Machine Learning for Brain Histological Image Analysis

**Amy Hynes, February 21st 2020**

**Abstract**

Manual image analysis for immunohistochemistry (IHC) images is time consuming and quantification results vary between researchers. This project experimentally determines which machine learning model gives the best results when automating this task, and implements that 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 will allow for more efficient cellular-level quantification and potentially results that are as reliable as manual techniques. The software will be trained on sample images from Stephanie Tullo’s project in the CoBrA lab, then tested on the remaining images and optimized to provide accurate results for different stains.

**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 (LaCroix-Triki et al. 1). Automating this process would benefit researchers who use IHC imaging by reducing image analysis time and variability.

Previous research has implemented machine learning for image analysis of microscopy data, including IHC images. A variety of machine learning models have been used, from simple to complex. Complex models require large amounts of data for training. Simple methods often achieve accuracy close to that of complex neural networks.

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 will be studied and tested to determine which has the highest performance on the sample images provided by Stephanie Tullo. Once the model is trained and optimized on that data set, extending the model will be investigated. Ideally, the software will perform well on multiple types of immuno-stains to provide the maximum value for researchers. The goal for this project is to produce a python script to which researchers can provide their IHC images as input and tell the program which stain(s) were used, and the script reports the positive cell count in a precise and accurate manner.

**Methods**

**Dataset:**

The data used so far in this project are from a research project by Stephanie Tullo, and are IHC IBA1 images of M83 αSynA53T transgenic mice brains, a Parkinson’s Disease model. This dataset may be supplemented with images from the from the lab of Dr. Jamie Near to facilitate training on the model on multiple IHC stains. 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. 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.

**Models:**

This project follows the structure of the study by Pham et al. who multiple models on IHC image analysis. 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. Textures 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 input to the models.

Eight models have been tested, and the results can be found in the following section. The models are evaluated based on accuracy, recall, and precision as compared to manual analysis as the ground truth. K fold cross validation will be used to tune they hyperparameters of different models. Once the models have been trained and evaluated on the data from Stephanie Tullo’s project, the most promising model will be generalized to work with multiple stains.

**Software:**

Once the model has been generalized, it will be packaged as software for use by researchers. The code will be available on GitHub with instructions on how to install the requirements and run the script. The current progress can be found at: <https://github.com/amyhynes/ihc-ml>.

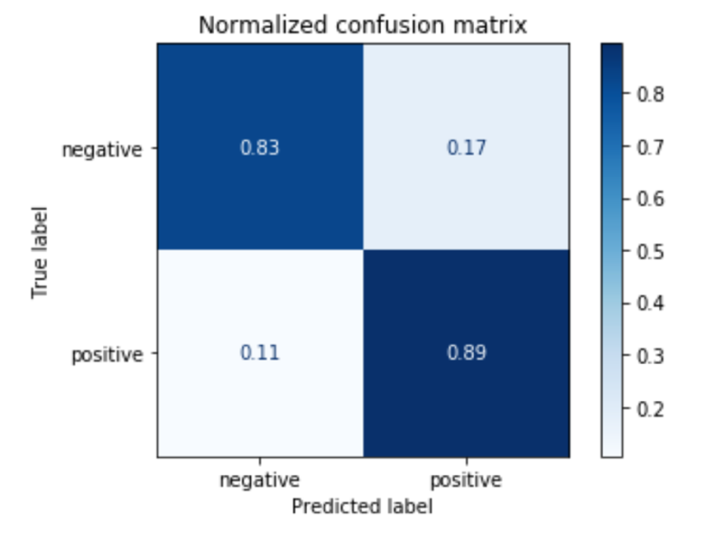
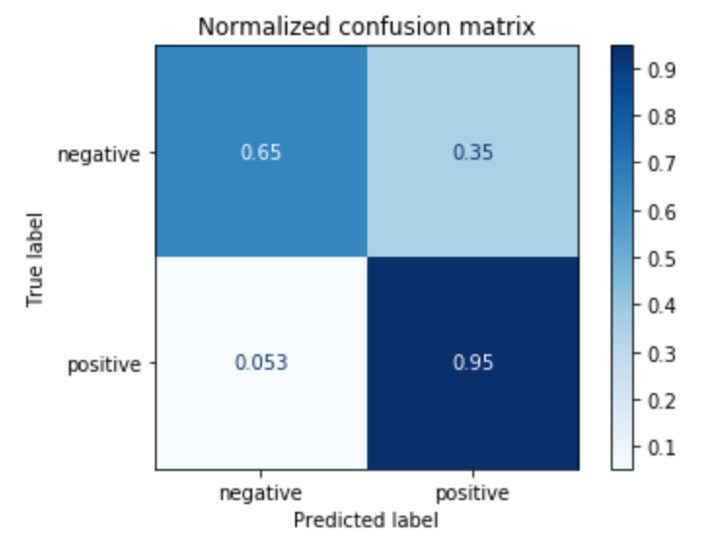
**Results**

One image has been run through the pipeline, and this image contains 208 candidate patches each containing one cell. 166 patches were used for training, and 42 for testing each of the models. The accuracy of these models on this set of patches serves as an indication of which models are promising for further investigation. All models are from the sci-kit learn library, and use default settings unless otherwise specified. The models tested were a K-nearest neighbours (KNN) classifier with 3 neighbours, a SVM classifier with a linear kernel and a regularization constant of 0.1, a SVM classifier with the default settings (RBF kernel), a Gaussian Process classifier with a 1.0 \* RBF(2.0) kernel, a Decision Tree classifier using the default settings, a Random Forest classifier with a max tree depth of 20, a MLP (multi-layer perceptron) classifier was used with a maximum of 800 iterations, and an AdaBoost classifier with 10 estimators. See below for the accuracy, recall, and precision of each classifier, and their confusion matricies.

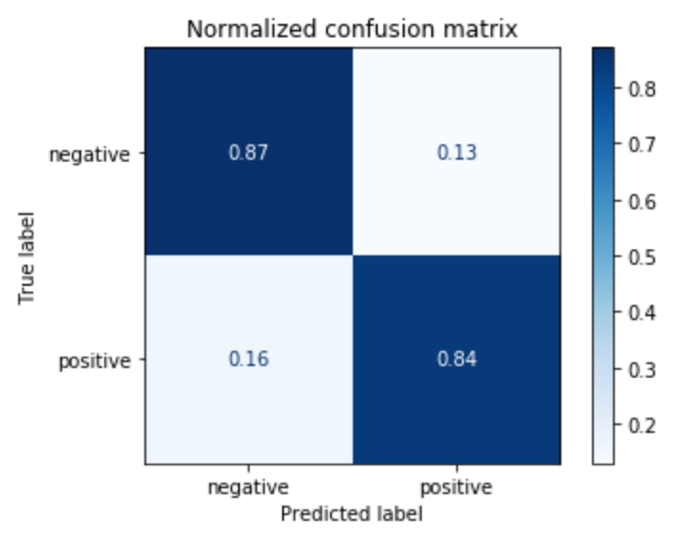
|  |  |  |  |
| --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Recall** | **Precision** |
| KNN | 0.786 | 0.947 | 0.692 |
| Linear SVM | 0.857 | 0.895 | 0.81 |
| RBF SVM | 0.857 | 0.895 | 0.81 |
| Gaussian Process | 0.857 | 0.895 | 0.81 |
| Decision Tree | 0.857 | 0.842 | 0.842 |
| Random Forest | 0.857 | 0.895 | 0.81 |
| MPL Neural Network | 0.857 | 0.895 | 0.81 |
| AdaBoost | 0.857 | 0.895 | 0.81 |

The confusion matricies for Linear SVM, RBF SVM, Gaussian Process, Random Forest, MLP Neural Network, and AdaBoost are the same.

Linear SVM, RBF SVM, Gaussian Process, Random Forest, MLP Neural Network, and AdaBoost

****

KNN

****

Decision Tree

**Additional Work**

Given the results, the KNN model will not be explored further. The next step is to segment and label additional image data. A second rater will be needed to count positive cells to reduce bias in the ground truth. Once more images have been labelled, patches from multiple images can be used as training data, and the algorithms can be tested on separate images, allowing positive cell counts to be obtained for an entire image. At that point, accuracy for each classifier can be re-evaluated, and the classifier which gives results closest to the true count of positive cells in an image will be chosen for further development. Once a model has been chosen, K-fold cross-validation will be used to tune the hyperparameters to optimize accuracy.

The preprocessing pipeline will then be generalized to work for multiple stains, allowing the model to classify different biomarkers. Once this is complete, the documentation will be written and the code made available online.

**Work Plan**

February:

* Label additional data and re-evaluate models
* Select best performing model

March:

* Tune hyperparameters to optimize performance
* Generalize model to work on multiple stains
  + Label additional data for other stains
* Document and release code

April:

* Final report

Works Cited

"Immunohistochemistry / IHC Antibody-Brain Tissue." *Sino Biological*, 2019, [www.sinobiological.com/immunohistochemistry-ihc-antibody-brain-tissue.html](http://www.sinobiological.com/immunohistochemistry-ihc-antibody-brain-tissue.html).

Lacroix-Triki, Magali et al. "High Inter-Observer Agreement in Immunohistochemical Evaluation of HER-2/neu Expression in Breast Cancer: A Multicentre Gefpics Study." *European Journal of Cancer*, vol. 42, no. 17, 2006, pp. 2946-2953, doi:10.1016/j.ejca.2006.06.020.

Pham, B. et al. "Cell Counting and Segmentation of Immunohistochemical Images in the Spinal Cord: Comparing Deep Learning and Traditional Approaches." *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 18-21 July 2018, pp. 842-845. doi:10.1109/EMBC.2018.8512442.