Machine Learning for Brain Histological Image Analysis

COGS 444 Final Report

Amy Hynes

April 14, 2020

1 Abstract

Manual image analysis for immunohistochemistry (IHC) images is time consuming and quantification results vary between researchers. This project experimentally determines that a decision tree machine learning model gives the best results when automating this task, and implements this model as software to analyze IHC images. These images are of mice brain slices which have been stained for different neural and neuropathological processes. Software allows for more efficient cellular-level quantification and results that are potentially as reliable as manual techniques. The models are be trained and tested on sample Iba1 images from Stephanie Tullo's project in the CoBrA lab.

2 Introduction

Neuropsychiatric disorders are studied using a variety of research methods including behavioural tests, assays of brain structure and function, and histological and immunohistochemical (IHC) analyses of brain tissue. IHC combines the immune response of various cells in the brain, including glial cells and neurons, with the visibility of histochemistry ("Immunohistochemistry / IHC Antibody-Brain Tissue"). Immuno-stains produce a variety of different IHC results depending on the immune response that stain causes. Different stains cause immune responses in different types of cells. Immuno-stained slices of brain tissue can be photographed under a microscope with or without fluorescence, depending on the stain. Once the tissue has been photographed, the IHC image can be analyzed to determine the number of cells stained.

Analysis of IHC images is typically done manually, and while manual segmentation and cell counting provide accurate results, the process can be time consuming and subject to rater bias. Multiple raters can reduce analysis time; however, this adds variance to the data due to inconsistent inter-rater reliability [2]. Automating this process would benefit researchers who use IHC imaging by reducing image analysis time and variability.

A variety of approaches have been used to automate cell counting in immunohistochemistry images. This automation started by using algorithms based on intensity thresholding, edge detection, template matching, and active shape models [4].

As machine learning became more popular, different models have been trained for IHC image analysis. Machine learning approaches began with basic algorithms including support vector machines (SVM) and random forests (RF)[4].

Arteta et al. explored the use of SVM for cell detection in microscopy images. Their algorithm was trained with simple dot annotation on sample images. Their use of SVM achieved state-of-the-art accuracy on hematoxylin and eosin (H&E)-stained images [1]. H&E staining reveals a considerable amount of microscopic anatomy, and is used to diagnose a range of histopathological conditions.

Pham et al. experimented with SVM and RF for cell counting and segmentation of IHC images in the spinal cord. They performed preprocessing for both algorithms, and their RF contained 200 bagged classification trees [4].

The purpose of this research project is to design and implement software that uses machine learning to perform analysis on IHC images. Multiple machine learning models are evaluated to determine which has the highest performance on the Iba1 sample images provided by Stephanie Tullo. The decision tree model showed the strongest performance in both accuracy (77%) and recall (87%) and would be the most useful in laboratory settings. Accompanying this report is a python script to which researchers can provide their Iba1 IHC images as input, and the program reports the positive cell count in an accurate and precise manner.

3 Methods

3.1 Dataset

The data used in this project are from a research project by Stephanie Tullo, and are IHC IBA1 images of M83 αSyn^{A53T} transgenic mice brains, a Parkinson's Disease model. The images are split into training and testing sets, and thresholded and segmented to find candidate patches containing a single cell. These candidate patches are manually tagged as positive or negative. There are 979 patches total, 837 patches in the training set, and 142 patches in the test set, for an 85-15 train-test split.

3.2 Models

This project follows the structure of the study by Pham et al. who tested multiple models on IHC image analysis [4]. Images go through a preprocessing pipeline where the signal to noise ratio is increased; the resulting image undergoes intensity thresholding, and candidate patches are extracted. Features including shape, texture, and histogram of oriented gradients are extracted from the patches. Shape features include solidity, orientation, diameter, area, eccentricity, convex area, major axis length, minor axis length, and extent. Texture features are based on the MR8 filter banks which include 36 bar and edge filters, a Gaussian filter, and a Laplacian of Gaussian filter. The eight highest responses are extracted to maintain rotation invariance. Histogram of Oriented Gradients (HoG) features were also extracted. These features are normalized using a min-max scaler and input to the models.

Eight models were tested, and the results can be found in the following section. These

models were implemented using Scikit-learn [3]. The models are evaluated based on accuracy, recall, and precision as compared to manual analysis as the ground truth. Cross validation search was used to tune the hyperparameters of each model. For models with less than 1000 hyperparameter combinations, exhaustive grid search was performed. For models with more than 1000 hyperparameter combinations, randomized search was performed with 1000 iterations to limit computing time.

3.3 Software

The decision tree model is packaged as software for use by researchers. The code is be available on GitHub with instructions on how to install the requirements and run the script.

4 Results

Of the 142 patches in the test set, 99 are positive, so the accuracy of a model which guesses positive for every patch is 69.72%. Each of the models investigated performs better than this baseline, and none of them guesses one class for every instance.

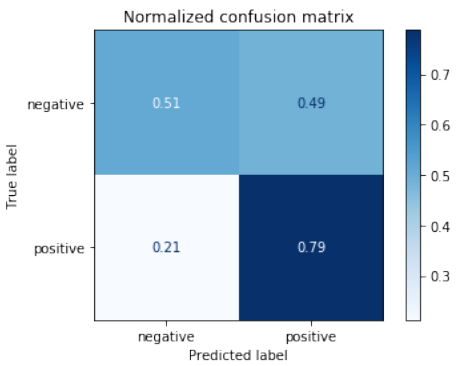
The models tested are a K-nearest neighbours (KNN) classifier with 7 neighbours, a support vector machine (SVM) classifier with a linear kernel, a SVM classifier with a Radial Basis Function (RBF) kernel, a Gaussian Process classifier, a Decision Tree classifier with a max tree depth of 50, a Random Forest classifier of 100 decision trees with a max tree depth of 50, a multi-layer perceptron (MLP) classifier, and an AdaBoost classifier with 50 estimators. See Table 1 below for the accuracy, recall, and precision of each classifier, and the number of positive cells they predict. Figure 1 shows their confusion matricies.

Table 1: Performance of each Model on Test Set

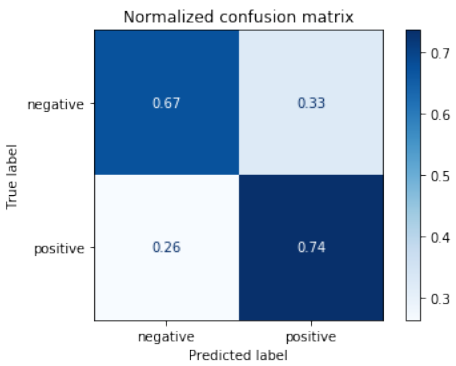
Model	**Accuracy %**	**Recall %**	**Precision %**	**Number of Positive Cells Predicted**
KNN	70.42	78.79	78.79	99
Linear SVM	71.83	73.74	83.91	87
RBF SVM	71.13	74.75	82.22	90
Gaussian Process	71.13	74.75	82.22	90
Decision Tree	76.76	86.87	81.13	106
Random Forest	71.13	74.75	82.22	90
MLP	70.42	74.75	81.32	91
AdaBoost	71.83	75.76	82.42	91

Figure 1: Confusion Matrices

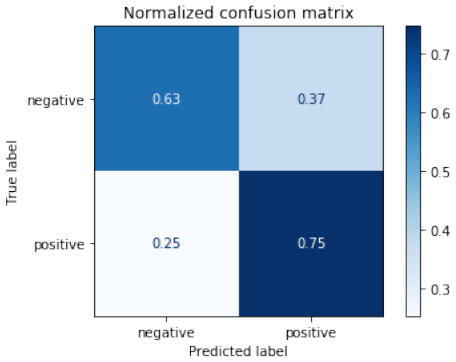
(a) K-Nearest Neighbors



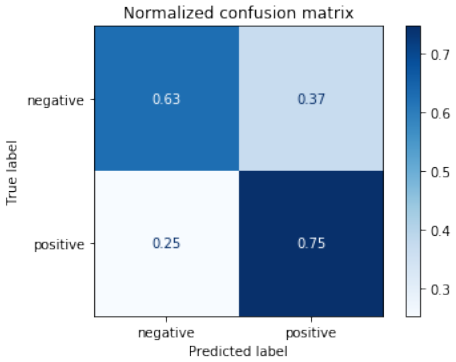
(b) Linear SVM



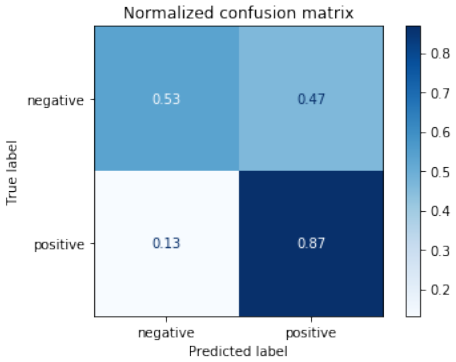
(c) RBF SVM



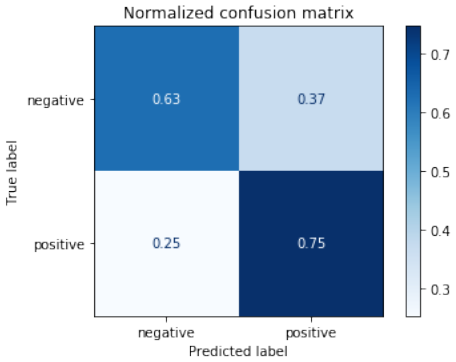
(d) Gaussian Process



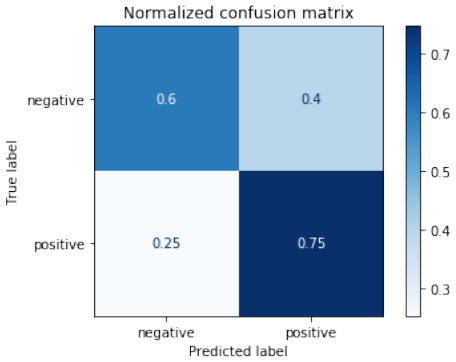
(e) Decision Tree



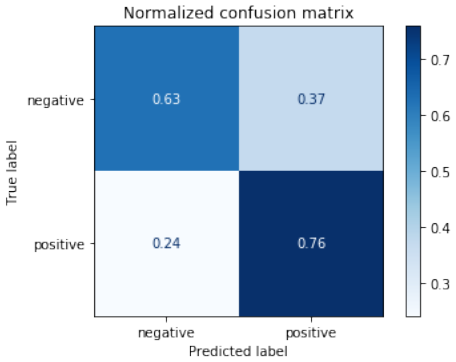
(f) Random Forest



(g) Multi-layer Perceptron



(h) AdaBoost



5 Discussion

These results show that the decision tree model performs better than the other models with about 77% accuracy. It also has a higher recall score, but the precision is about the same as the other models. The confusion matrix in figure 1(e) shows that it identifies positive cells correctly 87% of the time, but only classifies negative patches correctly 53% of the time. The Linear SVM model does the best at classifying negative patches with the correct label 67% of the time, but that comes with a trade-off as it only classifies positive cells correctly 74% of the time for an overall accuracy of 72% and the highest precision score of 84%. AdaBoost performs similarly well at correctly classifying both classes, but differs with a slightly higher recall and lower precision than the linear SVM model.

The true number of positive cells in the image is 99, and the KNN model predicts positive for 99 patches. This is a good result in terms of an accurate cell count, but the model does not classify negative or positive cells well so its accuracy is only 70%. Decision tree predicts 106 positive patches, while linear SVM predicts 87. These results are as expected since the decision tree predicts positive cells more accurately and also predicts positive for nearly half of the negative patches, while the linear SVM is wrong more often in its prediction of positive cells. The AdaBoost classifier is in the middle of these two and predicts 91 positive cells.

Grid search was used to tune the hyperparameters for the KNN, linear SVM, gaussian process, and AdaBoost models as they each had less than 1000 hyperparameter combinations. Randomized search was used for RBF SVM, decision tree, random forest, and MLP models. There is no significant difference in performance between these two groups, indicating the search methods perform about the same for tuning hyperparameters for this task.

6 Conclusion and Future Work

The decision tree model showed the strongest performance in both accuracy and recall. It also provided the cell count closest to the true value aside from KNN. This model would be the most useful in laboratory settings as it classifies the most positive cells correctly and allows the researcher to check the positively classified cells and remove false positives. If raw cell counts are more important than specific cells being classified correctly, the KNN model provides the correct number of positive cells in this instance.

As future work, this image analysis pipeline could be generalized to other stains. Adding preprocessing pipelines for other stains will allow the model to classify different biomarkers. Different IHC stains are used to visualize different cell responses in a brain slice, including multiple types of glial cells and their activation, neurons, and leukocytes.

This project would also benefit from having a second human rater to compare the "true" classifications for each cell. Pham et al. reported a significant difference in cell counts between raters, so it is possible that the models presented here perform better than another human when compared to the first rater [4]. Having a second rater would reduce bias in the model and allow comparison of human vs. human and human vs. model accuracy.

References

[1] Carlos Arteta, Victor Lempitsky, J. Alison Noble, and Andrew Zisserman. Detecting overlapping instances in microscopy images using extremal region trees. *Medical Image Analysis*, 27:3 – 16, 2016. Discrete Graphical Models in Biomedical Image Analysis.

[2] Magali Lacroix-Triki, Simone Mathoulin-Pelissier, Jean-Pierre Ghnassia, Gaetan Macgrogan, Anne Vincent-Salomon, Véronique Brouste, Marie-Christine Mathieu, Pascal Roger, Frédéric Bibeau, Jocelyne Jacquemier, Frédérique Penault-Llorca, and Laurent Arnould. High inter-observer agreement in immunohistochemical evaluation of her-2/neu expression in breast cancer: A multicentre gefpics study. *European Journal of Cancer*, 42(17):2946 – 2953, 2006.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] B. Pham, B. Gaonkar, W. Whitehead, S. Moran, Q. Dai, L. Macyszyn, and V. R. Edgerton. Cell counting and segmentation of immunohistochemical images in the spinal cord: Comparing deep learning and traditional approaches. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 842–845, 2018.