Project Module 3 : Prediction modelling

Bahnisikha DUTTA, Mathieu LAPEYRE, Palash SHASTRI

Georgia Institute of Technology, ECE 6780, Group 9

Atlanta, GA

Keywords: Classifiers; Cross validation; Random Forest; SMO; Performance Metrics

I. CLINICAL NEED

Labeling a histopathology image as having cancerous regions or not is a critical task in cancer diagnosis [21]. Existing image classification techniques require detailed manual annotations for the cancer pixels, which are time-consuming to obtain. With the dramatic improvements in computational power, computer-assisted diagnosis has been made easier. Today, pathologists need the support of a clinical decision system for diagnosis and prognosis of cancer.

II. PROBLEM STATEMENT

Now that we have segmented image and features available for a given image, we have to develop and validate computer-based prediction models to help designing a decision support system for clinical diagnosis. As some classifiers may perform better than other for some specific tasks, we will need to evaluate our models according to different performance metrics and compare our models to see which one fit the best for every specific cancer.

First, we will briefly review the methods and techniques we saw on module 1 and module 2. Then we will conduct a literature survey on computer-based classifiers for image analysis. We will emphasize three of those classifiers and analyze their performance on the 3 different dataset we have.

III. LITERATURE CRITIQUE

A. Literature critique for image segmentation and feature extraction

Below is a brief review of our work for module 1 and module 2. We will then focus on the review of classifier. The methods we have chosen for our project are **highlighted in yellow**.

Subject	Strategies	Papers and Location	Strength and Weakness
Normalization	• Reinhard's Method [11] • CDV-L [12] • CDV-MM [12] • Colormap Normalization [4]	• Reinhard et al, IEEE Computer Graphics and Vision, 2011 • Magee et al, MICCAI Workshop, 2009 • Kothari et al, Journal of the American Medical Informatics Association. 2013	• First two approaches introduce high background color distortion. They are highly dependent on reference batch selection. • They are all easy to implement. • Our method: statistical colormap does not require manual selection of reference image / reference batch.
Supervised Segmentation	• LDA classification and reclassification in sample color space [16] • Supervised grayscale conversion, learning based marker detection and classification, watershed segmentation [20] • Bayesian, KNN and SVM training, color watershed [18]	• R.A. Fisher, Annual Eugenics, 1938 • K. Mao et al, IEEE Transactions on Biomedical Engineering, 2006 • M. Datar et al, Biomedical Imaging: From Nano to Macro, 2008	• Supervised classifiers are highly dependent to ground truth selection • Our method: LDA classification followed by Level Set method is fast and works well for small reference dataset • Training classifiers takes time • In general, supervised classifiers produce better results than unsupervised, if well used.
Unsupervised Segmentation	• Hierarchical Self-Organizing Map [18] • EM algorithm with the Fisher-Rao criterion as its kernel [16] • K-Means clustering in the L*a*b color space • Kernel Graph Cut [8]	• Datar et al, Biomedical Imaging: From Nano to Macro, 2008 • R.A. Fisher, Annual Eugenics, 1938 • • Salah et al, IEEE Transactions on Image Processing, 2011	• Heavily relies on strong assumptions on data distribution, e.g. Gaussian • For all methods, we suffer from the lack of in-depth prior knowledge for the segmentation task • Our method: K-means followed by Kernel Graph Cut method. K-means generates good initial segmentation and is fast. KGC performs better than traditional Graph Cut
Performance Evaluation	• Foreground/Background variance [15] • Rand Index [9] • Squared Color Error [14] • Entropy based Evaluation [13] • Intra/Inter index (or Zeb) [13]	• Weszka et al, IEEE Transactions on Systems, Man and Cybernetics, 1978 • Unnikrishnan, IEEE Workshop on Applications of Computer Vision, 2005 • Liu et al, IEEE Transactions on Pattern Analysis and Machine	• Foreground/Background relies on strong assumption on the data distribution. Can lead to negative values • Supervised cluster similarity evaluation with Rand Index • Compute error on the overall region instead of region boundary • • Unsupervised per-region evaluation, weighted by boundary length and region size

		Intelligence, 1994 • Zhang et al, Proceedings of SPIE- Storage and Retrieval Methods and Applications for Multimedia, 2004 • /Same/	

Table 1: Module 1

Subject	Strategies	Papers and Location	Strength and Weakness
Color features extraction	• Histogram analysis and Gleason Grading [31]	• IEEE Transaction on Medical Imaging, 2007	• While most methods extract color-based features from raw color histograms, this method preprocess the image by removing the white background pixels so as to remove any bias.
Shape features extraction	• Object- and Spatial-level Quantitative Analysis [24]	• Boucheron et al, PhD thesis, University of California, Santa Barbara, 2008	• Comprehensive review of feature extraction techniques for histopathological images. Lack of information on how to combine the statistics for a tiled image. • Our method: Estimate a distribution amongst the tiles using the statistics described in [24]
Texture features extraction	• Textural Features for Image Classification [22] • Variation in illuminant direction on texture classification [23]	• Haralick et al. IEEE Transaction on System, Man and Cybernetic, 1973 • Chantler, Department of Computing and Electrical Engineering Heriot-Watt University, 1994	• Texture features can be model-based (fractals, markov random fields, …) or non model-based (GLCM, Gabor filters, …) • The author states that non model-based texture features produces better result than model-based ones. We disagree because it highly depend on the input image. • Our method: GLCM is reportedly better than other approaches on several general applications. It is fast and provides more than 14 easily computable features.
Feature ranking	• Statistical Dependency • Sequential Forward Selection • Sequential Backward Selection • Mutual Information • Minimal-Redundancy and Maximal-Relevance • Random Subset Feature Selection	• Pohjalainen et al, Computer Speech & Language, 2015	• There are various ways to combine feature ranking algorithms. • Some algorithms can be computationally intensive. • Combining Statistical Dependency and Mutual Information is reportedly a good approach to select top ranked features
Dimensionality Reduction	• LLE [26] • t-SNE [25] • LDA [27] • PCA [27]	• Van der Maaten et al, Journal of Machine Learning Research, 2008 • Martonez et al, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001 • /same/	• Linear techniques cannot adequately handle complex nonlinear data • t-Distributed Stochastic Neighbor Embedding is a recent non-linear technique that is particularly well suited for the visualization of high-dimensional datasets • PCA performs better than LDA when dealing with small training datasets • It is interesting to compare linear and nonlinear algorithms performance

Table 2: Module 2

B. Literature critique for computer-based classification

After module 1 and module 2, we have a few top ranked features that we can use to train our classification model. We distinguish linear from nonlinear classifiers. A classifier is linear if its

decision boundary on the feature space is a linear function: positive and negative examples are separated by an hyperplane. In other terms, a linear classifier decides class membership by comparing a linear combination of the features to a threshold.

Subject	Strategies	Papers and Location	Strength and Weakness
Reviews	• General Guidelines	• Belazzi et al, Int J. Med Inform, 2008	• Very elaborate paper that discusses problems and issues we can face when dealing with classification
Linear Classification	• Fisher's Discriminant Analysis [36] • Support Vector Machine [37] • Naive Bayes [43]	• Mike et al, IEEE Signal Processing Society Workshop., 1999 • Chang and Lin, Intelligent Systems and Technology, 2011 • Rish, American Association for Artificial Intelligence, 2001	• FDA is very popular and usually used on linear classification problem. The main weakness is that it does not work well for overlapping data distributions • SVM are very popular and can handle nonlinear data as well. It produces satisfying results for a very small computation overhead. However, a single model struggles with mul-class classification problem and can become very complex and expensive • Bayes classifier is simple and with strong (naive) independence assumptions between the features.
Nonlinear Classification	• k Nearest Neighbor [34] • Neural Networks [38] • Random Forest [44]	• Perreira, Neuroimage, 2009. • Jain et al, IEEE computer vol 29, 1996 • Breiman, University of California, Berkeley, 2001	• KNN is one of the simplest classifier. It is very fast to implement with few parameters to tune. It is mainly used for well separable clusters. However, this classifier is highly dependent on the choice of neighbors and the data distribution. It does not work well for overlapping distributions. • Neural Network is effective with high dimensional features but it is really hard to tune, time consuming and it suffers from overfitting • Random Forest is considered as the state-of-the-art decision model. It is built on several decision trees. For each decision tree, a random sampling is done which lower the variance of the overall model. However, the runtime complexity increases with the increase in number of trees and candidate features
Performance Metrics	• Reviews and General Guidelines [42]	• Powers et al., Journal of Machine Learning, 2011	• Classification through accuracy alone cannot be trusted to select a well performing model. This is the Accuracy paradox • Matthews coefficient is regarded as being the best coefficient to summarize the confusion matrix in one number. • F-Score is a good metric to see balance between precision (exactness) and recall (completeness)

Table 3: Module 3

More insights about the 4 classifiers we chose can be found in the next sections.

IV. OUR APPROACH

From image segmentation to classification, our approach to the problem at hand can be described by the following flow chart:

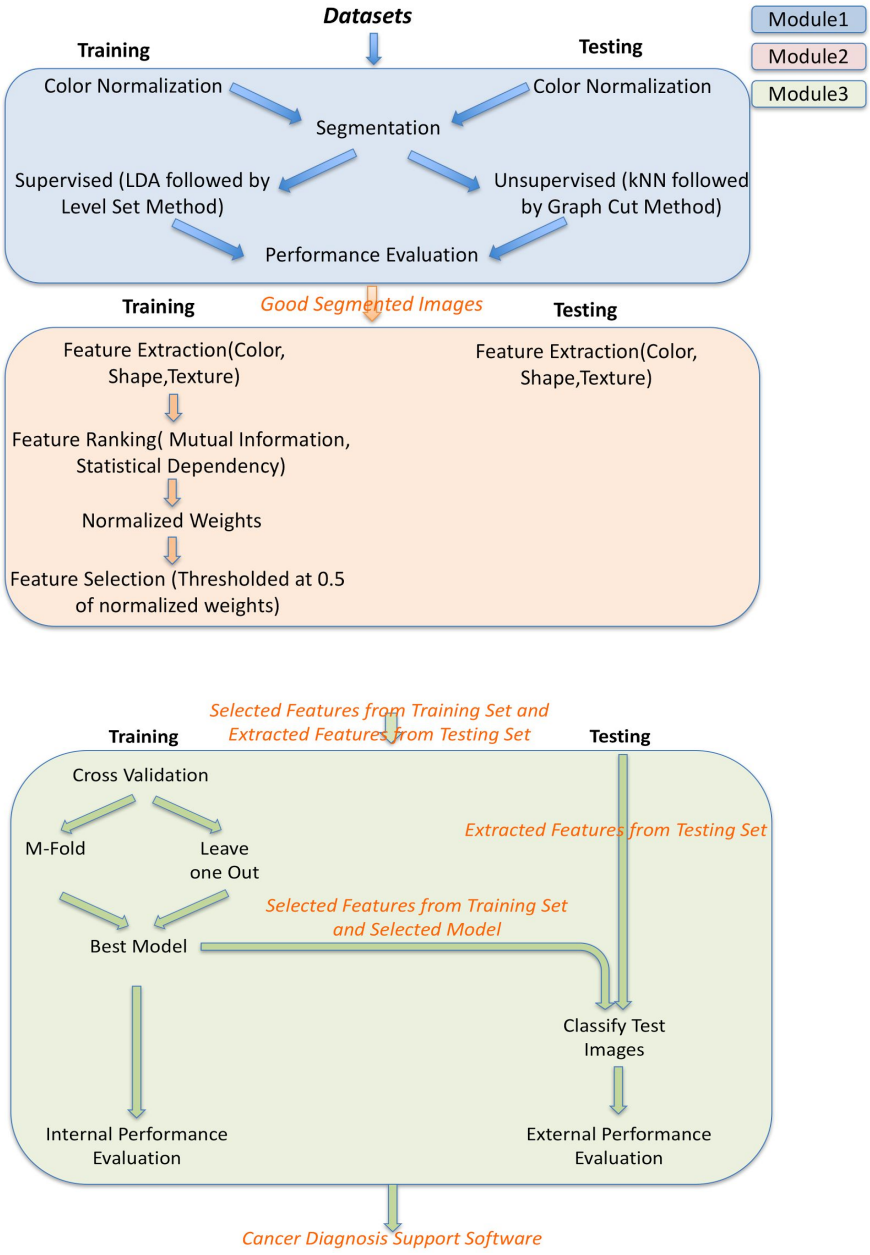


Figure 1: System design flowchart

All the steps from module 3 are explained at length in the following sections. For more details on module 1 and module 2, please refer to the corresponding reports.

A. Classifier training

Classifiers used information to be provided below-

1. KNN

K-Nearest Neighbors is one of the most commonly used machine learning algorithms. Simple and intuitive to understand, it employs the distance metric (euclidean distance standard) and voting based scheme where the given test sample is assigned the label represented by most of its neighbors. Takes the number of neighbors, an important hyperparameter, as input. KNN's are mainly used after feature reduction in a lower dimensional manifold space for better classification results.

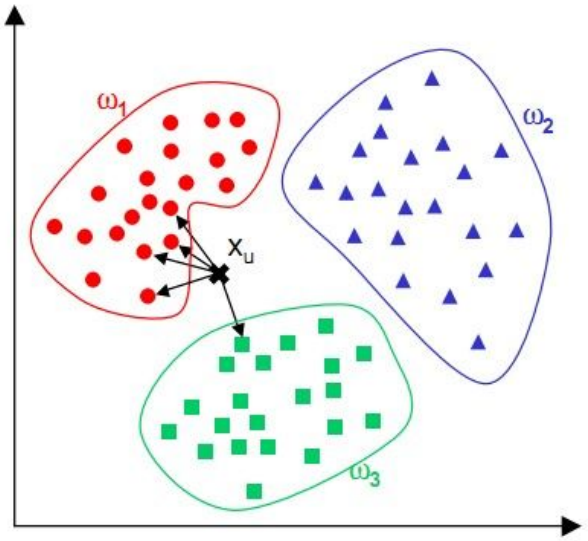


Figure 2- Visualization of kNN for three clusters

This worked really well for Dataset 1, however as was quite evident from the feature plots of Dataset 2 and 3 (Module 2); the non-linearity of these datasets demanded a non- linear classification technique, some of which we tried and are listed as below.

2. Neural Networks

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) which are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. Now, perhaps it is not so complicated to adjust the weights of such a small network, but also the capabilities of this are quite limited. If we need a network of hundreds of neurons, how would you adjust the weights to obtain the desired output? There are methods for finding them, and now we will expose the most common one. The backpropagation algorithm (Rumelhart and McClelland, 1986) is used in layered feed-forward ANNs. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers. The backpropagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of

the backpropagation algorithm is to reduce this error, until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal.

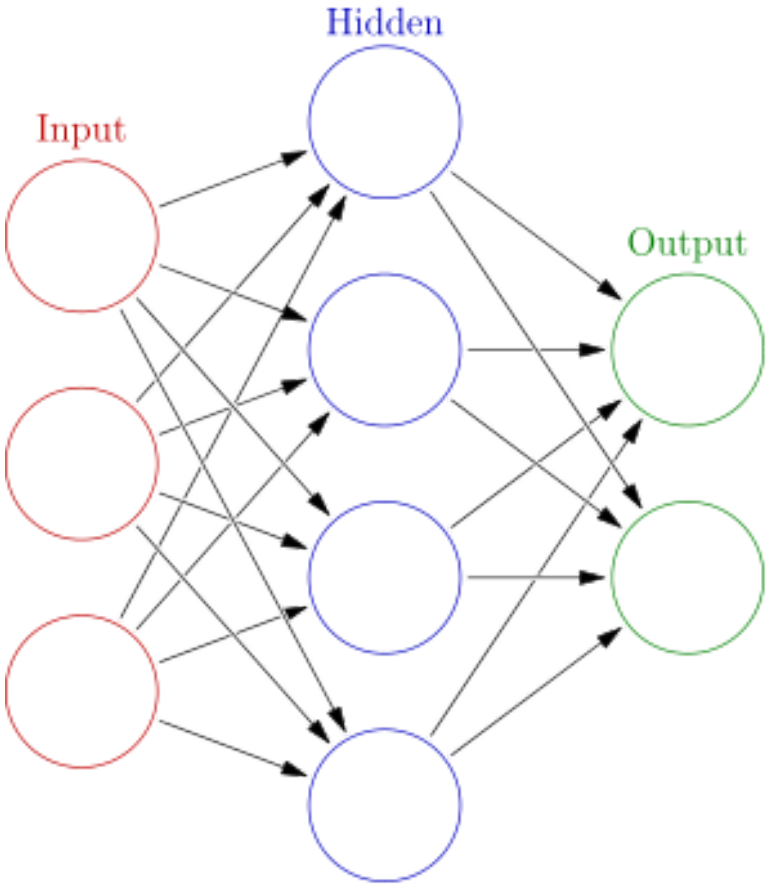


Figure 3: Artificial Neural Network

For our Training Purposes, we tried different numbers of layers and neurons and used some cross validation methods on our training set to come up with a model which gave us the best efficiency on the training set. For practical reasons, ANNs implementing the backpropagation algorithm do not have too many layers, since the time for training the networks grows exponentially. Time efficiency and Performance Accuracy forced us to look and implement other non linear classification techniques for Datasets 2 and 3.

3. Support Vector Machines

Support Vector Machines are kernel based machine learning algorithms that essentially leverage the fact that non-linear data in low-dimensional space can be separated linearly separable if projected into higher-dimensional space. Primary objective function is to find the optimal hyperplane that maximizes class separability and lowers misclassification error. It's performance on a dataset is governed by the choice of kernel.

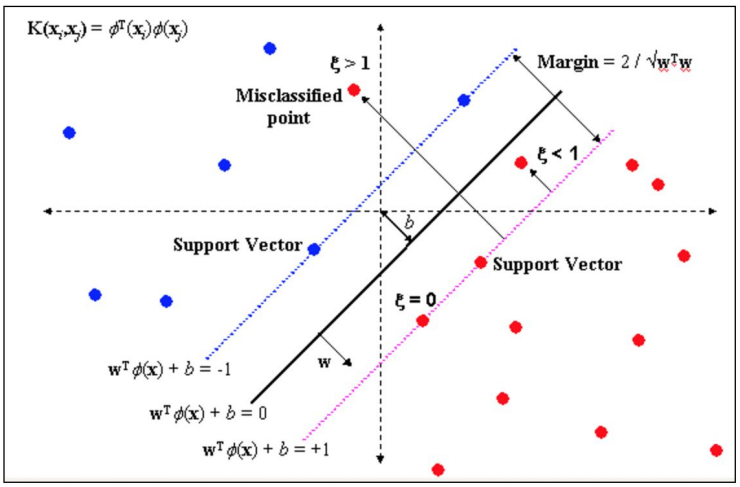


Figure 4: Support Vector Machine visualization with maximum separation hyperplane

The objective function can be formulated as follows-

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i \qquad \text{s.t.} \qquad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0 \quad \forall_i$$

4. Random Forests

Decision trees also known as random forests are an ensemble based learning mechanism wherein a group of weak classifiers are combined to form a strong classifier, leading to reduction in variances. Random forests work on the fundamentals of bagging and random selection of features. Multiple decision trees, learnt on the basis of information gain, are trained on randomly sampled features in an

attempt to approximate the underlying data distribution. Major advantages of using this family of classifiers is:

1. Very efficient to train
2. Inherent randomness reduces the underlying variance amongst weak learners

Provide the formula of information gain (format in word)

$$I(T) = \sum_i -P(\ell_i) \log_2 P(\ell_i)$$

$$I(D,T) = \frac{|T_p|}{|T|} I(T_p) + \frac{|T_n|}{|T|} I(T_n)$$

Above formulas show the mathematical interpretation of information theory . Equation 1 shows the information content at each node, where P(li) is the probability of input labels. Equation 2 represents information content after partition of the data subset based on a decision function. Information gain is then defined as-

Information Gain (IG) = I(T) - I(D,T)

Graph visualization of tree from MATLAB

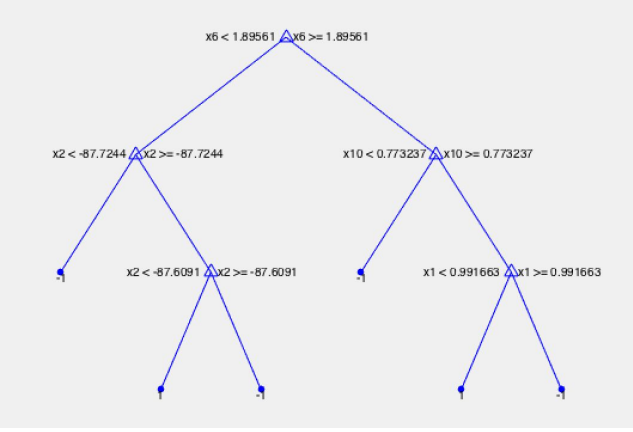


Figure 5: Tree visualization using MATLAB

B. Cross validation

Cross validation is one of the key important factors in determining the generalizability of a model by tuning the parameters of the classifier on various partitions of training and test-data, specifically declared through the 'k' variable. Two important cross-validation techniques used in our project include-

- K-Fold Cross Validation Approach

In K-fold cross validation approach, the data is divided into k folds with k-1 folds as training sets and the kth fold as testing set.

- Leave One Out Cross Validation Approach

Leave one out cross validation approach is another method of implementing cross-validation where-in each sample is taken as a test sample and the remaining sample act as the training set. Essentially it is a special case of cross validation where you have same number of models as samples. Computational complexity increases immensely with the increase in number of training points.

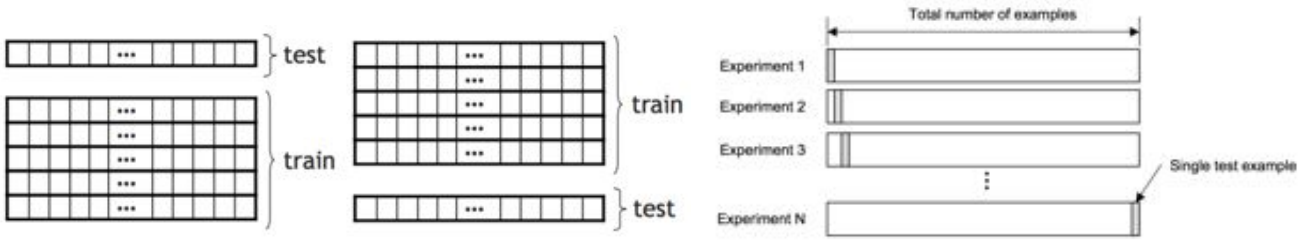


Figure 6: Visualization of k-fold and leave one out cross validation approaches

C. Performance metrics

After building a classification model, the first thing which comes to mind is verifying the accuracy of that model and checking the number of correct predictions made over the total number of

predictions. Although this approach is fast and easy, it may be not give enough relevant information for the problem we are trying to solve. This is the accuracy paradox.

We first used the confusion matrix as a clean and unambiguous way to present the prediction results of our classifiers for every classes. From that, we were able to compute several measures which, alltogether, can give enough insights on the performance of a classifier.

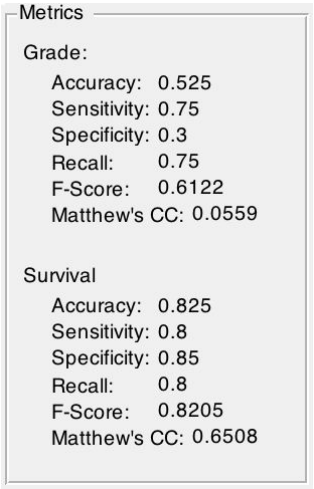


Figure 7: Example of the metrics used to analyse the performance of a classifier (taken from our GUI). The image was correctly label for both grade and accuracy using Random Forest. However, we can not rely on this model for the grade classification. The accuracy is really bad and the Matthew's coefficient shows that the classification is considered as random (near 0)

Performance metrics used are:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Population} \qquad Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad F\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad Matthew's\ Correlation\ Coeff. = \frac{TP \cdot TN + FP \cdot FN}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}}$$

V. RESULTS AND PERFORMANCE

Before delving into the result and performance of our classifiers, it is important that we re-iterate some important observations from our feature extraction step in module 3. We extracted a total of 1728 features which are tabulated in the appendix. After extraction, we used two different feature ranking methods; Statistical Dependency(SD) and Mutual Information(MI). Then we used two different methods to select the top features from the two ranked feature arrays which is explained below:

Feature Selection from SD ranked feature array:

The following graph plots the normalized weights obtained for each feature in an ascending fashion. The weights are then thresholded at 0.5 as shown by the red line and features above the threshold are selected.

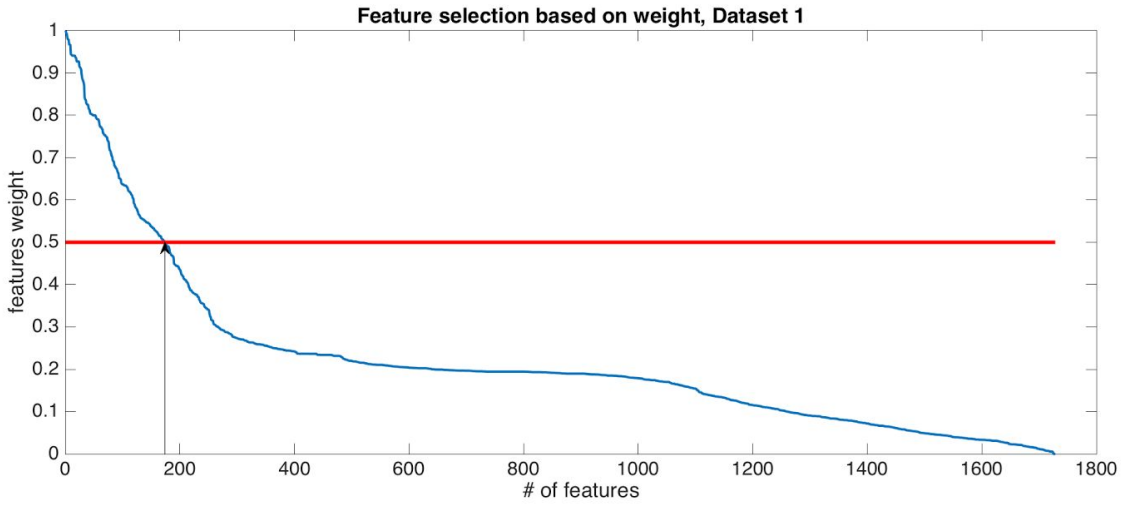


Figure 8: Plot of weights vs # of features, arrow indicates the cut-off point beyond which no features are taken into account

Feature Selection from MI ranked feature array:

Mutual Information feature selection algorithm is the second feature selection algorithm we use for selecting the most relevant features that could aid in better classification results. Below is a accuracy vs

number of iterations plot obtained using a KNN classifier where only the training set is used. For us top-100 features gave best results.

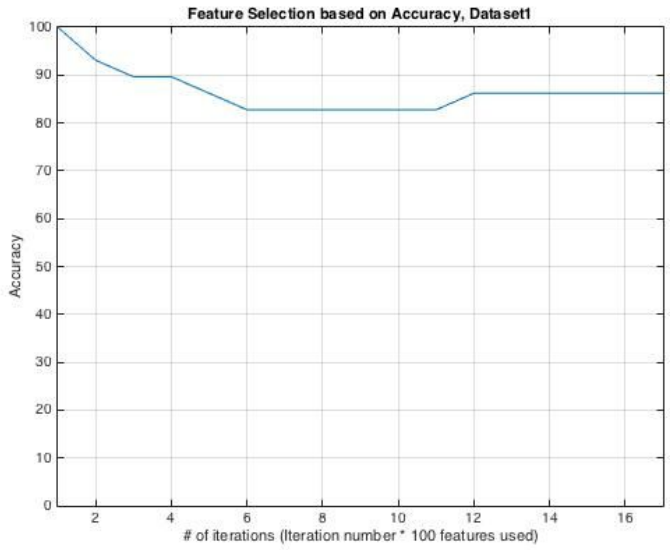


Figure 9: Plot of accuracy vs # of iterations

Here is a list of top 5 features selected by both the methods:

TOP5	D1(SD)	D2(SD)	D3(SD)	D1(MI)	D2(MI)	D3(MI)
1	Orientation	Perimeter	Eccentricity	Convex Area	Perimeter (Median)	Eccentricity
2	Extent	Minor axis	Orientation	Major Axis	Perimeter (Standard Deviation)	Value Channel
3	Area	Major axis	Contrast	Minor Axis	Minor Axis	Extent
4	Perimeter	Green channel	Dissimilarity	Perimeter	Saturation Channel	Major Axis
5	Eccentricity	Area	Area	Area	Hue Channel	Perimeter

Table 4: The list of top features

As we can see from the above table, most of the selected features are shape based. While visualizing the selected features through PCA and tSNE, we saw the best separation among classes for MI followed by tSNE.

We carried forward the selected features for our classifier training step and trained 4 different classifiers: kNN (Dataset1 only), SVM, Random Forest, and Neural Networks. After internal validation, the following parameters were freezed for the aforesaid classifiers:

	kNN	SVM	Random Forest	Neural Network
Parameter 1	Neighbors- 5	Kernel- Radial Basis Function	No of Trees- 200	No. of Hidden layer=12
Parameter 2	Weighted	K- 5 fold cv	Method- Bootstrapping	Type=Patternnet

Table 5: Parameters for the different models

Here are some sample sample performance metrics for some of our best performing models for the three datasets.

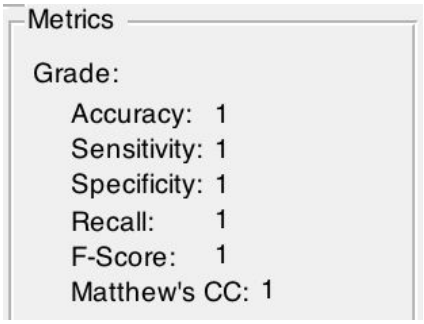
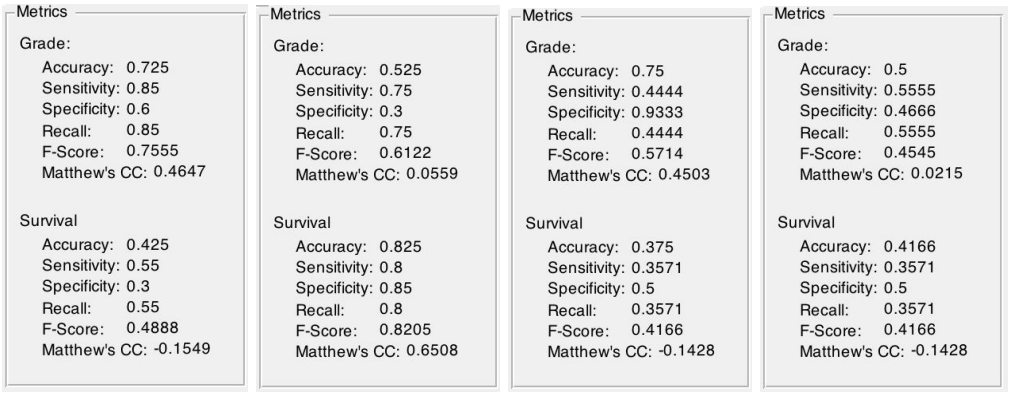


Figure 10: Dataset 1 using Random Forest



a b c d

Fig11:a)Dataset 2 using SVM b)Dataset 2 using Random Forest c)Dataset 3 using SVM d)Dataset 3 using Random Forest

At this point, it is interesting to see how the performance varies if we used the lower ranked features (last 100) instead to train our model. Here is one such example:

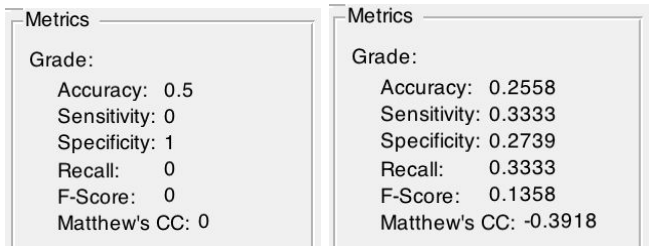


Fig12: Dataset 2 using SVM (for grade), Dataset 1 using Random Forests

These metrics indeed show the importance of a feature selection algorithm and the effectiveness of our implemented methodology as the performance greatly reduces with selection of lower ranked features.

Below you will find some example of how the Random Forest and SVM classifiers behaves on some Dataset 2 images.

Positive Grade:	G3 and G4 = 1
Positive Survival:	Low Survival = -1

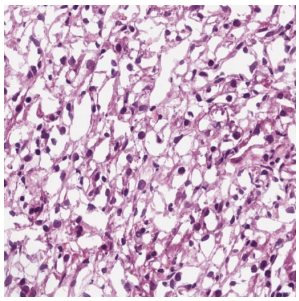
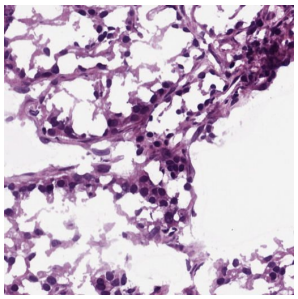
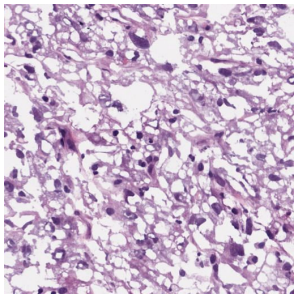
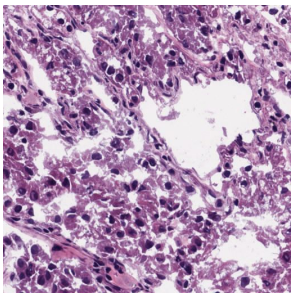
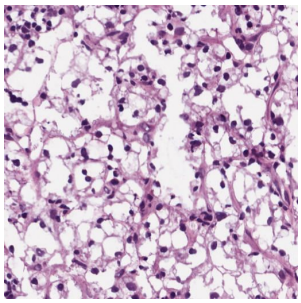
	 TCGA-B0-5104	 TCGA-B0-4813  TCGA-CZ-5468	 TCGA-BP-4337	 TCGA-CJ-4874
True labels	Grade: G2 (-1) Survival: 2751 (1)	Grade: G3 (1) Survival: 18/59 (-1)	Grade: G4 (1) Survival: 2 (-1)	Grade: G3 (1) Survival: 2283 (1)
Random Forest	Grade: Predicted FP Survival: Predicted FP	Grade: Predicted TP Survival: Predicted TP		Grade: Predicted TP Survival: Predicted TN
SVM			Grade: Predicted TP Survival: Predicted FN	Grade: Predicted TP Survival: Predicted TN

Table 5: Some images and related results

Random Forest produced one complete FullPositive (FP for grade classification and FP for survival classification in the same time) for patient TCGA-B0-5104. One good thing is that it did not produced full FalseNegative (FN for grade classification and FN for survival classification in the same time) which is great! Indeed, we do not want the pathologist to miss any potential cancerous sample.

Random Forest produced many FalsePositive classification for grade (it classify a negative sample as positive i.e classify G3 or G4 grade as G2). This is understandable as early stage grade are difficult to detect: our algorithm promotes over-classification.

Random Forest did great in classifying survival. A few patient only were misclassified. It successfully detected entire TruePositive (TP for grade and TP for survival) with 90% accuracy (TCGA-B0-4813, TCGA-CZ-5468, …).

SVM did not produce full FalsePositive (FP for grade and FP for survival at the same time). It did not produce full FalseNegative (FN for grade and for survival) which is great as with Random Forest!

It did better than Random Forest in classifying grade, but it was really bad in classifying survival. It did not detect extremely low survival case (TCGA-BP-4337) but fortunately it raised an alarm according to the binary grade.

Both algorithm correctly classified complex label (TruePositive for grade and TrueNegative for survival) as with TCGA-CJ-4874.

Finally, both algorithms raised an alarm for at least one class (either survival or grade) when necessary. This is a good news, the pathologist is warned if necessary thanks to one of the class.

VI. THE GUI

The targeted end-users of this project are the pathologists who want to get support when diagnosing cancer images. To this end, we need to design a GUI that can help the pathologist analyzing several images in an appealing way. Every parameters used to tune the algorithms in the backend should be

easily available. Our GUI shows both the image results of segmentation and the analytical results of the classification.

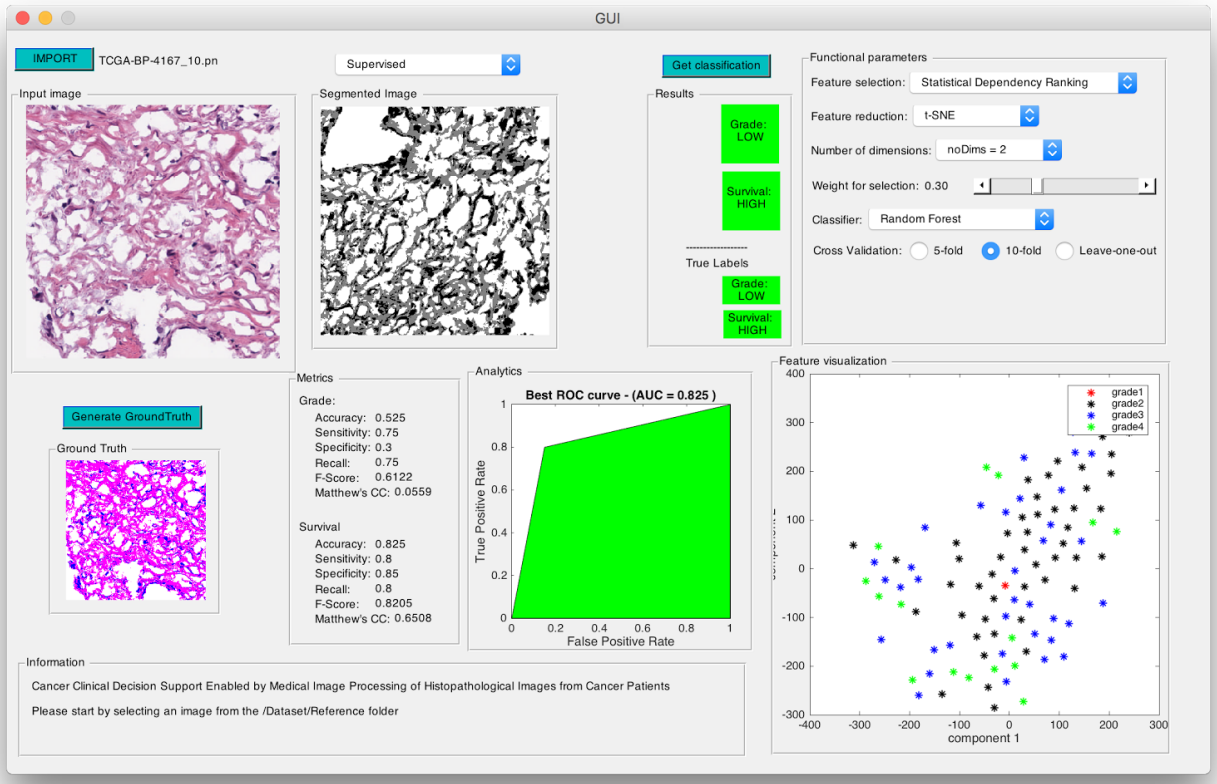


Figure 13: The interactive GUI

The interactive feature of the GUI let you choose the feature selection and feature reduction algorithms. You also have the possibility to generate a manual ground truth image that can be used as the input image for feature extraction. We can also select the number of dimensions or tune some feature selection parameter. For each change, the data automatically updates and a new result is displayed. As a result, the interface determines which class is the image.

VII. FUTURE WORK

There are plenty of improvements that can be done for the project. We would like to incorporate more bins in our color feature extraction to have more discreet results. Texture features can be computed from a lot of tools such as GLCM, but also Wavelet, Fractals or other object-based techniques. It would be really interesting to test them all and find the well suited one for histopathological images. For Feature Selection, we would like to try more combination of methods and see if we attain any improvement in results. After detailed analysis, we learnt that nuclei information plays a very important role for different grades. For example, Grade 2 images will have more circular and uniform distribution of nuclei, while Grade 4 will have unstructured pleomorphic tendency for nuclei and non-uniformly distributed as well. We would like to incorporate this information to further enhance our feature set and improve performance for Dataset2 and Dataset3. Working with machine learning algorithms requires skill. There a lot of parameters that need to be tweaked for each algorithm. We can perhaps improve more by playing around with more parameters for each algorithm and come up with better models. With the recent advancements in the field of deep learning, it would be great to see how medical data can leverage this. Scarcity of Data was a limiting factor and so, we had not been able to try such algorithms at this point of time, but would want to incorporate in future. Thanks to the flexible design of our GUI, we would be able to incorporate these new techniques easily.

References

1. Linda G. Shapiro and George C. Stockman, "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, 2001
2. Yan Xu, Weakly supervised histopathology cancer image segmentation and classification, 2014
3. Hui Zhang et al., Image Segmentation Evaluation : a survey of unsupervised methods, 2007
4. Kothari S, Phan JH, Stokes TH, Wang MD. Pathology imaging informatics for quantitative analysis of whole-slide images. Journal of the American Medical Informatics Association. 2013
5. Tony Chan, Luminita Vese, "An active contour model without edges", Scale-Space Theories in Computer Vision 1999
6. Tony Chan, Luminita Vese, "A multiphase level set framework for image segmentation using the Mumford and Shah model", International journal of computer vision, 2012
7. Olcay Sertel et al., "Histopathological Analysis Using Model-Based Intermediate Representations and Color Texture", Journal of Signal Processing Systems, 2009
8. Mohamed Salah et al., "Multiregion image segmentation by parametric kernel graph cuts", IEEE Transactions on Image Processing,

2011
9. R. Unnikrishnan et al., “Measures of Similarity”, IEEE Workshop on Applications of Computer Vision, 2005
10. M. Levine, A. Nazif, “Dynamic Measurement of Computer Generated Image Segmentations” IEEE Transactions on Pattern Analysis, 1985
11. Reinhard E, Adhikhmin M, Gooch B, Shirley P. Color transfer between images. Computer Graphics and Applications, IEEE. 2001
12. Magee D, et al. Colour normalisation in digital histopathology images. 2009
13. H. Zhang, J. Fritts, S. Goldman, An entropy-based objective evaluation method for image segmentation, in: Proceedings of SPIE-Storage and Retrieval Methods and Applications for Multimedia, 2004.
14. J. Liu, et al, Multi-resolution color image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994
15. J. Weszka, A. Rosenfeld, Threshold evaluation techniques, IEEE Transactions on Systems, Man and Cybernetics, 1978
16. R.A. Fisher, “The statistical utilization of multiple measurements,” Annals of Eugenics, 8:376-386, 1938.
17. Meyer, F, “Topographic distance and watershed lines”, Signal Processing, 1994
18. M. Datar et al., “Color and texture based segmentation of molecular pathology images using HSOMS”, Biomedical Imaging: From Nano to Macro, 2008
19. Faliu Yi et al,. “A survey of graph-cut methods”, International Conference on Systems and Informatics, 2012
20. K. Mao et al. “Supervised Learning-Based Cell Image Segmentation for P53 Immunohistochemistry”, IEEE Transactions on Biomedical Engineering, 2006
21. Yan Xu, Weakly supervised histopathology cancer image segmentation and classification, 2014
22. Textural Features for Image Classification, Robert M. HARALICK et al, IEEE Transaction on System, Man and Cybernetic, 1973
23. The effect of variation in illuminant direction on texture classification, Michael J Chantler, Department of Computing and Electrical Engineering Heriot-Watt University, 1994
24. L. Boucheron, "Object-and spatial-level quantitative analysis of multispectral histopathology images for detection and characterization of cancer," PhD thesis, University of California, Santa Barbara, 2008
25. Visualizing data using t-SNE, Laurens van der Maaten et al, Journal of Machine Learning Research, 2008
26. Nonlinear dimensionality reduction, JA Lee, M Verleysen, Information Science and Statistics, Springer, 2007
27. PCA vs LDA, Aleix M. Martonez et al, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001
28. Eric Postma, “Dimensionality Reduction, A Comparative Review”, Tilburg University, 2009
29. http://blog.kaggle.com/2012/11/02/t-distributed-stochastic-neighbor-embedding-wins-merck-viz-challenge/
30. Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits, Computer Speech & Language ,Volume 29, Issue 1, January 2015, Pages 145–171 .
31. Multifeature Prostate Cancer Diagnosis and Gleason Grading of Histological Images , IEEE Transaction on Medical Imaging, Vol. 26, No. 10, October 2007
32. Histological Image Feature Mining Reveals Emergent Diagnostic Properties for Renal Cancer, IEEE International Conference on Bioinformatics and Biomedicine, 2011
33. R. Bellazzi and B. Zupan, "Predictive data mining in clinical medicine: current issues and guidelines," Int J Med Inform, vol. 77, pp. 81-97, 2008.
34. F. Pereira, T. Mitchell, and M. Botvinick, "Machine learning classifiers and fMRI: a tutorial overview," Neuroimage, vol. 45, p. S199, 2009.
35. J. Kong, O. Sertel, H. Shimada, K. L. Boyer, J. H. Saltz, and M. N. Gurcan, "Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation," Pattern Recognition, vol. 42, pp. 1080-1092, 2009.
36. S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers, "Fisher discriminant analysis with kernels," in Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop., 1999, pp. 41-48.
37. C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, p. 27, 2011.
38. A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," IEEE computer, vol. 29, pp. 31-44, 1996.
39. T. M. Mitchell, "Machine learning. 1997," Burr Ridge, IL: McGraw Hill, 1997.
40. S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," Systems, Man and Cybernetics, IEEE Transactions on, vol. 21, pp. 660-674, 1991.
41. S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability," Soft Computing-A Fusion of Foundations, Methodologies and Applications, vol. 13, pp. 959-977, 2009.
42. Evaluation: from precision, recall and f-score to ROC, informedness and correlation, Powers et al., Journal of Machine Learning, 2011
43. Irina Rish, “An Empirical Study of naive Bayesian Classifier”, American Association for Artificial Intelligence, 2001
44. Leo Breiman, “Random Forests”, Statistics Department of University of California, Berkeley, 2001

APPENDIX

TYPE	FEATURE	COUNT
	Red Channel	16
	Green Channel	16
	Blue Channel	16
	L Channel	16
COLOR	A Channel	16
	B Channel	16
	Hue Channel	16
	Saturation Channel	16
	Value Channel	16
	Area	8
	Major Axis	8
	Minor Axis	8
	Eccentricity	8
	Orientation	8
SHAPE	Convex Area	8
	Solidity	8
	Filled Area	8
	Euler Number	8
	Extent	8
	Perimeter	8
	Autocorrelation	20

	Correlation	40
	Contrast	20
	Cluster Prominence	20
	Cluster Shade	20
	Difference Entropy	20
TEXTURE	Difference Variance	20
	Dissimilarity	20
	Energy	20
	Entropy	20
	Homogeneity	40
	Information Measure of Correlation	40
	Maximum Probability	20
	Sum Average	20
	Sum Entropy	20
	Sum Variance	20
	Inverse Difference Moment	60
TOTAL		(16*9)+(88*3)+(440*3) = **1728**

Appendix 1: List of all features used in the project