Lymphocyte Classification in Hyperspectral Images of Ovarian Cancer Tissue Biopsy Samples

Benjamin Paulson, Theodore Colwell, Natalia Bukowski, Joseph Weller, Andrew Crisler
John Cisler, Alexander Drobek, Alexander Neuwirth

Department of Electrical Engineering and Computer Science
Milwaukee School of Engineering
1025 N Broadway, Milwaukee, WI 53202
(paulsonb, colwellt, bukowski, wellerj, crislera, cislerj, drobeka, neuwirtha) @msoe.edu

Abstract

Current methods for diagnosing the progression of multiple types of cancer within patients rely on interpreting stained needle biopsies. This process is time-consuming and susceptible to error throughout the paraffinization, Hematoxylin and Eosin (H&E) staining, deparaffinization, and annotation stages. Fourier Transform Infrared (FTIR) imaging has been shown to be a promising alternative to staining for appropriately annotating biopsy cores without the need for deparaffinization or H&E staining with the use of Fourier Transform Infrared (FTIR) images when combined with machine learning to interpret the dense spectral information. We present a machine learning pipeline to segment white blood cell (lymphocyte) pixels in hyperspectral images of biopsy cores. These cells are clinically important for diagnosis, but some prior work has struggled to incorporate them due to difficulty obtaining precise pixel labels. Evaluated methods include Support Vector Machine (SVM), Gaussian Naïve Bayes, and Multilayer Perceptron (MLP), as well as analyzing the comparatively modern convolutional neural network (CNN).
1 Introduction

As effectively treating cancer continues to be a focus for doctors across the globe, an increasing number of hospitals are using needle biopsies to diagnose and monitor multiple types of cancer. The full procedure makes use of a hollow needle to extract a narrow column of tissue from a suspected tumor, which is then sliced, chemically stained to reveal features of cell biology, paraffinized for preservation, and annotated under a microscope to identify signs of disease progression. This current process of analyzing tissue is time-consuming, requires a trained pathologist, and is susceptible to human error during the Hematoxylin and Eosin (H&E) staining and deparaffinization stages.

Other recent works have explored the effectiveness of hyperspectral imaging: a technology for producing images of tissue by measuring light absorbance on precise wavelengths corresponding to specific chemical signals. Hyperspectral imaging of biopsy cores has shown potential as a faster, sample-preserving alternative to distinguish between tissue types without the need for H&E staining or deparaffinization. Inspired by “Deep Learning for FTIR Histology” [11], which used Fourier Transform Infrared (FTIR) images of breast biopsy cores and a convolutional neural network (CNN) to classify disease-relevant tissue types, we propose a method for detecting lymphocytes within biopsy cores of ovarian samples based on FTIR image data.

Previous works performing histological tissue-type segmentation were limited by poor annotation quality due to the presence of unannotated lymphocytes embedded within annotated tissue types. Therefore, we propose a method to overcome data-quality limitations from earlier models by comprehensively classifying lymphocyte annotations in a way that accurately assesses model performance. This standalone model for identifying lymphocytes within biopsy cores could be incorporated as a preprocessing step in future machine learning pipelines to intelligently filter lymphocytes out of the annotated regions of other tissues prior to training.

To create labeled data, lymphocyte and non-lymphocyte regions were annotated within ovarian biopsy cores. Adjacent stained cores – annotated by a professional pathologist, were referenced as a ground truth. Various machine learning models were then compared on their effectiveness of lymphocyte detection.
2 Methods

Ovarian biopsy cores from 100 patients were provided by Biomax US, Rockville, MD (TMA ID: BC11115c). For each patient, 2 adjacent cores were analyzed: one stained with traditional H&E staining, and a hyperspectral image for the other sample – with paraffin still on. The H&E-stained samples (single sample provided in Error! Reference source not found.) were analyzed and annotated by a professional pathologist (Yanping Zhang) as ground truth for reference while creating labeled data for the classical and modern neural networks.

The hyperspectral images of ovarian tissue cores (single sample provided in Error! Reference source not found.) were created using a Cary 600 FTIR spectroscopic microscope (Agilent Technologies) with 0.62 numerical aperture and averaged with 16 co-additions. Each hyperspectral image has 394 color channels ranging between 1000 cm⁻¹ and 3900 cm⁻¹.

2.1 Annotations

Using the open-source software GIMP, lymphocyte and non-lymphocyte tissue types were annotated at a pixel level on separate layers superimposed on the FTIR hyperspectral image. Tissue types were distinguished as white pixels within their respective lymphocyte and non-lymphocyte masks. Sections of the hyperspectral image were distributed among the team, where members were responsible for annotating specific tissue types on their respective regions. Member annotations masks were combined by tissue type.

Annotated white pixels serve as active pixels, which define a specified tissue type depending on the mask (lymphocyte or non-lymphocyte mask). Inactive pixels, defined as all the pixels without annotation, were ignored. Lymphocyte and non-lymphocyte masks
were exported as separate PNG files, which were used to create NumPy arrays for lymphocytes and non-lymphocytes. The layer masks for examples of lymphocytes and non-lymphocytes were exported and combined to create two mask files. These mask files and the hyperspectral data files were imported and loaded into NumPy arrays using NumPy, PIL[3], and spectral[4] libraries.

To visualize the location of our annotations and the number of annotations on one biopsy, the annotation was split into clusters based on their proximity. Figure 2 shows the graph of our current annotated data for lymphocytes on our biopsy core data.

![Figure 2: Annotation Data Visualizations](image)

To generate the visualization (see Figure 3), a k-means clustering algorithm was used to group each of the different annotation clusters on different biopsies together. The exact k-means algorithm was pulled from the sklearn implementation of k-means [12]. Matplotlib was then used to graph out the points of the annotations on top of the picture. The number of annotated pixels in each region is pictured in Figure 3 (bottom). The data visualization graph showed the distribution of the data. This helped guide data collection to limit bias towards a few samples. It also highlighted any misplaced annotations on incorrect biopsies.
2.2 Data Splitting

In experiment 1, the train test split method from the Ski kit-learn model selection module was used to split annotated pixels randomly between test and train groups. Due to pixels being annotated in groups, we decided on cutting the data in half vertically to avoid overfitting to the training data. Samples on the left half of the data were used for training while samples on the right half of the image were used for validation testing. This allows for better comparison between methods, as the training data is from different patients than the testing data. As shown in Figure 3 below, each sample is from a different patient. By training and testing models on different patient samples, samples can be evaluated in a manner that more closely resembles a clinical setting.

2.3 Classical Models

To begin testing classical models, an SVM was chosen due to its ability to provide effective results with our greater dimension size recorded with FTIR hyperspectral imaging (394 bands), alongside our smaller sample size – 100 total ovarian biopsy cores – in a 20292 px by 17912 px hyperspectral image. The SVM’s ability to create a subset of training points in its decision function allowed the model to consider only extreme features: when the kernel demonstrated both features of having lymphocytes and non-lymphocyte characteristics.
This ability was initially assumed to provide better predictability knowing that differences between annotations were minimal.

Gaussian Naïve Bayes classifiers inherently have broad “naïve” assumptions, assuming a Gaussian nature of the features. While this model was not chosen for a distinctive reason – as opposed to SVM or MLP – the results achieved, as outlined in 3.1, were surprising based on this model’s methods.

The multilayer perceptron model is one of the simplest artificial neural networks and was chosen based on this similarity to modern algorithms – such as a CNN. As an overview, the MLP structure contains an input and output layer, as well as hidden layer(s). During our testing, the default scikit-learn [2] hyperparameters were used. The MLP’s training process feeds inputs values forward – through the hidden layer(s) to the output layer – reaches a cost value for the model, backpropagates to restructure weights and biases based on that value, then continues training.

Each classical model outlined above-achieved results that were analyzed in the Experiments section.

2.4 Deep Learning

In addition to experimenting with classical machine learning models, deep learning was applied to our data through the use of a convolutional neural network (CNN). Our decision to experiment with CNNs was due to their ability to maintain the spatial structure of the data, recognizing key patterns that humans could likely distinguish as well. The Python framework Keras was used to construct the CNN. This allowed us to focus on optimizing the high-level architecture of the neural network rather than focusing on the low-level implementation details. In the future, we plan to use transfer learning with existing CNN architectures which are optimized for hyperspectral data to improve our results.

3 Experiments

Using implementations provided in scikit-learn, specific classical models (outlined in table 1; on next page) were tested and trained on the same dataset. While the classical classification models generally performed well with random-split data – with a 95.5% average between models consistently achieving an accuracy of >90% -- there are two noteworthy outliers: Gaussian Naïve Bayes, which achieved an accuracy of less than 50%; and a Support Vector Machine (SVM), which achieved an accuracy around 76%.

However, random-split data utilized the built-in train-test split with scikit-learn, resulting in frequent adjacently trained pixels being used during the testing phase. This unexpectedly increased the accuracy of each classical model. To combat this, half the annotated
A hyperspectral image was used for training while the other was used for testing (as further outlined in 2.2). This decreased the accuracy for each – except for the SVM, discussed later – increasing the legitimacy of the results achieved by each classical model.

A modern algorithm’s (CNN) performance will also be analyzed further.

| Classical Model                                   | Random Data Split | Biopsy Core Data Split |
|---------------------------------------------------|-------------------|------------------------|
| Support Vector Machine (SVM)                      | 76.78%            | 80.90%                 |
| Stochastic Gradient Descent (SGD) [7]             | 95.65%            | 86.13%                 |
| Decision Tree [8]                                 | 94.60%            | 78.33%                 |
| Multilayer Perceptron (MLP)                       | 99.15%            | 88.09%                 |
| Gaussian Naïve Bayes                              | 40.80%            | 40.44%                 |
| Nearest Neighbors [9]                             | 97.70%            | 77.79%                 |

Table 1: Classical Classification Model Results

### 3.1 Support Vector Machine (SVM)

The SVM [5] achieved an accuracy of 81%. A potential explanation for this lower-than-expected accuracy is a poorly created decision boundary for the SVM to base its decisions on. This likely occurred due to the imbalance of annotated lymphocyte and non-lymphocyte data, with more non-lymphocytes being annotated and assumably causing lymphocytes to be missed during the training phase.

### 3.2 Gaussian Naïve Bayes

Regarding Gaussian Naïve Bayes [6] – which performed comparatively worse than other classical models – the results achieved (40.44%) were unexpected. However, preprocessing normalization steps were not taken to allow the data to best accommodate a Gaussian Naïve Bayes model. The lack of preprocessing and the assumptions that the model makes are, together, most likely why the model performed comparatively worse than the other classical models.

### 3.3 Multilayer Perceptron (MLP)

The MLP[1] outperformed all other classical models tested, achieving an accuracy of 88.09% with the split data. This comparatively higher performance likely came from the model’s default parameters being inherently more effective for the analyzed datasets compared to the other classical models.
3.4 Deep Learning

Using Keras[10], we constructed a convolutional neural network model with the structure displayed in Figure 5 – pictured below. This network took hyperspectral tiles of size 17x17 pixels as input and used all 394 channels correlating to hyperspectral bands, which surrounded either lymphocyte or non-lymphocyte pixels. This input was passed through two convolutional layers with kernels of size 3x3, both wrapped in max-pooling layers. From there, the data is flattened and passed to a 500-node dense layer with a ReLU activation function. Finally, the data is passed to a single node dense layer with a sigmoid activation function to predict a binary output of where the input contains a lymphocyte.

To compose the model (Figure 5), we used a 70:30 train-test split and binary cross-entropy loss function. After initially training for three epochs, we achieved an accuracy of 99.7% and a total loss of 0.03.

With an appropriate data split, as described in section 2.2, applied to achieve accurate accuracy results, the CNN performed with a validation accuracy of 92.47% after two epochs using the methods described above. Further epochs resulted in overfitting.

4 Challenges

4.1 Addressing Random-Split Data

When running our data through our first model – an SVM – the model initially returned an accuracy of 99.7%. While this high accuracy is usually something to aim for, no extensive tests were done to tune hyperparameters, and an SVM achieving significantly high results on this dataset was unexpected. Analysis found that data was not being split across biopsy samples; instead, similar pixels, due to their proximity within a sample, were likely being
trained and tested, allowing the model to frequently test on similar data from training. Therefore, the model was overfitting based on training data rather than generalizing the features of lymphocytes from different patients. Because of this, the model was not generalizing the features of a lymphocyte.

 Appropriately splitting the data to have a section of the hyperspectral image dedicated to testing, and another dedicated to training, fixed this issue. This was done by creating the data split described in section 2.2. Splitting our data in half allowed us to have separate subgroups of biopsies in the testing and training data. This meant that, when testing, the model would not have already seen adjacent lymphocytes on the same biopsy. This fix produced realistic accuracies from the SVM model, showing that our assumption of the random-splits influence was correct.

### 4.2 Crowd Annotation

Another challenge that occurred throughout this experiment was combining and using data that was generated by multiple researchers. Having multiple researchers annotate different sections of the data caused the state of the annotations on the cancer biopsies to not be well known by any single researcher. While this problem was minimized by the ground truth provided by the professional pathologist, annotations made for training and testing data were not fully understood. To solve this, figure 3, was created to visualize how our data was distributed.

Running a K-means algorithm on the annotations to cluster them by biopsy core and graphing the data on top of the biopsies created a way for our team to visualize lymphocyte and non-lymphocyte annotations. The specifics of how the data visualization was solved are explained in section 2.1. Our data visualization solved the issue of visualizing annotated data as a sanity check. On top of that, it showed that our data was properly spread out for non-biased results, as well as catching incorrectly annotated data. In fact, one of the biopsies was found to be incorrectly annotated, throwing off accuracy results in the initial stages of testing. Without this visual aid, that data would have continued adversely affecting model accuracy.

### 5 Conclusion

We describe a way to accurately identify lymphocytes within a hyperspectral image containing ovarian biopsy cores of human tissue samples as a potential preprocessing step to classify cellular regions more accurately, isolating lymphocytes – which are frequently abundant throughout biopsy cores – from other tissue types. Both classical models and a modern CNN were analyzed to investigate which method produced the highest results. From these tests, the CNN was proven to be the most effective. Future studies for improving the accuracy of region identification within hyperspectral images of biopsy cores will likely assess the effectiveness of changing hyper-parameters for model tuning.
References

[1] “1.17. Neural Network Models (Supervised).” Scikit, https://scikit-learn.org/stable/modules/neural_networks_supervised.html.

[2] Pedregosa, Fabian, et al. “Scikit-Learn: Machine Learning in Python.” Journal of Machine Learning Research, vol. 12, no. 85, 2022, pp. 2825–30, jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

[3] Pillow, https://pillow.readthedocs.io/en/stable/.

[4] “Welcome to Spectral Python (Spy)¶.” Welcome to Spectral Python (SPy) - Spectral Python 0.21 Documentation, https://www.spectralpython.net/.

[5] “1.4. Support Vector Machines.” Scikit, https://scikit-learn.org/stable/modules/svm.html.

[6] “1.9. Naive Bayes.” Scikit, https://scikit-learn.org/stable/modules/naive_bayes.html.

[7] “1.5. Stochastic Gradient Descent.” Scikit, https://scikit-learn.org/stable/modules/sgd.html.

[8] “1.10. Decision Trees.” Scikit, https://scikit-learn.org/stable/modules/tree.html.

[9] “1.6. Nearest Neighbors.” Scikit, https://scikit-learn.org/stable/modules/neighbors.html.

[10] Chollet, Francois and others, “Keras.” GitHub, https://github.com/fchollet/keras

[11] Berisha, Sebastian, et al. “Deep Learning for FTIR Histology: Leveraging Spatial and Spectral Features with Convolutional Neural Networks.” ResearchGate, Royal Society of Chemistry, 20 Dec. 2018, www.researchgate.net/publication/329822061_Deep_learning_for_FTIR_histology_leveraging_spatial_and_spectral_features_with_convolutional_neural_networks.

[12] NN-SVG. (2018). NN-SVG. http://alexlenail.me/NN-SVG/AlexNet.html