Automatic Classification of C. Elegans Mitochondria using Machine Learning Algorithms

Abstract

Morphology of mitochondria is interesting for cell biologists because it is believed to be linked to various biological pathways and processes. Researchers from the Cell and Developmental Biology Department of Ludwig Maximilians Universität Munich examine these organelles from a free-living worm under a fluorescence microscope and classify them into two categories based on their shapes. This classification is performed by biologists by looking at the sharpest image from the stack of images of the worms obtained from the microscope. This process is biased because the biologists working with the worm have prior expectation on the morphology. Moreover, the human based image analysis is time consuming. There is, hence, a need for an unbiased metric for the morphology of mitochondria, and possible automation of the morphometric analysis. In this report, the authors suggest use of two features for classification of mitochondria and present their findings on performance of the machine learning algorithms MLE, logistic regression, SVM and neural networks on the classification task.

1. Background

Mitochondria are cell organelles that generate energy currency of the cell called ATPs by a series of chemical reactions in the body. ATP can then be readily absorbed by the cell to meet its energy demands. In doing so, mitochondria also generate free radicals mostly in the form of reactive oxygen species as a by product of the reactions. These free radicals are harmful to the cell. There is an ongoing research at Cell and Developmental Biology Department at LMU about how mitochondria changes its morphology with age in the free-living transparent worms called *Caenorhabditis elegans.* The hypothesis in the field is that mitochondrial production of ATP over time damages cells and brings about aging. Moreover, not only the morphology of single mitochondria, but also the dynamics of mitochondiral fission and fusion in a cell are associated with cell death (Eltyeb Abdelwahid, 2011).

In order to analyse mitochondrial morphology, worms with fluorescent mitochondria are put under a fluorescence microscope and 10 to 20 pictures of it are generated. As the worm is transparent, photographing it with different focuses yields to photographs of different levels of the worm. After that, each image "*slice*" of the worm is scrutinized on the computer screen to find just one picture where cells are clearly visible. Using computer imaging software, the photographs are discretized until mitochondria can be separated from background fluorescence. Afterwards, another routine of this program computes different attributes of the shape of the areas refined during the previous step. These are attributes like "circularity" (shape compared to a perfect circle), length of one mitochondria, kind of clustering, etc. These data are used to classify the morphology of the mitochondria into two classes: *fragmented* and *tubular.* As this process involves human based classification, it is prone to biases based on the hypothesis of the experiment. Hence, there was a need for an automatic classification of the organelles based on just the images.

This report presents an overview of four machine learning algorithms along with the findings regarding their accuracy, for this classification task.

2. Data Pre-processing and Feature Extraction

This section describes how the images in the stack file are processed before classification can be done.

2.1. Data Pre-processing

To extract relevant features from the images preprocessing is necessary. To do so, a pre-processing pipeline

Report submitted for review for Machine Learning course project (IN20164).

comprising of three steps was created.

Sharpest Image: Each image stack consists between around 10 to 20 photographs of the same part of one worm. The images in a stack differ are taken at different focal planes of the microscope. In most of the cases only one of these images shows the mitochondria in a sharp and clear way that can be used for classification. The usual process is to look at each of the images in the stack, select the sharpest and proceed with the next classification tasks only with the sharpest image. If a program is to be able to classify the mitochondria automatically, it should be able to select the sharpest image automatically as well.

For this purpose a simple gradient based estimation function is used. The grayscale image is represented by the matrix $I \subseteq \mathbb{N}^{m \times n}$ with values in the range $[0, 255]$ with 0 for black and 255 for white pixels. The sum over the gradients I_x and I_y in both image dimensions x and y normalized by the number of gradients gives a good estimation on how sharp a photograph actually is.

$$I_x = \frac{\partial I}{\partial x} \qquad I_y = \frac{\partial I}{\partial y} \tag{1}$$

$$s = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij}}{mn} \tag{2}$$

$$S_{ij} = \sqrt{({I_{x}}_{ij}^{\,2} + {I_{y}}_{ij}^{\,2})} \tag{3}$$

The higher the value s the sharper the image. A disadvantage of this simple method is that, one usually can only compare images showing the same motif. Hence, in real image processing software much better methods for estimating image sharpness are in use. Nevertheless it has proved itself to be a sufficient method for the purpose of classification presented in this report. This is because only images within a single stack are compared, and they all show the exact same photograph.

Histogram Analysis: To adjust the optimal contrast of each image it is necessary to have a look at the histograms and how the different levels (0 to 255) of gray are distributed all over one image.

The task is to find suitable values for b_l, the lower bound, below which everything goes to 0 (black) and b_u, the upper bound, above which everything goes to 255 (white).

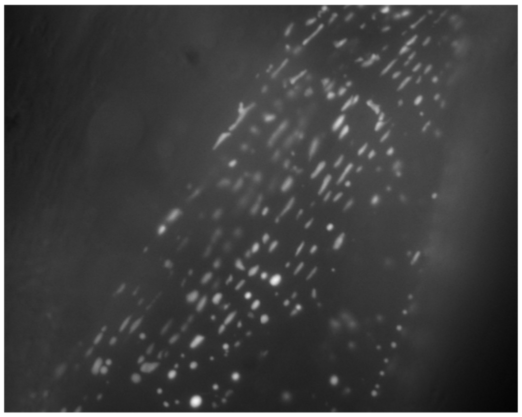
Figure 1. Fragmented mitochondria

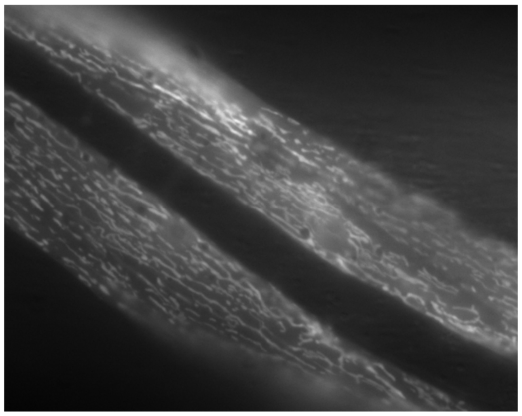
Figure 2. Tubular mitochondria

$$f_l(l) = \sum_{i=1}^{l} h_i \leq \beta \sum_{i=1}^{256} h_i \qquad (4)$$

$$f_u(u) = \sum_{i=1}^{u} h_{256-i+1} \leq \alpha \sum_{i=1}^{256} h_i \qquad (5)$$

$$1 \leq l \leq 256 \qquad 1 \leq u \leq 256 \qquad (6)$$

$$b_l = \frac{\text{argmax}_b(f_b)}{256} \qquad b_u = \frac{\text{argmax}_u(f_u)}{256} \qquad (7)$$

$$\alpha + \beta \leq 1 \qquad (8)$$

The gray scale levels in between are scaled to fit into the histogram of the output image. The result is a high contrast image in which mitochondria are much easier to find. Because all the images have different histograms, it is not possible to use the same lower bound and upper bound for all of them. So the variables α and β were defined, corresponding to how many of the pixels should have the grayscale value of 0 (black) or 255 (white) after the adjustment. Using this procedure the optimal bounds for each individual image were found. The performance of the later classification process depends highly on the quality of output of this contrast adjustment. In order to find the best values for α and β we used the optimization tools of MATLAB. Misclassification rate on the training set is used as the optimization objective.

Boundaries and Blob Detection: After improving the contrast of each image it can easily be converted to a binary image in that the mitochondria can be found as white areas. Using the MatLab function '*bwboundaries*' it was possible trace the exterior boundaries of the mitochondria in the binary image. After filtering out objects that are too big or too small to be valid mitochondria a set of vectors were generated, each containing the borders of one of the mitochondrion.

2.2. Feature extraction

Circularity: The circularity of mitochondria is one of the most important and reasonable ways to compare mitochondria in tubular and fragmented cells. If the cell is classified as *tubular*, the bounds of each mitochondrion are far from being circularly dirtibuted whereas in a fragmented classified cell almost all of the mitochondria are indeed shaped like circles.

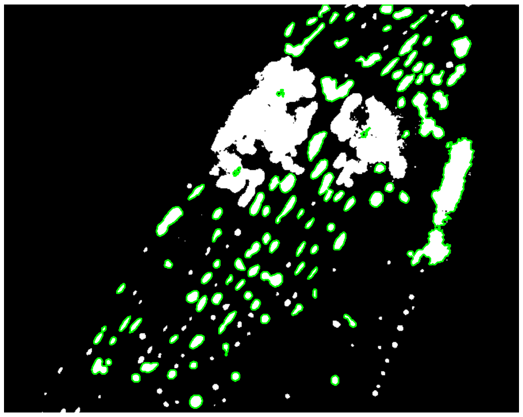

Figure 3. Detected fragmented mitochondria

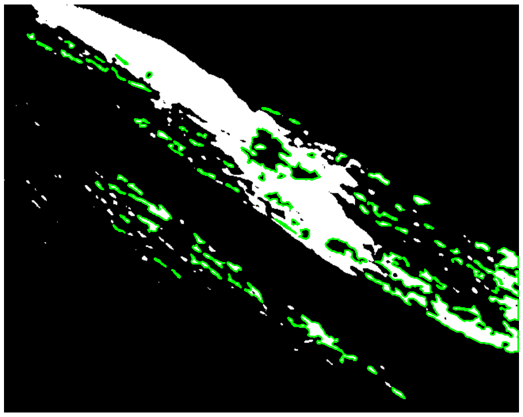

Figure 4. Detected tubular mitochondria

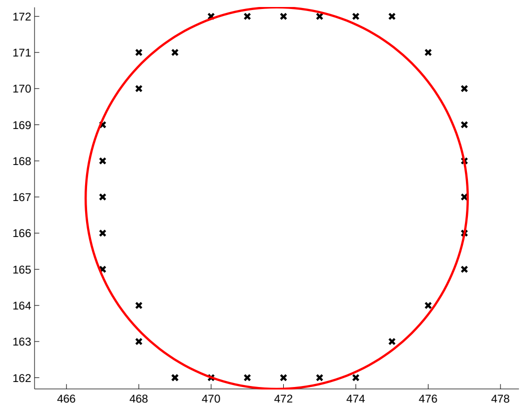

Figure 5. Circular distributed boundary of a blob

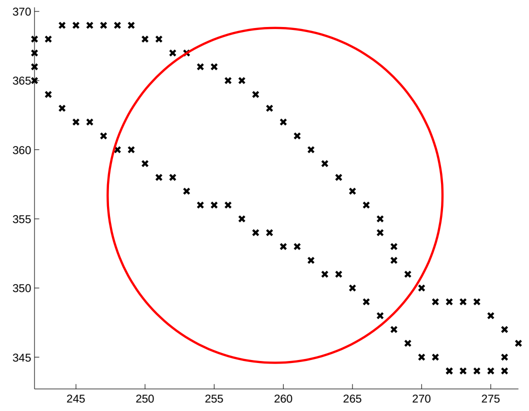

Figure 6. Tubuar shaped boundary of a blob

Since the boundaries have already been found in an earlier pre-processing step, FuzzyCShells (Runkler, 2010) method was used to calculate the error that represents the average of the distance between the points and the circle. The centre of the circle is the mean μ of the data set of size N. The radius r giving the smalles error can be calculated in one step, too.

$$r = \frac{\sum_{i=1}^{N} \|x_i - \mu\|}{N} \tag{9}$$

The distance between each of the data points and the circle is given by

$$d(x_i, (\mu, r)) = |\|x_i - \mu\| - r|. \tag{10}$$

The error value to compare the circularity is then given by the sum over all distances $d(x_i, (\mu, r))$ between the data points x_i and the circle normalized by the number of data points N.

$$e_c = \frac{\sum_{i=1}^{N} d(x_i, (\mu, r))}{N} \tag{11}$$

An error $e_c = 0$ means that the blob detected is perfectly shaped like a circle.

Covariance Analysis: The covariance analysis is yet another way to generate a feature based on the shape of a blob. If data points are distributed in a tubular fashion, then the covariance of the points has two very different values for the covariance's eigenvalues λ_b and λ_s, where $\lambda_b \geqslant \lambda_s$. The biggest eigenvalue λ_b will belong to the eigenvector pointing in the direction along which the original blob data is distributed.

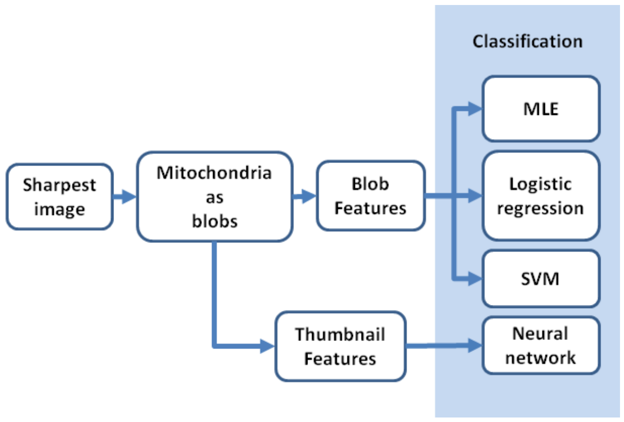

Figure 7. Overview of the classification pipeline

So the ratio of the two eigen values, e_λ, is close to zero if the data is distributed as a line.

$$e_\lambda = \lambda_b / \lambda_s \tag{12}$$

Thumbnail Features: After blob detection, the ratio of the eigenvalues and the circularity measure were passed as input to various classification algorithms as shown in figure 7. Note that features for neural network is slightly different from the rest of classification methods.

For neural network (only) the blobs have been used directly to train the model and make predictions. Instead of computing e_c and e_λ for each blob and using these values as features for classification, the pixel values of the blob are used directly to make predictions. After blobs are detected, each blob is resized to fit into a 20×20px image. The blobs are then centered in the image. A binary image is created by filling the blob with grayscale value of 255 (white) and that for rest of the image is set to 0 (black). Vectorizing these single blob images produces a 400 dimensional feature space that is then used to train a neural network and make predictions.

3. Classification

Four machine learning algorithms: maximum likelihood estimation (MLE), logistic regression, support vector machine (SVM) and neural networks (NN), were used for classifying the mitochondria from the sharpest image. Each of them is discussed in detail in this section.

3.1. MLE

For the MLE classification, it is assumed that the data is gaussian distributed for each of the classes.

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{13}$$

To fit a multivariate Gaussian (equation 13) on the data of one class, it is necessary to compute the maximum values for the mean μ and the covariance matrix Σ. This can be done by solving the closed form equations 14 and 15.

$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} \tag{14}$$

$$\Sigma_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\mu)(x^{(i)}-\mu)^T \tag{15}$$

The MLE optimization is repeated for each of the classes, i.e. *fragmented* and *tubular*, to evaluate μ_f, μ_t, Σ_f and Σ_t. In 2D the Gaussians can be shown as ellipses over the data points (see figure 8). To make predictions using this model for new samples, the values for both Gaussians $p(x|\mu_f,\Sigma_f)$ and $p(x|\mu_t,\Sigma_t)$ are calculated. The blobs are assigned to the class whose Gaussian returns the highest value.

Figure 8 shows the ellipses corresponding to the Gaussians belonging to the two classes and the decision boundary. Despite the fact that MLE classification is one of the simplest methods, because there exists a closed form and no complicatet optimization algorithms are needed, it turned out to be the model with the best classification results for the mitochondria data.

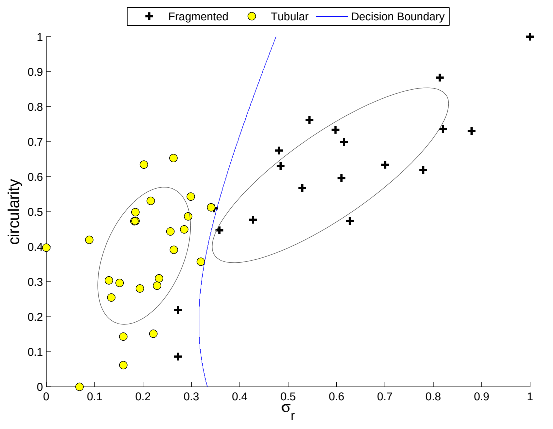

Figure 8. Multivariate Gaussians fitted on data set

3.2. Logistic Regression

Unregularized logistic regression with the objective function $J(\theta)$ and the gradient as shown in equations 16 and 17 were used for classifying the data, where m represents the number of training samples, $x^{(i)}$represents the i^{th}training example, $y^{(i)}$represents the label for $x^{(i)}$, and $h_\theta(x)$ means the prediction on x. The input for the logistic regression learning algorithm was the mean of the ratio of eigenvalues and the circularity measures. Standard MATLAB optimization function *fminunc* was used for optimizing the cost function, with maximum number of iterations set to 500.

$$\frac{1}{m}\sum_{i=1}^{m}\left[-y_i\log\left(h_\theta\left(x^{(i)}\right)\right)-(1-y_i)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right] \tag{16}$$

$$\frac{\partial J(\theta)}{\partial\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)x_j^{(i)} \tag{17}$$

The prediction is made by using the standard sigmoid function once the weights are learned. Figure 9 shows decision boundary for the classification. For the entire data set, logistic regression seems to perform classification quite well.

3.3. SVM

Support vector machines (SVM) belong to a group of classifiers called sparse kernel machines (Bishop, 2006). A big advantage of these machines is that they have sparse solutions, so that the predictions for new inputs depend only on the kernel function eveluated

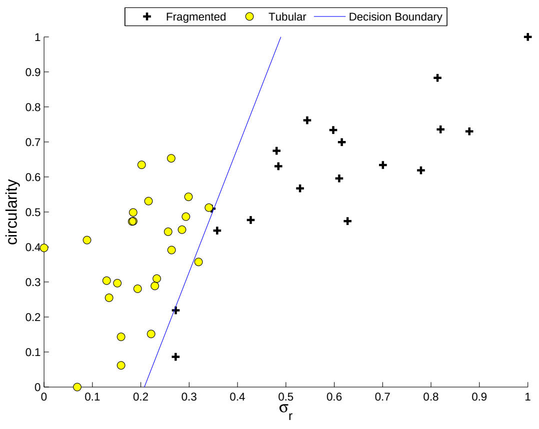

Figure 9. Classification using logistic regression

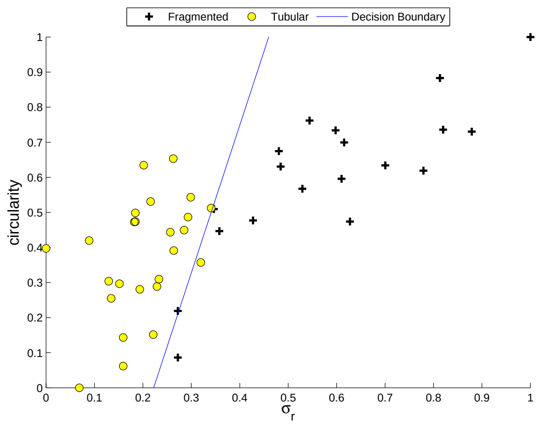

Figure 10. Linear SVM

at a subset of the training data points. The result of their computation is a decision boundary. This boundary can be linear or non-linear, depending on the used kernel.

Linear SVM The linear SVM computes a linear decision boundary that sepperates the two classes, fragmented and tubular. It finds that boundary that maximizes the margin between the classes and the decision boundary. The data points that are closest to the boundary become the so called support vectors and the solution only depends on them. To classify new samples the function

$$y^{(i)} = \text{sign}(w \cdot x^{(i)} + b) \tag{18}$$

whereas the classes have labels $\{-1, 1\}$. The primal formulation of the problem is then

$$L_P = \frac{1}{2}\|w\|^2 - \sum_i \alpha^{(i)} y^{(i)} (w \cdot x^{(i)}) - \sum_i \alpha^{(i)} y^{(i)} b + \sum_i \alpha^{(i)}. \tag{19}$$

$$\sum_i \alpha^{(i)} y^{(i)} = 0 \tag{20}$$

$$\alpha^{(i)} \geq 0 \tag{21}$$

$$0 \leq \alpha^{(i)} \leq C \tag{22}$$

Because the mitochondria data set has shown not to be perfectly linearly seperable (see figure 10) the variable C has been introduced to control the trade-off between slack variables and the margin. The slack variables allow missclassification but a penalty increasing with the distance from the boundary will contribute to the optimization objective.

The linear SVM has been used to classify the mitochondria data set. The value used for C was 100. Because the data is almost linearly seperable it produced good classification results for new samples, too (see table 1). The linear decision boundary computed by SVM is almost identical to the one found using logistic regression (figure 9).

Non-linear SVM Using the dual representation of the SVM classifiation problem it is posible to use kernels to find non-linear decision boundaries in the feature space. The dual presentation is

$$L_D = \sum_i \alpha^{(i)} - \frac{1}{2} \sum_{ij} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)}) \tag{23}$$

under the same constraints from equations 20, 21 and 22 like in the the primal formulation. The kernel $K(x^{(i)}, x^{(j)})$ is substituted to the inner product and transforms the decision problem to a higher dimension in which the data will become linearly seperable. For the mitochondria classification problem a Gaussian kernel (equation 24) has been used.

$$K(a, b) = \exp\left(\frac{\|a - b\|^2}{2\sigma^2}\right) \tag{24}$$

The result for a decision boundary using the Gaussian kernel is shown in figure 11. C has been set to 100 and σ^2 to 0.2. The SVM learned the shape of the data distribution perfectly, but there is also a danger

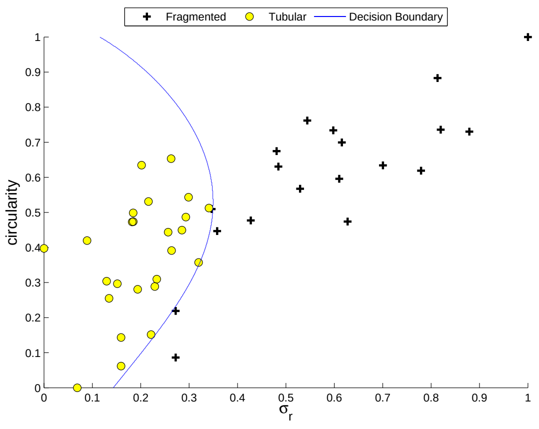

Figure 11. Gaussian Kernel SVM

of overfitting on the possible outliers on the bottom of the figure.

3.4. Neural Networks

Unlike features for other learning algorithms, which were based on circularity of blobs, the pixel values of blobs were used as input for the neural network learning algorithm. As mentioned in section2.2 in on page 4, 20×20px binary images were created for each mitochondrion found in the blob detection phase. The pixel values of the ***thumbnail features*** were converted to a 400 dimensional vector. The vector was then passed as an input to a 3-layered neural network with 10 hidden layer units. Sigmoid function of the edge weights were used as activation function for the nodes of the neural network. The network was trained using 20 iterations of back-propagation learning algorithm, with regularization constant set to 2.

Figure 12 shows how the neural network classifies the mitochondria on a cell marked as *fragmented*. Similar results were obtained on a cell marked as *tubular*.

4. Conclusion

4.1. Classification accuracy

Relative performance of each of the classification method described in section 3 is shown in table 1. The accuracy was calculated by just counting the number of correctly classified images. Because the sample size for both classes were comparable to each other, there was no apparent need to use other accuracy measures like F1 score.

Only one image from an entire stack file could be used

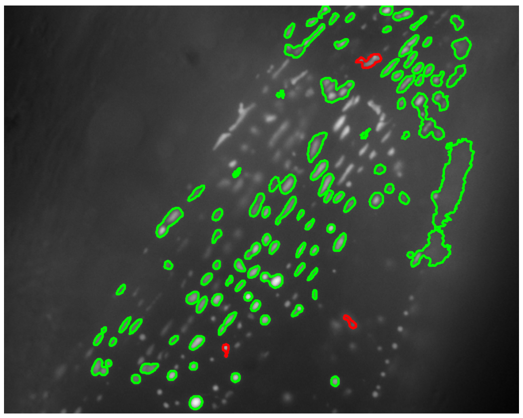

Figure 12. Classification using 3-layered NN; the red and green outlines show the tubular and fragmented mitochondria as classified by the learning algorithm.

Table 1. Classification accuracies for various ML methods.

	FEATURES	ERRCV	ERRTRAIN
MLE	BLOB FEATURES	0.0667	0.0694
SVM	BLOB FEATURES	0.0667	0.0601
SVM KER	BLOB FEATURES	0.0917	0.0598
LOGREG	BLOB FEATURES	0.2250	0.2273
NN	THUMBNAIL	0.2333	0.1172

because the boundary of mitochondria were not distinct. The blob detection algorithm on these images performed poorly because multiple fragmented mitochondria were (wrongly) detected as a single tubular mitochondrion. k-fold cross-validation was used to find the best set of hyper-parameters, with $k = 20$.

As mentioned in section 2, the black and white percentages α and β are quite important for the performance of learning algorithms. Interior point optimization was performed to find the best values for these two parameters. The error to optimize on was how linear seperable the data set becomes while changing the values for α and βunder the inequality constraint $\alpha + \beta \leqslant 1$. The lower the error the more linear seperable the has been. The values of b_l and b_u used for the comparison in table 1 are 0.8438 and 0.1094 respectively.

4.2. Software and Data

The software used for the project is available for free under the url: https://saksham@github.com/saksham/Mitochondria-Classification.git. The repository contains all the code used along with some sample images that are used for functional tests. Code for assignments from Stanford Online Machine Learning class [Ng (2011)] have been adapted, reused and appropriately attributed. Due to the large size of stack files, not the entire sample data has been uploaded. The authors would gladly provide the data to any interested reader upon request.

4.3. Improvements and Future Work

Collecting more images from the worm and re-running the learning algorithms would definitely enhance their performance in terms of accuracy. At the moment of this writing, deconvolving [Cybernetics] and thresholding [Metamorph] images from the microscope is a time consuming process, and the training sample available to comprised of less than 50 images. Although there are upto 50 mitochondria detected in each image, getting more images and enriching the training sample would improve the performance of the learning algorithms. A possible extension of the project would be to investigate whether images from the stack file other than the sharpest image could provide additional spatial information so that edges could be better inferred from the sharpest image.

References

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 3.3

Media Cybernetics. Autodeblur & autovisualize `http://www.mediacy.com/pdfs/AutoDeblur_AutoVisualize.pdf` (accessed 03/15/2012). 4.3

Xinchen Teng Barbara Conradt J. Marie Hardwick Kristin White Eltyeb Abdelwahid, Stephane Rolland. Mitochondrial involvement in cell death of non-mammalian eukaryotes. In *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 2011. 1

Metamorph. Metamorph software `http://www.moleculardevices.com/Products/Software/Meta-Imaging-Series/MetaMorph.html` (accessed 03/15/2012). 4.3

Andrew Ng. Machine learning online lecture `http://ml-class.org` (accessed 03/15/2012), 2011. 4.2

Thomas A. Runkler. *Data Mining*. Teubner, 2010. 2.2