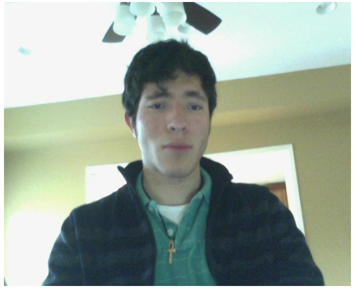
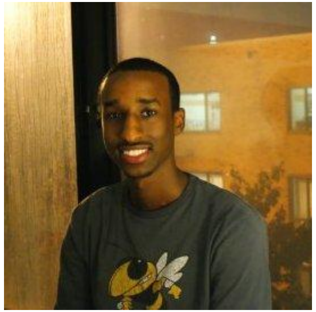
Module 3 - Prediction Modeling

Alex Coyle, Miguel S. Flores, Adam Kahin, Yuanda Zhu

Contents

Introduction and Problem Statement

- Cancer diagnosis is still a very imprecise process that often involves multiple physicians with differing opinions and yet there are many cases of misdiagnosis
- The development of effective diagnostic technologies is a crucial part of improving cancer care
- Our goal for this project is to develop a clinical diagnostic tool that uses image processing to distinguish tissue types in H&E stained images.
- In this module, our aim is to use the features selected in the previous module to develop methods to identify different tissue types, and correctly discriminate between tumorous and non-tumorous tissues.

Module 1 Summary - Image Segmentation

- Segment different tissue types within each image to have clearer boundaries
- Supervised Method
  - Histogram Thresholding
- Unsupervised Method
  - K-means Clustering
- Results
  - The unsupervised outperformed the supervised method due to the quality of of reference images

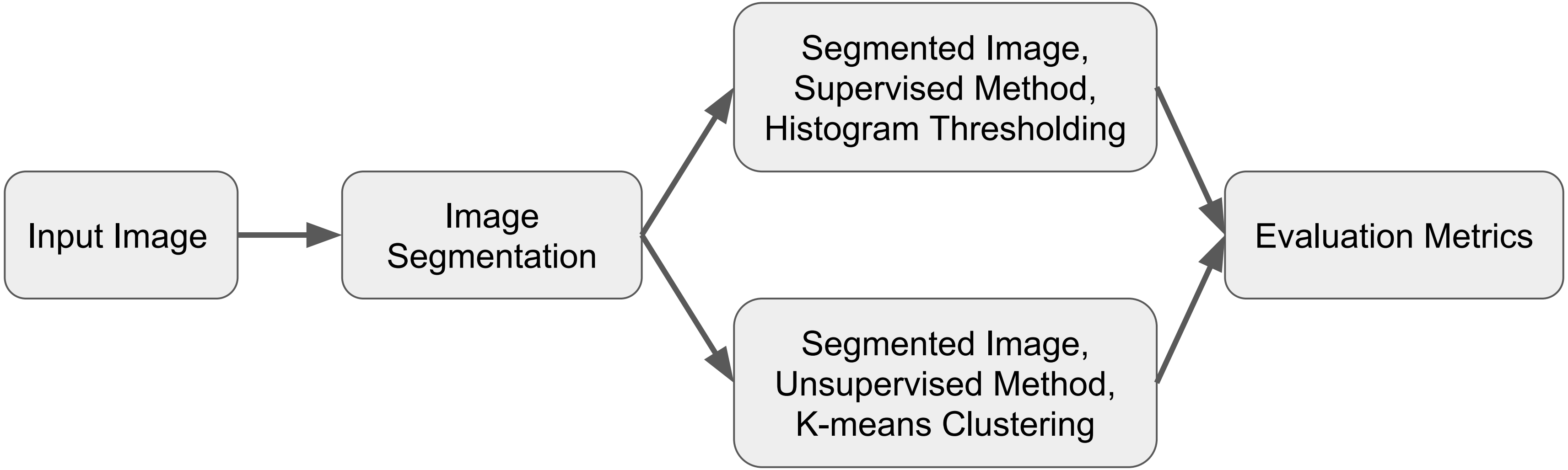
Module 2 - Feature Extraction and Selection

- Goal: Identify the statistical features that best distinguish images of different tissue classes.
- Over 100 features for color, texture and morphology were extracted for each image. These were compared for tissue classes to one another using kruskal-wallis test for statistical significance and R correlation coefficient.
- The 20 features with the most significance were chosen for use in our prediction models

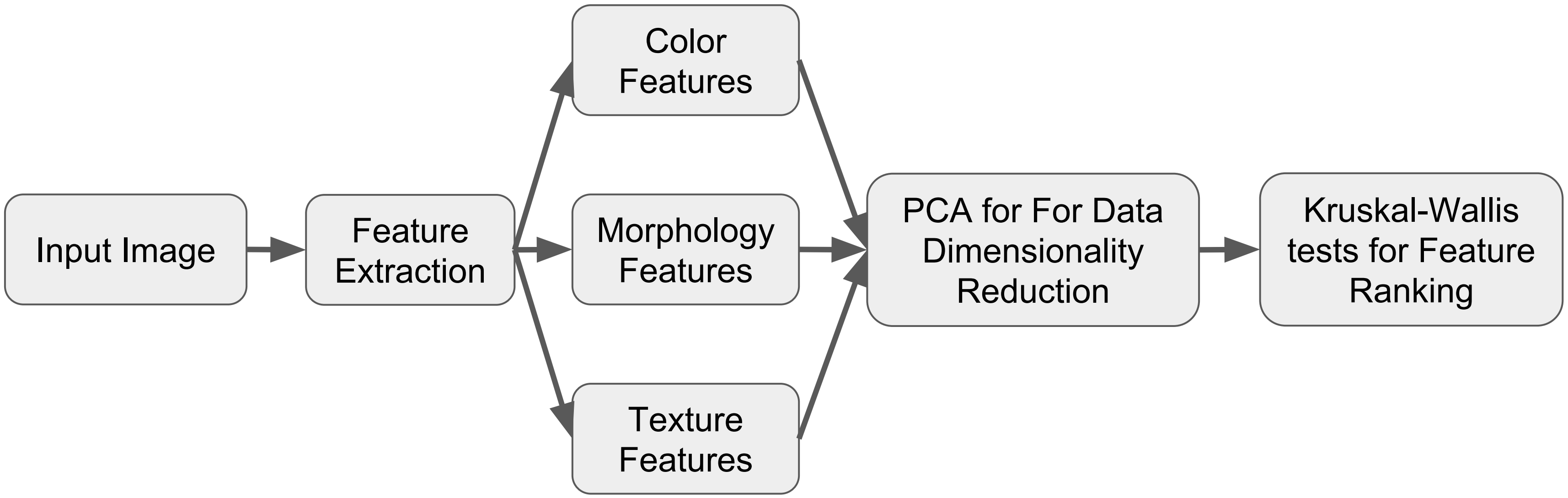
Literature Review

Summary	Title	Authors	Year
The paper explained neural network method by a multi-layer perceptron. The authors defined a clustering of data built from hidden layer representation after the model was trained with relevant variables.	A Methodology to Explain Neural Network Classification	Raphael Feraud, Fabrice Clerot	2001
The metric is trained with the goal that the k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin	Distance Metric Learning for Large Margin Nearest Neighbor Classification	Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul	2005+
This document explains how Support Vector Machines works as well as the procedure to implement SVM in Matlab successfully.	Support Vector Machines for Binary Classification	MathWorks Documentation	2009

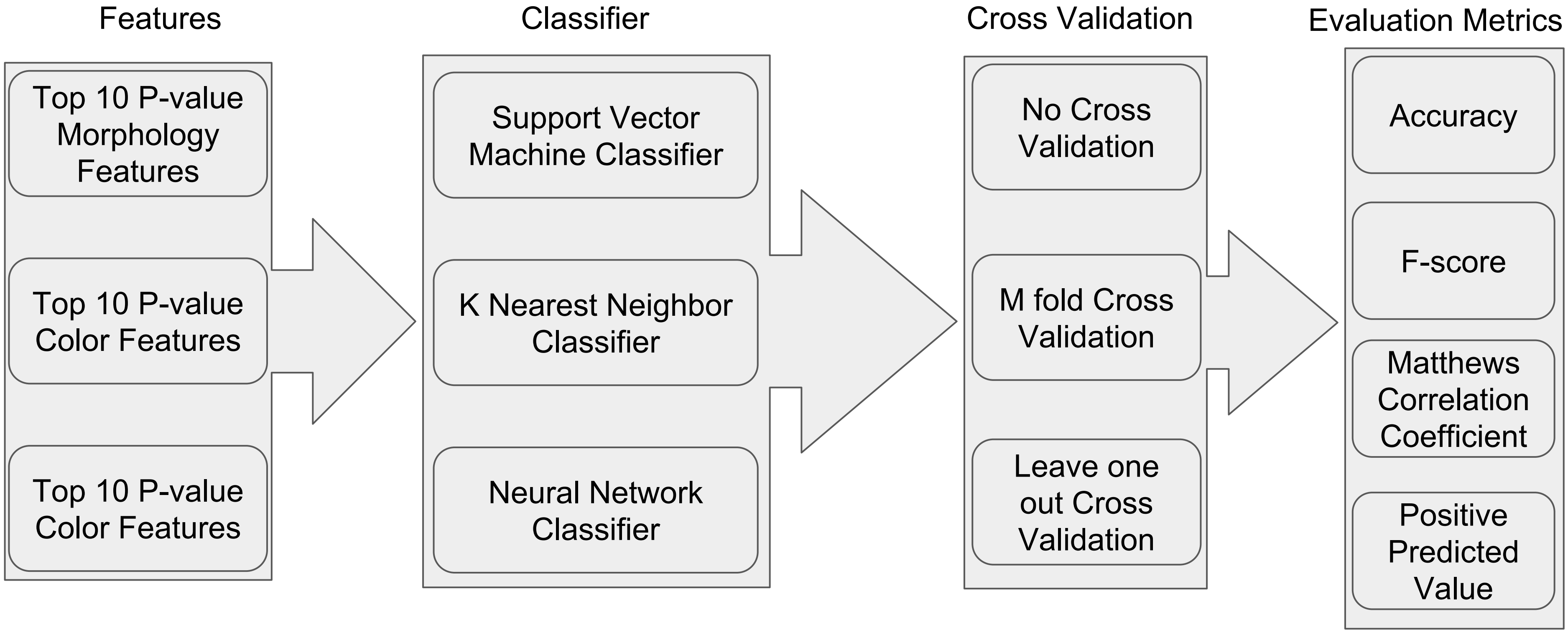
Flow Chart for Module 1 Image Segmentation



Flow Chart for Module 2 Feature Extraction



Flow Chart for Module 3 Classification



K-Nearest Neighbor (KNN)

- The first method we chose to use was K-Nearest Neighbor
- This method compares the data point of an image in the testing set to every image in the training set
- It then finds the value in the training set that is closest to the testing value and gives an output of the location of this value
- Based on this location we are able to assign a value of 1, 2, or 3 corresponding to to Stroma, Necrosis, and Tumor respectively

K-Nearest Neighbor (KNN)

- This methodology is then repeated 30 times for each feature based on our top 30 features from module 2, giving 1 tissue prediction per feature.
- The mode of these 30 predictions is then taken as the final prediction
- This method is then cross validated through K-folding and the leave-one-out method

K-Nearest Neighbor (KNN) Cross Validation

- Our KNN K-folding validation is accomplished using 4 iterations of 10 images from each class in the testing set with 60 images from each class in the training set giving each data point 270 total comparison neighbors
    - Ex: 61-70 Testing set , 1-60 Training set, 71-100 Validation set
- In our leave-one-out method we test 1 image from each class against 60 in the training set, with 39 in our validation set using our KNN method for a total of 297 comparison neighbors
- Leave-one-out is then repeated 40 times so every image in each class is given a prediction
    - Ex: 65 Testing set, 1-60 Training set, 61-64 66-100 Validation set

K-Nearest Neighbor Sample Table and Accuracy

Img #	Stroma	Necrosis	Tumor
61	1	1	3
62	1	2	3
63	1	1	3
64	1	2	3
65	1	3	2
66	1	2	3
67	1	2	3
-	-	-	-
100	3	2	2

KNN 1-60 Training 61-100 Test

Overall Accuracy: 77.5%

KNN K-Fold Cross Validation

Overall Accuracy: 77.67%

KNN Leave-one-out Validation

Overall Accuracy: 81.34%

Neural Network (NN)

- The second method we chose was neural network method.
- We used Matlab built-in GUI nnstart, Pattern Recognition to help us select data, create and train a network, and evaluate its performance using cross-entropy and confusion matrices

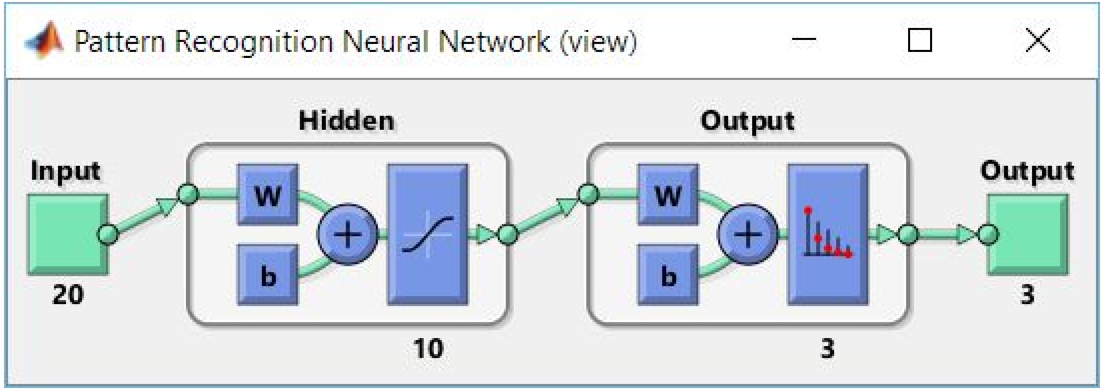


Figure 1. Neural Network Schematic Diagram.

Neural Network (NN) Continued

- Input: A 20*300 matrix. 20 selected features of 300 images.
- Target Output: A 3*300 matrix.
  - N: Necrosis; S: Stroma; T: Tumor
  - N/S/T 1: the first labeled image of Necrosis, Stroma or Tumor

	N1	N2	...	N99	N100	S1	S2	...	S99	S100	T1	T2	...	T99	T100
N	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 2. Target output matrix. Only the 1s and 0s are in the actual target output.

Neural Network (NN) Cross Validation

Cross validation is approached through changing the percentage of validation and testing set.

Original:

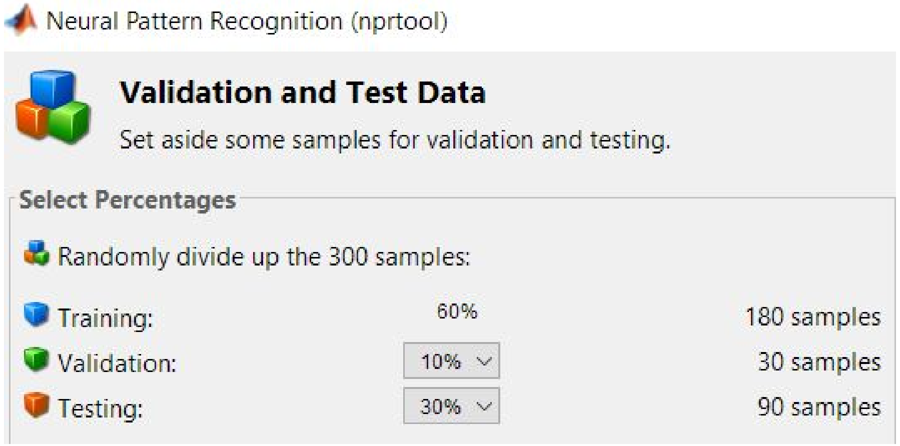


Figure 2. Screenshot of percentage of training, validation and testing sets

K-folding: Validation 20%, Testing 20%

Leave-one-out: Validation 30%, Testing 10%

Neural Network (NN) Results

- Prediction matrix:
  - Different prediction results are generated even with exactly the same inputs and settings.

Wrong Prediction Results

	N1	N2	...	N99	N100	S1	S2	...	S99	S100	T1	T2	...	T99	T100
N	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
S	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
T	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1

Table 3. Prediction result matrix. Only the 1s and 0s are in the actual result.

Neural Network (NN) Results Continued

Figure 3. Output Confusion Matrix.

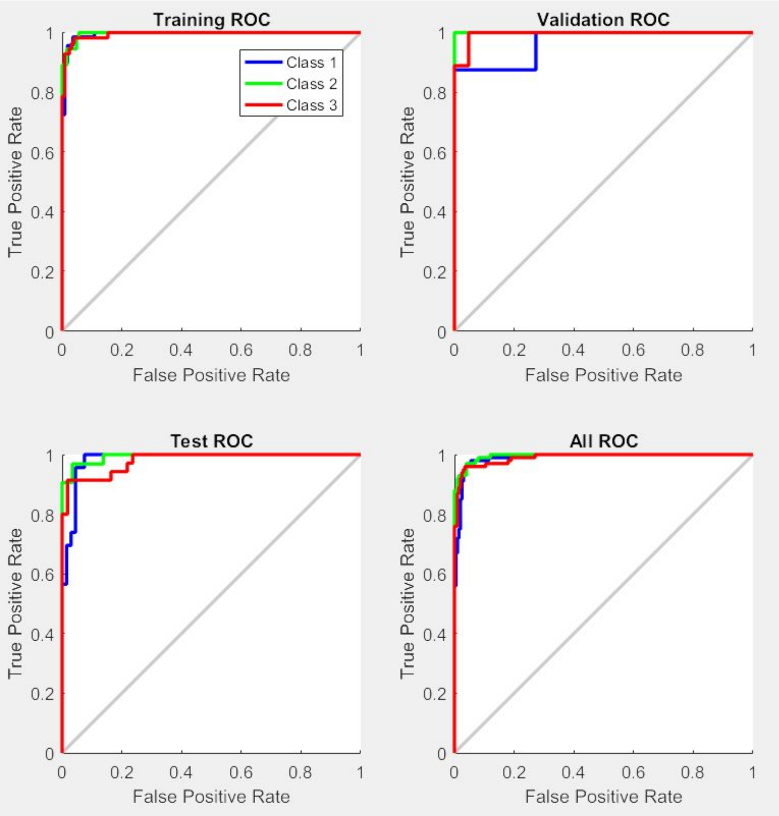


Figure 4. Output ROC.

Support Vector Machines (SVM)

- SVM is a supervised learning model with many learning algorithms that analyzes data for classification.
- Classifies data by finding the best plane that separates all data points between classes.
- SVM is implemented by two built-in functions in Matlab:
  - `SVMModel = fitcsvm(X,Y)`
  - `[label,score] = predict(SVMModel,TestX)`

Support Vector Machines (SVM)

- Three separate models are created:
  - Stroma Model ”using Stroma specific Training Set”
  - Necrosis Model ”using Necrosis specific Training Set”
  - Tumor Model ”using Tumor specific Training Set”
- A testing set containing 180 images are then input to each model.
- Prediction with the highest score is chosen as the prediction for the image.
- Cross Validation Methods:
  - K-Fold
  - Leave One Out

Support Vector Machines (SVM)

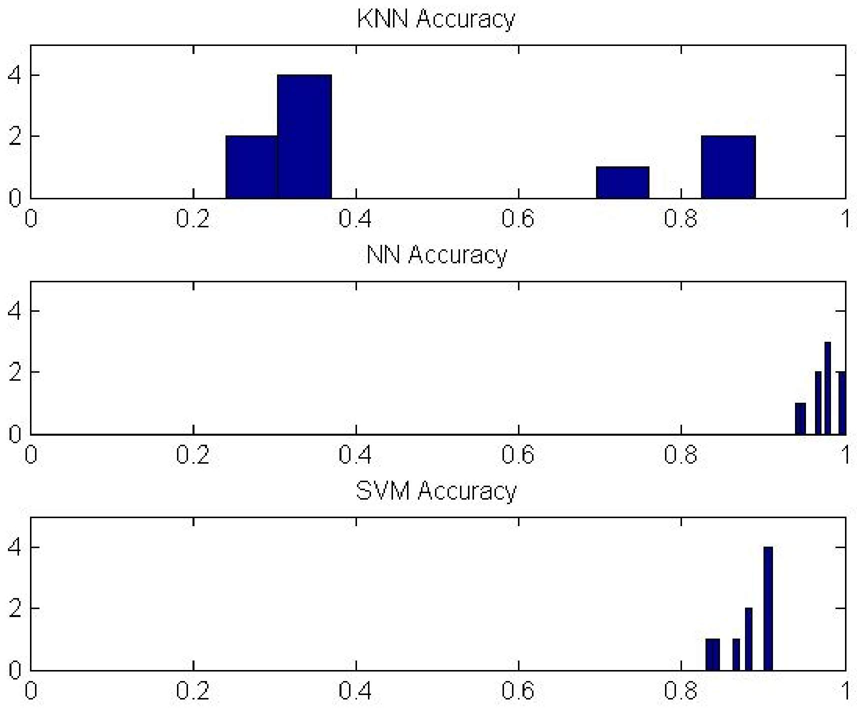
- Accuracy:
  - Stroma Model : 87%
  - Necrosis Model : 88%
  - Tumor Model : 88%
  - **Combined** : 87.6%
  - K-Fold : 88%
  - LOO : 88%

Performance Metrics

- Goal of these metrics was to assess these classification methods critically in order to better design and optimize our diagnostic assistance tool.
    - Accuracy
    - F-score
    - Mathew’s Correlation Coefficient (MCC)
    - PPV

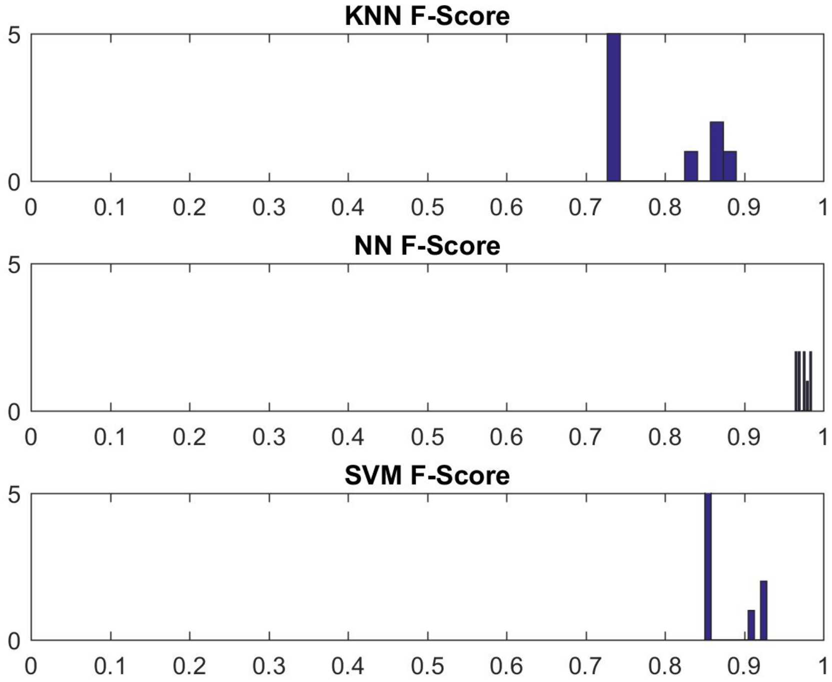
Accuracy

$$Accuracy = \frac{TP}{N}$$



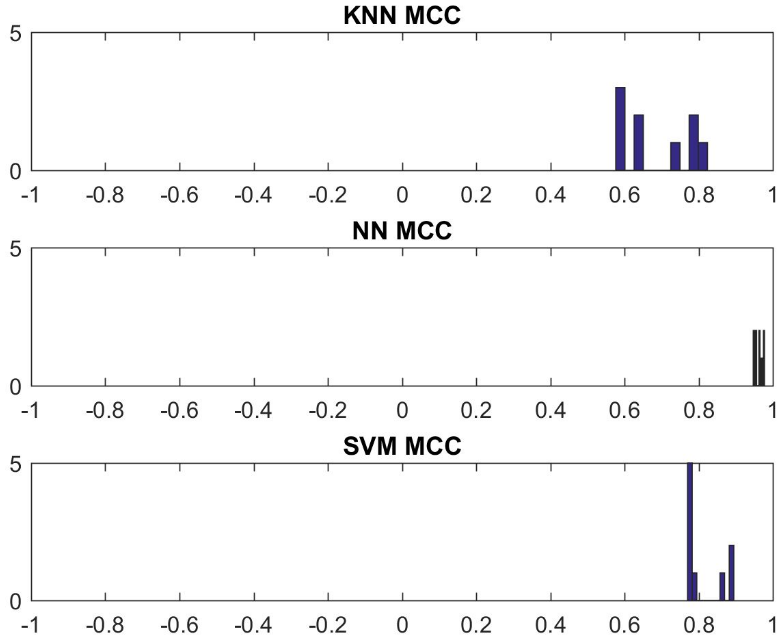
F-score

$$F = \frac{2TP}{2TP + FP + FN}$$



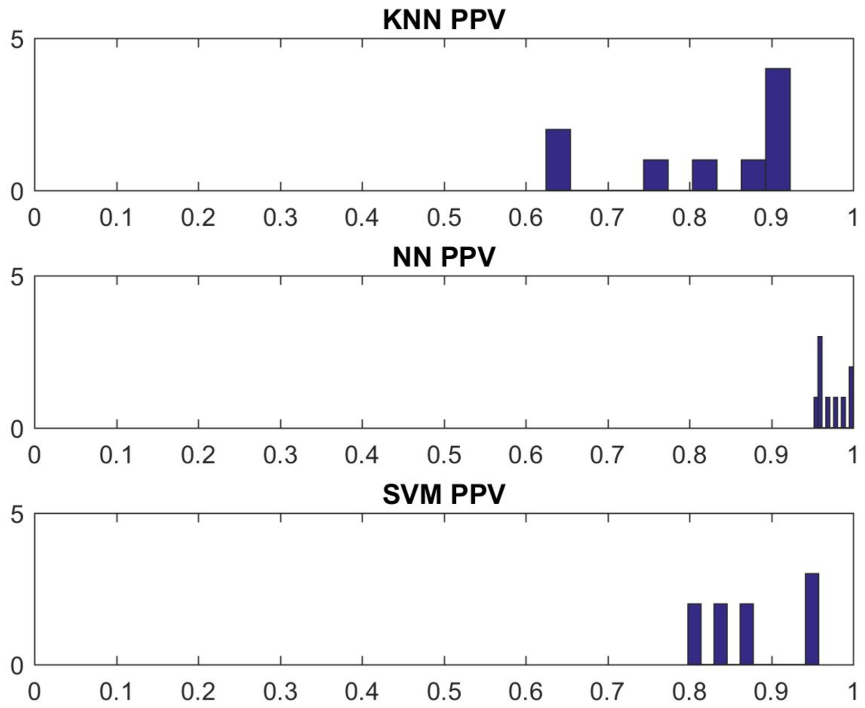
Mathew's Correlation Coefficient (MCC)

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(FN+TP)(TN+FP)(TN+FN)}}$$



Positive Predictive Value

$$PPV = \frac{TP}{TP+FP}$$



Conclusions

- KNN
  - The majority of this methods inaccuracies come from the mis-prediction of Stromas as Necrosis and Necrosis as Stromas
  - Our KNN method was accurate at distinguishing tumors from the other two tissues types (~91%)
- NN
  - The accuracy of neural network method is almost equal to detect the three classes of tissue image.
  - Overall accuracy is ranging from 90% to 97%.
- SVN
  - The SVM method used was able to distinguish between each class of tissue at 87.6%.
  - Due to the similar accuracy between the LOO and K-Fold method. We suggest that the K-Fold method be used due to the LOO processing time.

Future Work

- Identify more and better features for each classifier
- Give each feature a weight in our classifiers based on its P-value rank
- Adjust certain parameters on each classifier to achieve better performance