Use of Texture Classification To Quantify Stem Cell Differentiation During Time Lapse Imaging

Sunil Pai, Nathan Loewke, Thomas M. Baer
Department of Applied Physics

1 Introduction

One of the major roadblocks in understanding the efficacy of stem cell therapy is the prediction of stem cell performance during treatment or "quality assurance" of stem cell therapy. There have been some documented cases (notably one a few years ago in Europe) of stem cell treatments that have led to tumors, which are caused by stem cells that do not develop properly during treatment. Our lab is focusing on introducing novel imaging approaches to improve our current understanding of stem cell behavior. In one approach, we are tracking stem cells and analyzing stem cell morphologies to understand how cells socially interact. In another approach, and as this paper addresses, we are trying to classify cells as they mature from the stem cell stage to mature cell stage based on their morphology and texture. Together, these two approaches will give us a better picture of not only how stem cells differentiate as a function of time and morphology, but also how we can bring more practical software solutions into the stem cell clinic.

In our experiments, we human embryonic stem cells onto a plastic dish for imaging over a period of about a week using a Nikon Biostation. This microscope gives relatively high-resolution phase contrast images of cell proliferation. The goal of the experiments is to monitor stem cell growth and migration patterns over several days as they begin to differentiate or mature into various different cell types.

This allows us to arrive at several different revelations about stem cell behavior: we can quantify how much of the cell plate's area changes from embryonic stem cells to mature, fully developed cells. We can also monitor cell death rates (as it turns out, dead cells have a unique texture as well) as well as cell division rates (again, there is a distinct shape cells make when they divide which can be correlated to a texture), both of which can inform us of a stem cell colony's propensity to be useful for treatment. All of these events and cell types have distinct textures that can be identified by texture classification techniques on phase contrast images. Whereas phase contrast image analysis is one of the most difficult tasks in biological computation studies today, this project presents a straightforward analysis of stem cell images that may have the potential to tackle the one of the most pressing challenges in stem cell therapy.

2 Texture Feature Extraction From Images

2.1 Haralick Features

2.1.1 Gray-Level Cooccurrence Matrix

One common texture classification technique is Haralick feature extraction, which uses gray-level co-occurrence matrices (or GLCMs), which give an indication of how uniform or nonuniform a given texture is. While gray-level co-occurrence matrices are rotation invariant, statistical analysis performed on these GCLMs (developed by Haralick) provide useful information about how grayscale values in the image matrix vary in a rotation invariant manner. This is done by calculating the GLCM over the four possible orientations for the four pixel neighbor orientations.

2.1.2 Haralick Statistics

The idea of Haralick texture classification is that these statistics give a general idea of the pattern of neighboring pixels. Since GLCMs give information about the relations of neighboring pixels to each other, they are a very good measure of the variational patterns of image pixel values in an image. There are several different statistics that are used on each GLCM to formalize these so called "patterns." Features calculated include entropy, correlation, variance, contrast, mean, among others over the 2-dimensional probability density given by the GLCMs (given by normalizing the matrix elements by the size of the image). To reduce dimensionality and induce rotation invariance, these quantities were averaged over the four possible GLCMs for the typical Moore neighborhood pixel representation.

2.2 Gabor Features

The Gabor wavelets are constructed using the Gabor filter method, which is a bandpass filter applied to the image at different orientations and scales, similar to a mechanism used by the human visual system. I used the starter code from the author of [1,2] and use of these features was inspired by a patent on phase contrast texture classification [3].

2.2.1 Radial Component

The first step in calculating the Gabor features is to use a low pass filter in order to ensure that the normalization of the Gabor filters can be performed. Since low pass filters use the fuction $\text{sinc}x = \frac{\sin x}{s}$, multiplying this to the Gabor filters will allow the values of the Gabor filters to fall off at the corners of the image as required. Then, I apply the Gabor filter given by the following Gaussian radial formula evaluated at the scale s(and yielding a matrix the same size as the training example):

$$G_s = \exp\left(-\frac{(\log R\lambda_s)^2}{2\sigma_R^2}\right) F \tag{1}$$

Note that the part of the Gabor filter here that depends on the scale s is the wavelength which exponentially depends on s as: $\lambda_s = \lambda_0 m^{s-1}$, where λ_0 and m are pre-defined parameters. Additionally, R is the radius matrix, discretized based on pixel distances from the center of the image, and σ_R^2 is just the variance parameter. Finally, a low pass filter F is multiplied to this matrix to get the final "log Gabor" filter.

2.2.2 Angular Component

The angular component also requires the calculation of a matrix Θ, which accounts for the orientations of all of the pixels in the image relative to the center. Then, based on an orientation, another Gaussian wave packed is calculated as follows:

$$\Theta_o' = \left|\arctan\frac{\sin\Theta\cos\theta_o - \cos\Theta\sin\theta_o}{\sin\Theta\sin\theta_o + \cos\Theta\cos\theta_o}\right| \tag{2}$$

$$G_o = \exp\left(-\frac{\Theta_o'^2}{2\sigma_\Theta^2}\right) \tag{3}$$

2.2.3 Final Construction of Gabor Features

The final construction of the Gabor features is given by the mean energy and the mean amplitude at all of the various scales and orientatinos s and o. We construct this by convolving the image matrix $\mathcal{I}$ with the inverse Fourier transform of the filter: $A^{so} = \mathcal{I} * \mathcal{F}^{-1}(G_s G_o)$. From this, take the absolute value of all of the elements of A^{so}, which I use to find the energy and the mean amplitude for an $n \times n$ image:

$$E_{so} = \sum_{i,j}^{n} |A_{ij}^{so}|^2 \tag{4}$$

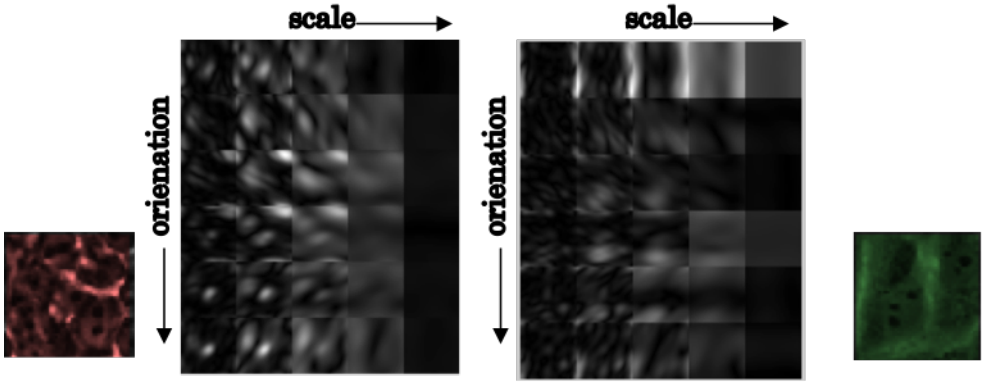

Figure 1: Gabor wavelet (A^{so}) banks for indices $s = 1, 2 \ldots 5$ and $o = 1, 2, \ldots 6$, giving a 6×5 bank.(left) Stem cell colony Gabor filter bank, (right) Mature cell Gabor filter bank. Notice the effect of orientation on white streaks and the relationship between scale and texture of the images

$$MA_{so} = \frac{1}{n^2} \sum_{i,j}^{n} |A_{ij}^{so}| \tag{5}$$

Since we are calculating features from 6 different orientations and 5 different scales, these values are just stacked into a single column vector to give a total of 60 values, 30 for the energy and another 30 for the mean amplitude. See Figure 1 for a picture of Gabor wavelet bands created from this method.

Feature Selection

Since we have 80 features (60 features from Gabor wavelets and 20 features from Haralick statistics), it is important to avoid using features that actually confound the data (resulting in possibly more misclassifications), so we decided to use a feature selection algorithm to help accomplish this. We chose the genetic algorithm (implemented using MATLAB's Optimization Toolbox) to act as an unsupervised heuristic search over the space of possible feature combinations. The population used was a random population, but we found that the genetic algorithm usually converged at a set of around half of the total features, with an improvement in cross validation of about 1% (less than what we were expecting, but it certainly helps with intuition to know which set of features provide the best separation). See Figure 2 for results. These bit vector values can be stored and used to select

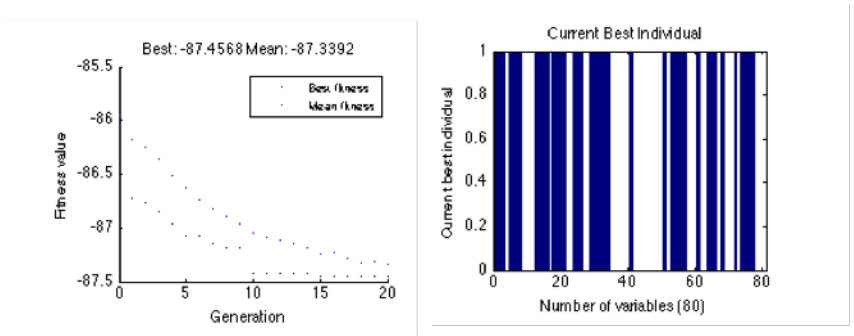

Figure 2: (left) Genetic algorithm learning curve on cross validations for linear kernel SVM with different feature sets; (right) Optimal bit vector selection for the feature set, where the first 20 features are Haralick features and the final 60 features are Gabor wavelet features.

features from the training set with which to use in the SVM model. The conclusion we can make from the feature selection is that most of the Haralick features do a fairly good job of separating the data and the mean amplitudes and mean energies are better predictors at small and large scales (values of s) respectively. Subsequently, we achieve our final classification accuracies described in the following section.

3 Comparison of Machine Learning Models in Texture Classification

Setting Up The Classification

Our ultimate goal is to classify our image into textures for stem cell colonies, mature cells, background, mitotic cells, and dead cells. The method we chose to do this was to develop a training GUI in MATLAB (one that any biologist can use) to classify regions of the image. From these regions, we pick 64 by 64 pixel windows which are each about the size of an average cell, and calculate features for each window. This feature calculation was demonstrated pictorially in Figure 1. For our tests, we interpolate through fixed pixel intervals (in this case, 16 pixels up and down) and look at the 64 by 64 window around each interpolated pixel to perform the classification for that region of the image. Note that the comparison shown in Figure 3 only provides a rough estimate of the testing accuracy because at present, we have not yet stained cells to determine whether a cell is mature or is a stem cell. We do know, however, that the cells shown in green in Figure 3 are indeed *more* mature, or differentiated, than the stem cells because they form a different colony structure than the stem cells do, but we are currently developing a method for experimentally supporting this.

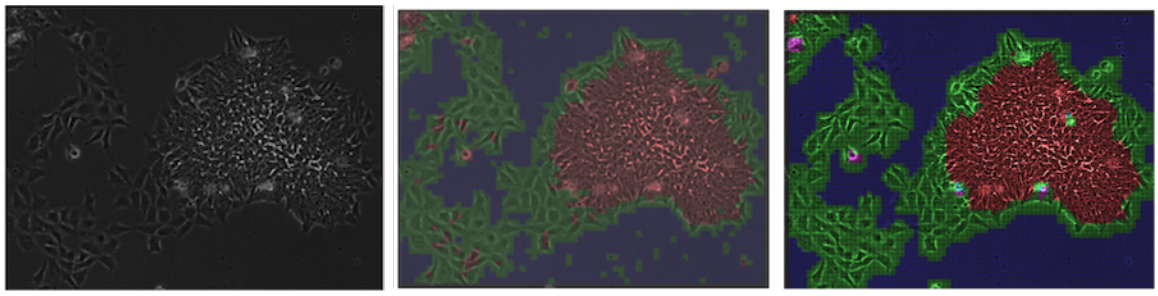

Figure 3: (left) Original; (center) LDA and Haralick (3 classes: background, flat/differentiating cell, colony/stem cell); (right) SVM and Haralick/Gabor Energies(5 classes: background, flat/differentiating cell, colony/stem cell, dividing cell,dead cell)

Comparing C-SVM and LDA Performance

To get a sense for how well our training data is separated in the space of only 2 features, we decided to use linear discriminant analysis (LDA), and chose the an optimal feature class within the Haralick feature space. Notice that there is still some noise using just the LDA algorithm to perform classification in the test image in Figure 3, but in Figure 4, one can already see that a simple application of LDA should provide quite nice separation of the data. We used our genetic algorithm feature selection to choose the best feature class for the one-versus-one multi-class support vector machine classifier (SVM). We subsequently ran our algorithm on several different kernels and the linear kernel provided the best cross-validation accuracy and the best testing accuracy (significantly less noise-prone performance than LDA with three features). These comparative results are shown in Figure 4. Note the dependence of features 1 and 2 in the graph; it is certainly not linear, and we suspect that this relation holds for other features in the data set, a reason we opted *not* to use PCA to describe our feature space in the end. We also suspect that the linear kernel works the best for SVM because it provides the *least* variance (a 3rd-degree polynomial, for example, would overfit the data resulting in poor classification) and relatively little bias since we have such good separation from our features anyway.

4 Event Curves

We extended our one-versus-one SVM model (which we picked due to having the least noise and high cross validation accuracy) to around 600 images (similar to that shown in Figure 3) of embryonic stem cells slowly differentiating into mature cell lines. We generated a movie of these textures, which we had put on display at the poster session, to determine some practical uses for our classifier. An *event curve* is a concept we hypothesize could be used as a time-series predictor for stem cell

therapy efficacy. The curves are the percentage of the field of view occupied by a certain classification as a function of time. We show a preliminary event curve for the simple 3-class classification in Figure 4. The interpretation of this plot is that stem cell prevalence continues to grow and eventually dominates the field of view, while the mature cell prevalence tends to "peak" at some point. We are currently working on optimizing our more advanced 5-class SVM model, after which we will plot a similar event curve with mitosis events and cell death events. We hope that these event curves can describe quantitatively the growth of stem cells and mature cells *in vitro* and can be used as a tool for identifying useful cells for therapy.

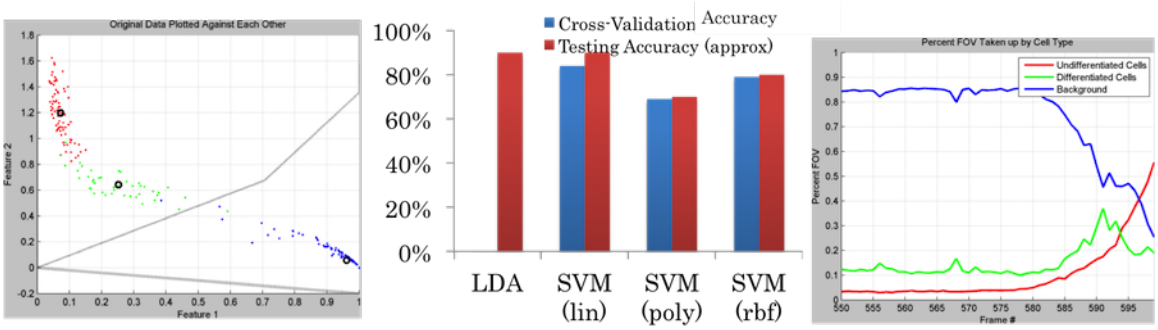

Figure 4: (left) LDA on space of two features; (center) comparison of testing accuracy between LDA and SVM and comparison of cross validation/testing accuracy between different SVM kernels, showing that the linear kernel gives best model.(right) Event curve set for 3-class SVM model

5 Conclusion

Texture classification is quite effective in classifying embryonic stem cells and mature human cells, giving a cross validation accuracy of close to 90 % and showing promising time-lapse results. This high accuracy allows us to describe a novel concept called an event curve which describes the frequency of a particular event in a given time lapse movie within a field of view as a function of time. By using only phase contrast images, we do not need explicit fluorescent screening or cell sorting, which may interfere significantly with cell function. Rather we could in the future use time lapse movies to make key decisions about whether a colony is fit for therapy or not, saving significant effort and time. This is a novel idea which may have major implications in future stem cell therapy approaches.

Our group plans to expand on the notion of a biological event curve and derive a very basic quantitative measure for therapeutic potential of a given colony based solely on phase contrast image texture analysis. We will continue to explore this problem by correlating our event curves with cell fluorescence data. This is an extremely powerful idea which, if proven to be a quality assurance measure, can be taken directly to the clinic. We are very excited about the prospect of continuing this vein of research. I'd like to acknowledge my colleague Nathan Loewke, who works in the EE department for writing the LDA code on Haralick features and helping me write the MATLAB GUI.

6 References

[1] Peter Kovesi, "Symmetry and Asymmetry From Local Phase" *Tenth Australian Joint Conference on Artificial Intelligence*. 2 - 4 December 1997. http://www.cs.uwa.edu.au/pub/robvis/papers/pk/ai97.ps.gz.

[2] Peter Kovesi, "Image Features From Phase Congruency". Videre: *A Journal of Computer Vision Research*. MIT Press. Volume 1, Number 3, Summer 1999 http://mitpress.mit.edu/e-journals/Videre/001/v13.html

[3] Sammak et al. "Image-based methods for measuring global nuclear patterns as epigenetic markers of cell differentiation." The Charles Stark Draper Laboratory, Inc. Mar 15, 2011.