of the histogram. [11])
| Uniformity              | The uniformity (U) has a maximum value for a cell image in which all intensity levels are equal [11]. \[
U(z) = \sum_{i=0}^{L-1} P^2(z_i) \] \[ (7) \]
| Entropy                 | The Entropy (e), which is a measure of variability, is zero for constant images [11]. \[
e(z) = - \sum_{i=0}^{L-1} P(z_i) \cdot \log_2 P(z_i) \] \[ (8) \]
2.3 Co-occurrence matrix based features

The simplicity of texture attributes can not completely characterize texture of the cells. Studies state that similar textures agree in their second-order statistics [5] and hence textures can be discriminated if they differ in their second-order statistics. Therefore one of the major statistical methods used in texture analysis is the one based on the definition of the joint probability distribution of pairs of pixels. Methods based on second-order statistics, i.e. statistics given by pairs of pixels, have been shown to achieve good discrimination rates in texture classification [16], and considered to be important in automated image analysis [17]. The second-order statistical features for texture analysis are derived from the co-occurrence matrix [6]. They were demonstrated to feature a potential for effective texture discrimination in biomedical images as well [18]. The second-order histogram is defined as the co-occurrence matrix \( h_{d\theta}(i, j) \). When divided by the total number of neighbouring pixels \( R(d, \theta) \) in the image, this matrix becomes the estimate of the joint probability \( p_{d\theta}(i, j) \) of two pixels, a distance \( d \) apart along a given direction \( \theta \) having particular (co-occurring) values \( i \) and \( j \) [5]. Formally, for image \( f(x, y) \) with a set of \( L \) discrete intensity levels, the matrix \( h_{d\theta}(i, j) \) is defined such that its \((i, j)^{th}\) entry is equal to the number of times that: \( f(x_1, y_1) = i \) and \( f(x_2, y_2) = j \), where \((x_2, y_2) = (x_1, y_1) + (d\cos\theta, d\sin\theta)\). This yields a square matrix of dimension equal to the number of intensity levels in the image, for each distance \( d \) and orientation \( \theta \). Most relevant co-occurrence matrix derived features used for the purpose of texture discrimination are the angular second moment, correlation, inertia, absolute value, entropy and maximum probability [6, 19]. Table 2 lists descriptions for these features. In this research we calculated the measures at distance \( d = 1 \) for horizontal, vertical and diagonal orientations at \( \theta = 0^\circ, 90^\circ, 45^\circ \) and \( 135^\circ \).

2.4 Multi-scale features and Wavelet-based texture measurement

Various methods adopted for calculating multi-scale features. The most commonly used are the Wigner distributions, Gabor function and wavelet transforms. Wigner distribution are found to possess interference terms between different components of a signal. These interference terms lead to wrong signal interpretation. Gabor filters are criticized for their non-orthogonality that result in redundant features at different scales or channel. On the other hand, the wavelet transform, being a linear operation, does not produce interference terms nor redundant features. For this reason, our interest is in the application of the wavelet transform to texture analysis. Discrete wavelet transform (DWT) derived features appear to be a suitable tool to be used for digital image texture analysis, because they allow analysis of images at various levels of resolution. The DWT provides powerful insight into an image’s spatial and frequency characteristics [20]. Moreover, it has shown to be an efficient descriptor for phenotyping [21]. In general, wavelet analysis is highly capable of revealing aspects of data such as trends, breakdown points, discontinuities in higher derivatives and self similarity [22]. Approximations and details are the most important terms in wavelet analysis. The approximations are the high-scale, low-frequency components of the image signal, while the decomposition process in the wavelet transform generates the coefficient matrices for the level-one approximation and horizontal, vertical and diagonal details. In this study, we include a bi-orthogonal wavelet in which texture details are derived from the three different directions on the same scale as the original image [21].

Table 2: Co-occurrence-matrix based features

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angular second moment (ASM)</td>
<td>Also known as uniformity and it is a measure of cell homogeneity. The maximum value is achieved when all the elements in the co-occurrence matrix are equal.</td>
</tr>
<tr>
<td></td>
<td>[ ASM = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} [p(i, j)]^2 ] (10)</td>
</tr>
<tr>
<td>Correlation</td>
<td>The correlation measures the dependencies between the yeast cell image pixels. ( \mu_x, \mu_y ) and ( \sigma_x, \sigma_y ) denote the mean and standard deviations of the row and column sums of the co-occurrence matrix respectively.</td>
</tr>
<tr>
<td></td>
<td>[ Correlation = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{ijp(i, j) - \mu_x\mu_y}{\sigma_x\sigma_y} ] (11)</td>
</tr>
<tr>
<td>Inertia</td>
<td>Inertia is also known as contrast. It is calculated by squaring the subtraction of the examined pixel values. Thus, the minimum value is when the pixels have the same grey-level value, and the maximum is achieved when squaring the subtraction of L and 1, i.e. ( L^2 ).</td>
</tr>
<tr>
<td></td>
<td>[ Inertia = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i - j)^2p(i, j) ] (12)</td>
</tr>
<tr>
<td>Absolute value</td>
<td>It calculates the absolute value of the subtraction of the examined pixel values. Hence it ranges between 0 and L.</td>
</tr>
<tr>
<td></td>
<td>[ Absolute\ value = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1}</td>
</tr>
<tr>
<td>Inverse difference</td>
<td>It has relatively high value when the high value in the co-occurrence matrix are near the main diagonal, where the difference ( (i - j) ) is smaller there.</td>
</tr>
<tr>
<td></td>
<td>[ Inverse\ difference = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{p(i, j)}{1 + (i - j)^2} ] (14)</td>
</tr>
<tr>
<td>Entropy</td>
<td>The entropy measures the complexity of the texture. It is a measure of randomness, achieving its highest value when the elements in the matrix are maximally random.</td>
</tr>
<tr>
<td></td>
<td>[ \text{entropy} = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j)\log_2[p(i, j)] ] (15)</td>
</tr>
<tr>
<td>Maximum probability</td>
<td>Gives an indication of the strongest response in the co-occurrence matrix.</td>
</tr>
<tr>
<td></td>
<td>[ \text{Maximum probability} = \max_{i,j} p(i, j) ] (16)</td>
</tr>
</tbody>
</table>
3 Building a Classification Model

Our dataset consists of 1440 yeast cell instances belonging to two major classes, one representing cells expressing 14-3-3 proteins attached to a green fluorescent protein (GFP) in a low (0mM) NaCl medium, and the other representing same cell strains in a high (50mM) NaCl medium. All these cells are measured for the features mentioned in the previous section after segmenting them. Each instance \( I \) in this dataset is mapped to one element of the set \((p, n)\) of positive and negative class labels representing the two different cell classes of low and high NaCl medium respectively. We need to build a classification model to map from instances to predicted classes. Given a classifier and a test set of instances, a two-by-two confusion matrix, also known as contingency table is constructed to represent the dispositions of the set of instances. This matrix forms the basis for many metrics we used to evaluate the classifiers [23].

To find our best classifier system, we prepared an experiment in the Weka machine learning workbench [24] with 23 different linear and non-linear machine learning algorithms including the popular predictors such as decision trees, naive Bayes, least-square linear predictors, and support vector machines. We applied a supervised classification on our dataset of 1440 instances (cells) with 10 fold cross validation.

In the evaluation step, our primary focus was on the Area Under ROC (AUC), but since our dataset is imbalanced with a ratio of cells in a low NaCl to that in a high NaCl medium being 2.7 : 1, we carefully considered several other metrics of the minority class as well, especially the AUC for that class (referred to as \( A_{min} \)). We tested various sampling techniques and normalization schemes and feature selection algorithms and evaluated their effect on the final accuracy of the classifiers. In the coming sub-section, the sampling techniques and imbalanced dataset are discussed. The second sub-section address the cross validation technique used on the dataset. Then normalization schemes are discussed. Subsequently, the adopted feature selection algorithms are highlighted. After that, a little detail about the used evaluation metrics. The last subsection is about the classifiers considered in the comparison.

3.1 Imbalanced Dataset and Sampling Techniques

Our dataset \( S \) is considered imbalanced since it exhibits an unequal distribution between its positive (yeast cells in low NaCl medium) and negative (yeast cells in a high NaCl medium) classes. Hence, in this domain, we require a classifier that will provide high accuracy for the negative minority class without severely jeopardizing the accuracy of the positive majority class. Conventional evaluation practice of using singular assessment criteria, such as the overall accuracy or error rate, does not provide adequate information in the case of imbalanced learning because most standard algorithms assume or expect balanced class distributions or equal misclassification costs. The problem of learning from imbalanced data is a relatively new challenge that has attracted growing attention from both academia and industry. The induction rules that describe the minority concepts are often fewer and weaker than those of majority concepts, since the minority class is often both outnumbered and under-represented. Successive partitioning of the dataspace results in fewer and fewer observations of minority class examples resulting in fewer leaves describing minority concepts and successively weaker confidence
estimates. In addition, concepts that have dependencies on different feature space conjunctions can go unlearned by the sparseness introduced through partitioning. The application of sampling techniques has shown to improve classifier accuracy. Therefore, we considered three different sampling techniques, namely under-sampling, over-sampling and Synthetic Minority Oversampling technique (SMOTE).

We define subsets $S_{\text{min}} \subset S$ and $S_{\text{maj}} \subset S$ where $S_{\text{min}}$ is the set of minority class instances in $S$, and $S_{\text{maj}}$ is the set of majority class instances in $S$, so that $S_{\text{min}} \cap S_{\text{maj}} = \{\phi\}$ and $S_{\text{min}} \cup S_{\text{maj}} = \{S\}$. Random under-sampling removes data from the original data set. In particular, we randomly select a set of majority class instances from $S_{\text{maj}}$ and remove these instances from $S$ so that $|S| = |S_{\text{min}}| + |S_{\text{maj}}| - |E|$, where $E$ represents the set removed by the sampling procedure. Under-sampling readily gives us a simple method for adjusting the balance of the original data set $S$; however, removing instances from the majority class may cause the classifier to miss important concepts pertaining to the majority class.

In oversampling, multiple instances of certain examples become "tied" since it simply appends replicated data to the original dataset, leading to overfitting. In particular, overfitting in oversampling occurs when classifiers produce multiple clauses in a rule for multiple copies of the sample example which causes the rule to become too specific; although the training accuracy will be high in this scenario, the classification performance on the unseen testing data is generally far worse.

The synthetic minority oversampling technique (SMOTE), on the other hand, is a powerful method that has shown a great deal of success in various applications. It creates artificial data based on the feature space similarities between existing minority instances. Specifically, for subset $S_{\text{min}}$, consider the K-nearest neighbours for each instance $x_i \in S_{\text{min}}$ for some specified integer $K$; the K-nearest neighbours are defined as the $K$ elements of $S_{\text{min}}$ whose euclidean distance between itself and $x_i$ under consideration exhibits the smallest magnitude along the n-dimensions of feature space $X$. To create a synthetic sample, we randomly select one of the K-nearest neighbours, then multiply the corresponding feature vector difference with a random number $\in [0, 1]$, and finally add this vector to the minority instance $x_i \in S_{\text{min}}$ as depicted in Eq. 17, where $\hat{x}_i \in S_{\text{min}}$ is one of the K-nearest neighbours for $x_i$, and $\delta \in [0, 1]$ is a random number. Therefore, the resulting synthetic instance according to Eq. 17 is a point along the line segment joining $x_i$ under consideration and the randomly selected K-nearest neighbour $\hat{x}_i$ [25].

$$x_{\text{new}} = x_i + (\hat{x}_i - x_i) \times \delta,$$

(17)

### 3.2 Training, Testing, and Cross Validation

In our prediction problem to predict the cells cultured in high $NaCl$ medium vs. those in low $NaCl$ medium, the classification models are given a training dataset of known ground-truth data, and a testing dataset of unknown first-seen data against which the models are tested. In order to limit problems like over-fitting and give an insight on how the model will generalize to an independent dataset, the widely used 10-fold cross validation is considered [26]. Over-fitting occurs when the classification model does not fit this validation data as well as it fits the training data. Cross validation is important in protecting against testing hypotheses suggested
by the data, known as Type III errors [27]. Its advantage is that all observations are used for both training and validation, and each observation is used for validation exactly once.

3.3 Feature Scaling or Normalization

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step [28]. Since the range of values of the raw data in our dataset varies, some machine learning algorithms will not work properly without normalization. For example in distance classifiers, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. In our work we considered two well-known normalization techniques which are unit-length normalization (UL) and zero-mean and unit-variance normalization (MV) [29]. Feature normalization techniques represent a vital part in building the classification model. They aim at normalizing the individual components of the extracted feature vectors in such a way that the resulting vectors are better suited for classification.

Unit length normalization (UL) scales all of the components \( x_i \) (\( i = 1, 2, \ldots, d \)) of vector \( x \) of all instances in our dataset \( S \) in accordance with the expression in Eq. 18 to produce the normalized feature vector \( x^* \), where \( ||.|| \) denotes the norm operator, and \( x^*_i \) stands for the \( i^{th} \) component of the normalized vector \( x^* \).

\[
x^*_i = \frac{x_i}{||x||}, \ i = 1, 2, \ldots, d,
\]

(18)

The zero-mean and unit-variance normalization is defined in Eq. 19, where \( \mu \) denotes the mean value of the feature vector \( x \) and \( \sigma \) represents its standard deviation. The MV technique transforms the feature vector \( x \) to a random variable with a mean value of zero and variance of one. It is assumed the individual components of the feature vector are normally distributed.

\[
x^*_i = \frac{(x_i - \mu)}{\sigma}, \ i = 1, 2, \ldots, d,
\]

(19)

3.4 Attribute Selection

Attribute selection also known as variable or feature selection, can be defined as a process that chooses a minimum subset of \( M \) features from the original set of \( N \) features, so that the feature space is optimally reduced according to a certain evaluation criterion. The reduced feature space are the most important parameters which help in predicting the outcome. Finding the best feature subset is usually intractable and many problems related to feature selection have been shown to NP-hard. The objective of feature or variable selection is three-fold:

- improving the prediction performance of the classifiers.
- providing faster and more cost effective classifiers.
Selecting the most relevant variables is suboptimal for building our predictor, particularly if the variable are redundant or irrelevant. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. For our data, we considered three well-known feature selection algorithms which are the Information Gain (IG) method, Correlation Feature Selection (CFS) and Principal Component analysis (PCA).

3.5 Evaluation metrics, ROC and AUC

A receiver operating characteristic (ROC) curve is a two dimensional graphical plot that illustrates the performance of a binary classifier system, which predicts a two-class problem in which the outcomes are labelled either as positive (p) or negative (n) [30]. The curve is created by plotting the true positive rate (TPR) on the Y axis against the false positive rate (FPR) on the X-axis at various threshold settings. TPR is also known as sensitivity in biomedical informatics, or recall in machine learning [31]. TPR defines how many correct positive results occur among all positive samples available during the test. On the other hand, FPR also known in biomedical informatics as (1-Specificity) defines how many incorrect positive results occur among all negative samples available during the test [32]. Each prediction result or instance of a confusion matrix represents one point in the ROC space [33].

The ROC graphs are useful for evaluating our classifiers and visualizing their performance. ROC graphs have been successfully used in medical decision making, and in recent years, they are gaining popularity in machine learning and data mining research, due to the realization that scalar measures such as simple classification accuracy, error rate or error cost are often poor metrics for measuring performance. ROC graphs have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs. These characteristics have become increasingly important as research continues into the areas of cost-sensitive learning and learning in the presence of unbalanced classes [23]. ROC analysis provides tools to select possibly optimal models [34]. Classifiers appearing on the left-hand side of an ROC graph, near the X axis, may be thought of as conservative: they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper right-hand side of an ROC graph may be thought of as liberal: they make positive classification with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates. Many real world domains are dominated by large numbers of negative instances, so performance in the far left-hand side of the ROC graph comes more interesting. We have used the area under the ROC curve (AUC), also known as c-statistic [35], which is usually interpreted according to the following ratings: $x = 1$, perfect; $1 > x \geq 0.9$, excellent; $0.9 > x \geq 0.8$, good; $0.8 > x \geq 0.7$, fair; $0.7 > x \geq 0.6$, poor; $0.6 > x \geq 0.5$, fail (random guessing for AUC); $x < 0.5$, unacceptable [36]. AUC is a common statistic most often used for model comparison in the machine learning community [37]. Despite its popularity, some machine learning researches show that the AUC is quite noisy as a classification measure [38], and has some other significant problems in model comparison [39, 40]. Therefore, we also considered the standard
accuracy metric, which is widely used to get an additional insight into the results. More importantly, we considered the $AUC$ for the minority class ($A_{\text{min}}$) as well. $A_{\text{min}}$ reveals the dangers of blindly looking into the $AUC$ alone in the raw dataset classifiers evaluation. However, the $AUC$ becomes a safe metric when the data is preprocessed with a sampling technique as the result will show in the following section.

### 3.6 Classifiers Evaluated

In this work, 23 different linear and non-linear famous classification systems were evaluated on our dataset. A list of these models along with short descriptions are shown in Table 3. These classifiers were evaluated using the Weka machine learning workbench [24], and for ROC analysis, we use $pROC$ package [41] of the R programming language and environment for statistical computing [42] connected with Weka through the RWeka package [43]. The next section shows how these classifiers perform on our dataset.

### 4 Results

Using our dataset with the presented sophisticated features, we evaluated the classifiers by considering the area under ROC curve ($AUC$) as our main metric in addition to the minority class $A_{\text{min}}$ and ACC (accuracy) of each classification system. We listed in Table 4 the $AUC$, $A_{\text{min}}$ and ACC scores for each classifier under the best sampling, normalization and feature selection algorithms. Most classifiers have optimal results with the SMOTE sampling technique and the MV normalization scheme. The Feature Selection with the best results was the IG method. It is clear from the results that a number of classifiers were able to excellently discriminate between the two cell groups with $AUC$, $A_{\text{min}}$ and ACC metrics above 0.9. The following subsections reveal the power of sampling, the effect of normalization and feature selection, in addition to the power of the discriminators based on our designed feature space.

#### 4.1 Power of Sampling

The first obvious fact from the results in Table 4 is the power of data sampling on the classifiers performance. SMOTE has not failed to improve the overall classification within all the algorithms tested, unlike the under-sampling and over-sampling methods. This makes SMOTE an excellent choice for our data. Moreover, SMOTE addresses the information loss of the under-sampling and the over-representation issue of the over-sampling methods. After applying SMOTE, the $AUC$ was improved in average by .025, and the average accuracy increased slightly by 0.007. However, the strongly significant different is shown when considering the $AUC$ for the minority class ($A_{\text{min}}$) where the average improvement was increased by .139 per classifier from an average of .708 to .847. This also reveals the reason why accuracy is not a sufficient measure in our imbalanced dataset. Generally, the power of sampling was more conspicuous in PART, C4.5, JRIP, SMO and VFDT classifiers.

Table 3: Classification Algorithms evaluated in this study

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>A decision tree classifier. At each node of the tree, C4.5 chooses the attribute that most effectively splits its set of samples into subsets enriched in one class or the other [44].</td>
</tr>
<tr>
<td>Adaptive Boosting (AdaB)</td>
<td>A machine learning meta-algorithm used in conjunction with a weak learner (Decision Tree) to improve its performance [45].</td>
</tr>
<tr>
<td>PART</td>
<td>Builds a partial C4.5 decision tree in each iteration and makes the &quot;best&quot; leaf into a rule [46].</td>
</tr>
<tr>
<td>Decision Table Majority (DTM)</td>
<td>A decision table with a default rule mapping to the majority class. It has a set of features (schema) and a set of labelled instances (body) [47].</td>
</tr>
<tr>
<td>Decision Stump (DSmp)</td>
<td>A model consisting of a one-level decision tree. i.e., it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves) [48].</td>
</tr>
<tr>
<td>One Rule (OneR)</td>
<td>OneR generates one rule for each predictor in the data, then selects the rule with the smallest total error as its &quot;one rule&quot; [49].</td>
</tr>
<tr>
<td>JRip</td>
<td>JRip implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [50].</td>
</tr>
<tr>
<td>Bayes Network (BNet)</td>
<td>Bayes Network learning represents the dataset variables via a directed acyclic graph (DAG) based on probability theory [51].</td>
</tr>
<tr>
<td>K-Nearest Neighbour (IBK)</td>
<td>Predicts the class of the single nearest training instance for each test instance [52].</td>
</tr>
<tr>
<td>Locally Weighted Learning (LWL)</td>
<td>Uses an instance-based algorithm (Decision Stump) to assign instance weights [53].</td>
</tr>
<tr>
<td>LogitBoost (ALR)</td>
<td>Performs additive logistic regression on the base learner (Decision Stump) [54].</td>
</tr>
<tr>
<td>Random Committee (RCom)</td>
<td>Builds an ensemble of randomizable base classifiers (Random Tree). The final prediction is a straight average of the predictions generated by the individual base classifiers.</td>
</tr>
<tr>
<td>Random Subspace (RSub)</td>
<td>Constructs a decision tree based classifier (REPTree) with multiple trees constructed in randomly chosen subspaces [55].</td>
</tr>
<tr>
<td>Hoeffding Tree (VFDT)</td>
<td>An incremental decision tree induction algorithm capable of learning from massive data. It assumes that the distribution of variables does not change over time [56].</td>
</tr>
<tr>
<td>Logistic Model Tree (LMT)</td>
<td>A logistic model tree basically consists of a standard decision tree structure with logistic regression functions at the leaves [57].</td>
</tr>
<tr>
<td>REPTree</td>
<td>Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning.</td>
</tr>
<tr>
<td>Random Forest (RFor)</td>
<td>Constructs a forest of random decision trees at training time and outputting the mode class (classification) or mean prediction (regression) of the individual trees [58].</td>
</tr>
<tr>
<td>Random Tree (RTre)</td>
<td>Constructs a tree with randomly chosen attributes at each node.</td>
</tr>
<tr>
<td>Logistic (Log)</td>
<td>Building and using a multinomial logistic regression model with a ridge estimator [59].</td>
</tr>
<tr>
<td>Stochastic Gradient Descent (SGD)</td>
<td>Implements stochastic gradient descent for learning various linear models (binary SVM, binary logistic regression, squared loss, Huber loss and epsilon-insensitive loss).</td>
</tr>
<tr>
<td>Sequential Minimal Optimization (SMO)</td>
<td>Sequential minimal optimization algorithm for training a support vector classifier [60].</td>
</tr>
<tr>
<td>SimpleLogistic (SLog)</td>
<td>Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models [57].</td>
</tr>
<tr>
<td>Voted Perceptron (VPer)</td>
<td>Based on a linear predictor function combining a set of weights with the feature vector, and a transformation of online learning, in that it processes elements one at a time [61].</td>
</tr>
</tbody>
</table>
Table 4: AUC, $A_{min}$ and ACC of classification algorithms using raw dataset, and after sampling, normalization and feature selection.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Raw Dataset</th>
<th></th>
<th></th>
<th>Sampled</th>
<th></th>
<th></th>
<th>Normalized</th>
<th></th>
<th></th>
<th>Features Selected</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>$A_{min}$</td>
<td>ACC</td>
<td>AUC</td>
<td>$A_{min}$</td>
<td>ACC</td>
<td>AUC</td>
<td>$A_{min}$</td>
<td>ACC</td>
<td>AUC</td>
<td>$A_{min}$</td>
<td>ACC</td>
</tr>
<tr>
<td>RCom</td>
<td>.972</td>
<td>.937</td>
<td>.940</td>
<td>.984</td>
<td>.980</td>
<td>.943</td>
<td>.983</td>
<td>.979</td>
<td>.939</td>
<td>.982</td>
<td>.977</td>
<td>.939</td>
</tr>
<tr>
<td>RSub</td>
<td>.959</td>
<td>.923</td>
<td>.911</td>
<td>.978</td>
<td>.978</td>
<td>.928</td>
<td>.976</td>
<td>.976</td>
<td>.921</td>
<td>.975</td>
<td>.974</td>
<td>.927</td>
</tr>
<tr>
<td>RFor</td>
<td>.965</td>
<td>.917</td>
<td>.920</td>
<td>.981</td>
<td>.974</td>
<td>.937</td>
<td>.977</td>
<td>.972</td>
<td>.922</td>
<td>.978</td>
<td>.970</td>
<td>.930</td>
</tr>
<tr>
<td>SLog</td>
<td>.926</td>
<td>.841</td>
<td>.867</td>
<td>.926</td>
<td>.911</td>
<td>.854</td>
<td>.926</td>
<td>.911</td>
<td>.853</td>
<td>.926</td>
<td>.911</td>
<td>.854</td>
</tr>
<tr>
<td>Log</td>
<td>.945</td>
<td>.839</td>
<td>.898</td>
<td>.916</td>
<td>.925</td>
<td>.872</td>
<td>.916</td>
<td>.919</td>
<td>.872</td>
<td>.916</td>
<td>.920</td>
<td>.872</td>
</tr>
<tr>
<td>LogB</td>
<td>.930</td>
<td>.837</td>
<td>.878</td>
<td>.933</td>
<td>.918</td>
<td>.874</td>
<td>.932</td>
<td>.918</td>
<td>.868</td>
<td>.932</td>
<td>.918</td>
<td>.868</td>
</tr>
<tr>
<td>DTab</td>
<td>.922</td>
<td>.833</td>
<td>.876</td>
<td>.942</td>
<td>.933</td>
<td>.872</td>
<td>.942</td>
<td>.936</td>
<td>.873</td>
<td>.943</td>
<td>.936</td>
<td>.872</td>
</tr>
<tr>
<td>LMT</td>
<td>.927</td>
<td>.808</td>
<td>.901</td>
<td>.937</td>
<td>.898</td>
<td>.908</td>
<td>.936</td>
<td>.896</td>
<td>.905</td>
<td>.937</td>
<td>.895</td>
<td>.904</td>
</tr>
<tr>
<td>REPT</td>
<td>.891</td>
<td>.789</td>
<td>.887</td>
<td>.932</td>
<td>.912</td>
<td>.892</td>
<td>.923</td>
<td>.903</td>
<td>.887</td>
<td>.923</td>
<td>.904</td>
<td>.887</td>
</tr>
<tr>
<td>AdaB</td>
<td>.908</td>
<td>.767</td>
<td>.855</td>
<td>.919</td>
<td>.899</td>
<td>.850</td>
<td>.917</td>
<td>.895</td>
<td>.849</td>
<td>.917</td>
<td>.895</td>
<td>.849</td>
</tr>
<tr>
<td>PART</td>
<td>.884</td>
<td>.764</td>
<td>.901</td>
<td>.916</td>
<td>.892</td>
<td>.904</td>
<td>.932</td>
<td>.906</td>
<td>.910</td>
<td>.925</td>
<td>.901</td>
<td>.909</td>
</tr>
<tr>
<td>C4.5</td>
<td>.853</td>
<td>.747</td>
<td>.893</td>
<td>.915</td>
<td>.885</td>
<td>.914</td>
<td>.900</td>
<td>.864</td>
<td>.906</td>
<td>.898</td>
<td>.861</td>
<td>.906</td>
</tr>
<tr>
<td>JRPt</td>
<td>.853</td>
<td>.747</td>
<td>.893</td>
<td>.916</td>
<td>.887</td>
<td>.905</td>
<td>.918</td>
<td>.891</td>
<td>.894</td>
<td>.911</td>
<td>.877</td>
<td>.894</td>
</tr>
<tr>
<td>BNet</td>
<td>.881</td>
<td>.703</td>
<td>.808</td>
<td>.888</td>
<td>.853</td>
<td>.830</td>
<td>.889</td>
<td>.862</td>
<td>.824</td>
<td>.889</td>
<td>.862</td>
<td>.824</td>
</tr>
<tr>
<td>SGD</td>
<td>.950</td>
<td>.683</td>
<td>.888</td>
<td>.869</td>
<td>.815</td>
<td>.869</td>
<td>.866</td>
<td>.810</td>
<td>.864</td>
<td>.866</td>
<td>.810</td>
<td>.864</td>
</tr>
<tr>
<td>RTre</td>
<td>.853</td>
<td>.669</td>
<td>.883</td>
<td>.877</td>
<td>.830</td>
<td>.877</td>
<td>.879</td>
<td>.835</td>
<td>.879</td>
<td>.875</td>
<td>.826</td>
<td>.874</td>
</tr>
<tr>
<td>IBK</td>
<td>.851</td>
<td>.668</td>
<td>.881</td>
<td>.881</td>
<td>.839</td>
<td>.881</td>
<td>.881</td>
<td>.839</td>
<td>.881</td>
<td>.881</td>
<td>.839</td>
<td>.881</td>
</tr>
<tr>
<td>LWL</td>
<td>.870</td>
<td>.644</td>
<td>.736</td>
<td>.874</td>
<td>.819</td>
<td>.773</td>
<td>.874</td>
<td>.819</td>
<td>.773</td>
<td>.874</td>
<td>.819</td>
<td>.773</td>
</tr>
<tr>
<td>SMO</td>
<td>.803</td>
<td>.628</td>
<td>.867</td>
<td>.866</td>
<td>.815</td>
<td>.866</td>
<td>.866</td>
<td>.814</td>
<td>.865</td>
<td>.866</td>
<td>.814</td>
<td>.865</td>
</tr>
<tr>
<td>OneR</td>
<td>.767</td>
<td>.553</td>
<td>.834</td>
<td>.795</td>
<td>.733</td>
<td>.794</td>
<td>.795</td>
<td>.733</td>
<td>.794</td>
<td>.795</td>
<td>.733</td>
<td>.794</td>
</tr>
<tr>
<td>DSmnt</td>
<td>.769</td>
<td>.450</td>
<td>.733</td>
<td>.774</td>
<td>.692</td>
<td>.774</td>
<td>.774</td>
<td>.692</td>
<td>.774</td>
<td>.774</td>
<td>.692</td>
<td>.774</td>
</tr>
<tr>
<td>VPer</td>
<td>.543</td>
<td>.289</td>
<td>.638</td>
<td>.526</td>
<td>.515</td>
<td>.527</td>
<td>.802</td>
<td>.735</td>
<td>.798</td>
<td>.802</td>
<td>.736</td>
<td>.798</td>
</tr>
<tr>
<td>VFDT</td>
<td>.500</td>
<td>.267</td>
<td>.733</td>
<td>.638</td>
<td>.582</td>
<td>.606</td>
<td>.736</td>
<td>.668</td>
<td>.661</td>
<td>.736</td>
<td>.668</td>
<td>.661</td>
</tr>
</tbody>
</table>

4.2 Effect of Normalization and Feature Selection

Normalization showed an extra average improvement of .015 and .013 for the AUC and $A_{min}$ respectively, and improved accuracy by 0.012 over the sampled dataset. The most significant difference of normalization was shown in VPer and VFDT classifiers. However, the best feature selection algorithm (IG) showed no change in the averages of AUC and $A_{min}$ and only a little accuracy increase for the RFor and RSub models. The negligible effect of feature selection could be addressed to our choice of well composed relevant features.

4.3 The Powerful discriminators

Wavelet-based texture measurement has shown their superiority to discriminate our instances in the top classifiers. Specifically, the Smoothness and Uniformity textures in the horizontal, diagonal and vertical wavelet detail images were used as root nodes in most base trees in the RCom, RFor, RSub and C4.5 Decision tree classifiers. In addition, the fused invariant moments with wavelet details in many dimensions obtained high weights in the functions of SLog and LMT classifiers revealing their discriminative power. They also showed up multiple times within the
decision trees built by various models. The second order histogram features extracted from the co-occurrence matrix had also played a major role in the discrimination of yeast cells, they showed up in almost every classifier, though at lower level in the decision trees or with a smaller weight in the regression functions.

Most of the top classifiers built complex models that are not easy to interpret. RCom has built ten random trees of sizes between 267 and 297. RSub built a ”relatively” less complex model of ten random REP trees of sizes between 47 and 87. RFor built ten random trees each using seven random attribute values in its construction. On the other hand, the Logistic and Simple Logistic algorithms built decent regression functions. Simple Logistic built its function with a few attributes (22 of the total 90 considered). This makes Logistic regression a much more appealing classification model for our domain. The Logistic regression classifier has two output terms, coefficients and odds ratios. High coefficient values when predicting the negative class were noticed in many moment invariant features in the wavelet detail images. High odds ratios were noticed in texture measurements and co-occurrence matrix features as well. The Simple Logistic has used in its function, eight features from the co-occurrence matrix, five features from the wavelet texture measurement, two features from the combination of wavelet and moment invariants, four features from moment invariants, one texture measure and two basic shape descriptors.

Support Vector Machines (SVMs) are famous classifiers. The SMO, which is an implementation of the SVMs, did not rank as an excellent classifier when using the default parameters. However, this changes when optimizing its parameters by increasing the complexity constant to 5, disabling any normalization within the classifier itself, fit logistic models to SVM outputs and use a normalized Polynomial Kernel. This optimization pushed the SMO into the top five classifiers with an AUC of 0.92.

Next we study the considered feature sets for their contribution to the power of discrimination.

### 4.4 Feature spaces performance

To investigate whether our composed feature sets has any added value to the discrimination power, we started by comparing the classification performance using different set of features including basic shape descriptors, invariant moments, wavelet texture measurement, invariant moments on wavelet detail images, co-occurrence matrix derived features, basic texture measurement and a full feature space combining all the feature sets. The difference is shown in Fig. 2 and 3. This test was performed on both Logistic and C4.5 classifiers. Logistic regression was chosen as it is the top ”non-random based” classifier and C4.5 as a non-linear approach for comparison. In both classifiers, the benefits of using the full set is obvious. In the Logistic classifier, the performance of individual feature sets were not sufficient, except for the basic texture measurement which shows a very good discrimination with an AUC of 0.82. However, using the full features set shifted the performance of the Logistic classifier into the excellent category with an AUC of 0.92. In the non-linear C4.5 decision tree classifier, all the individual feature sets except the basic set have good discrimination. However, none is ranked as excellent. Only when the feature sets are fused together the classifier has an excellent discrimination rate with an AUC of 0.91.
Since 3rd order moment invariants might be more noise-prone than their 2nd order counterpart, we study whether it really adds any discrimination value by creating only two small feature spaces. The first having only the second moment invariants while the second feature space contains the whole set of seven invariant moments. Figure 4 and 5 show the ROC graph of the performance of the Logistic and C4.5 classifiers on both feature spaces. The third order moments show to have an additional discriminative value in both classifiers for this dataset.

5 Conclusion

In this paper, we addressed our principal research question on how a machine learning approach can discriminate S. cerevisiae yeast cells cultivated in high or low NaCl medium? and the sub-questions on what extracted object features are the best to predict whether a measured cell belongs to high or low sodium chloride (NaCl) medium? and what machine learning algorithm can best work on our dataset of individual cell measurements? Moreover, what are the best sampling, normalization and feature selection algorithms to be used on such dataset? From our experiment, we show that a machine learning approach is efficient in our classification problem. The Wavelet-based texture measurements, co-occurrence matrix derived features, moment invariant features and texture measurement derived from the image histogram, forms a set of powerful discriminators in the top classification models, from which the Logistic model was chosen as the most efficient for classifying our cells. Furthermore, optimization of an SvM classifier might be possible. Sampling of our dataset with the SMOTE method showed to have a significant effect on building the classification model system. In addition, the MV normalization scheme showed an extra improvement. However, the best feature selection algorithm tested showed a little non-significant improvement probably due to the fact that we selected the right feature sets. With this machine learning process and the chosen feature sets, it becomes possible
as future work, to classify different cell strains and conditions in a high-volume high-throughput studies.

6 Acknowledgement

We would like to thank Paul van Heusden who offered us the yeast cell images. Moreover, this work is partly supported by the Erasmus Mundus JOYSLEEN project, the Landelijke Stichting voor Blinden en Slechtziendenand (LSBS) and the Raymond-Sackler organizations.

References


derived features for digital image texture analysis. In Proc. of International Conference on

bioinformatics, Liacs, Leiden University, 2014.


2006.

Morgan Kaufmann, 2005.

[25] H. He and E. A. Garcia. Learning from imbalanced data. Knowledge and Data Engineer-

IEEE, 2011.

a weighted cross-validation evolutionary artificial neural network ensemble. Neurocom-

and crosscorrelated signal features. In Industrial and Information Systems (ICIIS), 2014

[29] V. Štruc and N. Pavešić. A comparison of feature normalization techniques for pca-based

Rfid-coa: The rfid tags as certificates of authenticity. In RFID (RFID), 2011 IEEE Inter-

invasive prostate cancer screening based on serum surface-enhanced raman spectroscopy

In Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories

[33] G. Rastrelli, G. Corona, L. Vignozzi, E. Maseroli, A. Silverii, M. Monami, E. Mannucci,
G. Forti and M. Maggi. Serum psa as a predictor of testosterone deficiency. The journal
of sexual medicine, 10(10):2518–2528, 2013.


