# Pathological Lymph Node Classification

Jonathan Booher, Michael Mariscal and Ashwini Ramamoorthy
SUNet ID: { jaustinb, mgm248, ashwinir } @stanford.edu

***Abstract*— Machine learning algorithms have the potential to improve the efficiency and accuracy of breast cancer metastasis detection in whole-slide images of histological lymph node sections. Here, we experiment with Histogram of Gradients, SURF, and color bucket feature extraction methods on whole-slide images provided by the Camelyon Challenge. Support Vector Machines and Ensemble Learning methods were then run on these features with varying and moderate success. We then put forth a deep learning process consisting of preprocessing the images with CLAHE and Otsu, reducing required computational time using selective search, and then performing hierarchical classification on the slides using a residual network.**

## I. INTRODUCTION

Metastases in lymph nodes are one of the most important prognostic factors in breast cancer. This is because the lymph nodes are the first place breast cancer is likely to spread. The pathologic N-stage (pN-stage) classification is used to evaluate whether cancer has spread to regional lymph nodes. The current diagnostic procedure involving manual classification of these metastases, is time consuming and vulnerable to error. We aim to improve on current algorithms for the automated classification of breast cancer metastases in histological lymph node sections. The input to our algorithm is a series of whole-slide images of stained lymph node sections. We use a wide variety of methods, including SVMS, ensemble learning and convolutional neural networks (CNN) to output a predicted lymph node classification.

## II. RELATED WORK

Our project focused on the Camelyon Challenge [1], which currently has 27 posted submissions, each of which details their experiments run and outcomes found. Due to the size of the data, most submissions utilize preprocessing methods such as optimal thresholds to remove white background space [2], or false positive bootstrapping on randomly selected patches [3], so as to focus on only the regions that may contain a tumor. Additionally, due to its strength in classifying images, the vast majority of submissions to the Camelyon challenge employ CNNs (though projects on similar datasets have used Support Vector Machines on relevant features with some success [4]). This suggested to us that CNNs would be the best performing algorithm. Still, teams that attempted preprocessing and CNNs alone achieved only moderate success [5]. Instead, more success is seen when CNNs are first used to classify tissue regions as tumor or not tumor (or generate probability heat maps of that classification [6]), and then features extracted from the output of the CNN are used to train a classifier (i.e. random forest ensemble [2]) which then classifies each slide as negative, itc, micro, or macro.

## III. DATA

We used data from the Camelyon Challenge. The data in the challenge consists of whole-slide images (WSIs) of hematoxylin and eosin (H&E) stained lymph node sections. The dataset contains data from 100 patients, and for each patient we have 5 whole-slide images. Each slide has a label indicating if it had no metastases ('negative'), isolated tumor cells ('itc'), micro metastases ('micro') or macro metastases ('macro'). The classes are imbalanced and 62.6% of slides have the 'negative' class label.

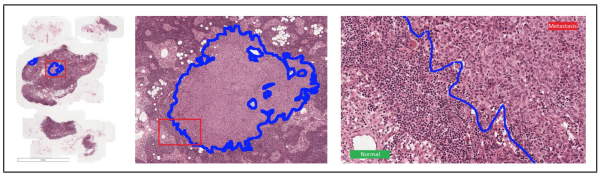

Fig. 1. Example of slide with metastases

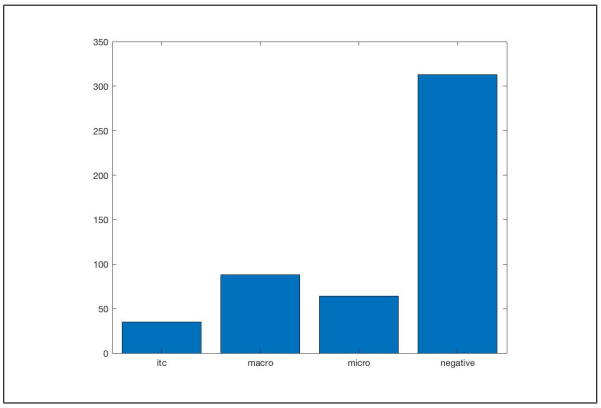


Fig. 2. Class imbalance in data

Whole-slide scanners are used to digitize glass slides containing the lymph node sections at a high resolution (160nm per pixel). This results in an image that is approximately 200000x100000 pixels (56 GB). The WSIs were stored in a multi-resolution pyramid structure, and we worked with the lower resolution, down-sampled version of the image.

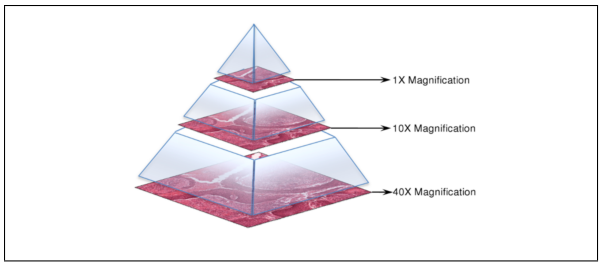


Fig. 3. Multi-resolution pyramid structure

## IV. METHODS

We initially performed binary classification. Slides labelled negative were considered one class, and slides labeled itc, micro or macro were grouped to be the other class. We then performed multi-class classification with four classes.

### A. Pre Processing

For preprocessing, we take the second lowest level of the image and proceed with the following stages:

- Grayscale: Convert the RBG image to grayscale ($0-255$ values)
- Contrast Enhancement: Use the CLAHE algorithm to perform dynamic contrast enhancement. This provides a level of normalization across the different centers where patients were scanned as the images have different white balances.
- Thresholding: We use Otsu's algorithm to filter out pixels that are not part of the lymph node ie sections of the edge of the slide. Note that this stage was only applied in the ResNet pipeline as it can result in loss of parts of the lymph nodes.

### B. Feature Selection

We used feature selection algorithms to derive a feature vector from each image, and used the derived feature vectors to train the classification algorithms.

*1) Histogram of Oriented Gradients (HOG):* HOG is a feature extraction method used for object recognition. The magnitude and orientation of the gradient is found at every pixel. The HOG method then finds a histogram of gradients in 8x8 blocks and provides the concatenated results as a feature vector.
We considered images from the second-lowest resolution level of the pyramid. Since the slides were obtained from different sources, the images came in different sizes. The images were down-sampled to a size of 700x350 and the HOG method was used to extract feature vectors. For a 700x350 image a feature vector of size 130032x1 is obtained.

*2) SURF with Bag of Visual Words:* The Speeded Up Robust Features (SURF) algorithm also relies on a local gradient computation method. It finds features with an approximation of the Laplacian of Gaussian spatial filter. The algorithm selects regions of the image with unique blobs.

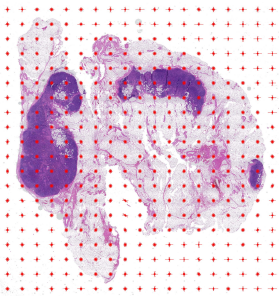

Fig. 4. Example of HOG Features

We then created a bag of visual words. The algorithm groups the features found using SURF into k-mutually exclusive clusters. Each cluster center represents a visual word. We can tune the parameter k, also known as the vocabulary size of the bag. The output for an image is a histogram of visual word occurrences that represent the image. This is used as the feature vector.

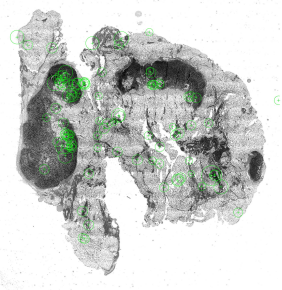

Fig. 5. Example of SURF Features

*3) Occurrences of "Color Buckets":* We implemented a simple initial model by creating buckets for each color, and for each image, counting the number of instances for each color bucket, as implemented in [7]. The number of occurrences found for each bucket was then used as the feature vector on which we trained different classifiers. We again chose the second-lowest resolution on the pyramid.

### C. Classification

We used the following classification methods on the derived feature space.

*1) Support Vector Machines (SVM):* SVMs can be understood intuitively as an algorithm that find the hyperplane which maximizes the margin between the decision boundary and the classes. In practice, data is not separable, so the constraint is relaxed - we have a soft margin classifier. We used SVMs because they can be implemented efficiently for high-dimensional data with kernels.

SVMs are inherently binary classifiers. For multi-class classification, we used two different techniques. In the

"one-vs-all" technique, we build a classifier for each class ($|\mathbb{C}|$ classifiers) and choose the class that classifies the test data point with the largest margin. In the "one-vs-one" we build $|\mathbb{C}|(|\mathbb{C}|-1)/2$ classifiers and pick the class that is selected by most classifiers.

*2) Ensemble Learning:* Ensemble learning methods use several weak learners to obtain a better predictive performance than any individual constituent learner. During each round of training, a new weak learner is added to the ensemble and a weighting vector is adjusted to focus on examples that were misclassified in previous rounds.

We used three ensemble methods. AdaBoost uses decision trees as learners. Subspace uses k-nearest neighbor (kNN) classifiers, and is effective when working with a large feature space. RUSBoost is designed to handle data with imbalanced classes. It randomly under-samples the majority class and then performs boosting with trees like AdaBoost.

## V. METHODS: DEEP LEARNING

This method uses a network of *neurons* which each have their own weights and biases like in logistic regression, however non linearities are introduced (in our case through convolutions and pooling) to differentiate from linear regression.

### A. Pre-processing

The ResNet specific preprocessing pipeline consisted of the normal preprocessing used in combination with a region proposal algorithm. This allows us to greatly reduce the dimensionality of the raw pixels.

The region proposal algorithm is an implementation of Selective Search [8]. We first perform a quick and rough segmentation using Felzenszwalb segmentation. From there, we continually combine regions together based on heuristics until the entire image is one region. We then filter the regions produced so that we return the 5 most significant ones which corresponds roughly to how many distinct lymph nodes are on a single slide image.

We then move one level down the pyramid and extract the identified regions from a higher dimensional version of the image. We then warp these images to 512$x$512 so that we can run them through the network.

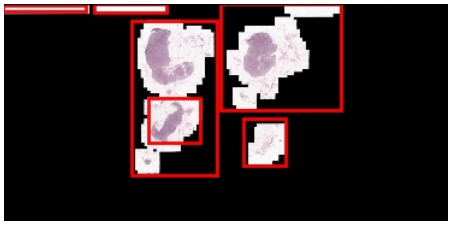

Fig. 6. Example of bounding box from selective search

*1) Selective Search Heuristics:* We are using the heuristics identified in the original paper on this algorithm. We merge regions based on similarities in Color, Texture, Size, and Fit (ie make sure there are no regions surrounded by a single region). We currently provide equal weight to each heuristic:

- Color: Ensure that we group similarly colored regions together as they are likely logical segments.
- Texture: Ensure that we group similarly textured regions as they are likely logical segments.
- Size: Group small regions together as the image is likely not going to have many logical segments.
- Fit: Ensure that no holes are created, ie a segment that completely surrounds another segment.

### B. Classification

We first ran the neural network on binary classification. To do this, we feed the 5 regions identified by the preprocessing through the network as one example with four dimensions $5,512,512,3$. In this way, we can think about the network as operating on $3D$ data. To predict and classify each example, we take the argmax over the class dimension of the network output.

*1) Network Design:* The network used for classification is based of of the Resnet101 architecture. It consists of a series of 'residual blocks' which contain convolutions, pooling, and a skip connection which adds the input of the block to its output. There are a total of 101 layers in this network.

We take the regions propose during preprocessing and run them each independently through the residual network and then add their activations together. We then take this 'pooled' output and run it through two fully connected layers the first of which is to 512 hidden units and the second is to the number of classes. In this way, we can capture the relations between the different regions that were proposed.

## VI. RESULTS

TABLE I
BINARY CLASSIFICATION: ACCURACY

| Feature extraction | Model | Accuracy[1] |
|---|---|---|
| HOG | SVM (Polynomial kernel, order 2) | 0.625 |
| Raw Pixels | ResNet | 0.75 |

[1]The average accuracy with 10-fold cross validation is reported for SVM. The test accuracy of the ResNet is reported.

TABLE II
MULTICLASS CLASSIFICATION: ACCURACY

| Feature extraction | Model | Accuracy[2] |
|---|---|---|
| Color Buckets | SVM (linear) | 0.624 |
| Color Buckets | Subspace Ensemble Learning | 0.442 |
| HOG | SVM, Polynomial kernel (order 2) | 0.6375 |
| HOG | Adaboost, 50 learning cycles | 0.6175 |
| SURF | SVM, Polynomial kernel (order 2) | 0.42 |
| SURF | Subspace | 0.6267 |

### A. Color Buckets

We implemented SVMs with a linear kernel on our color bucket feature vector, and achieved a surprisingly high accuracy of 0.624. We also ran Subspace Ensemble Learning on the color bucket feature vector, which achieved a low training accuracy of 0.442.

TABLE III
COLOR BUCKET SUPPORT VECTOR MACHINE CONFUSION MATRIX

| | Negative | ITC | Micro | Macro |
|---|---|---|---|---|
| Negative | 63 | 0 | 0 | 0 |
| ITC | 6 | 0 | 0 | 0 |
| Micro | 13 | 0 | 0 | 0 |
| Macro | 18 | 0 | 0 | 0 |

A closer look at the confusion matrix for this method shows the algorithm is classifying nearly every image as 0 (negative), which accounts for the majority of the images. As a result, the algorithm is clearly not effective. Though this was our most obvious example, we observed problems resulting from class imbalance on most of our algorithms.

To deal with class imbalance, one approach we took was to find class with the lowest number of examples (class 1 (itc)) and only include that many examples for each class in our training data. This resulted in a significantly lower test accuracy (.260), which better reflects the effectiveness of this approach.

TABLE IV
COLOR BUCKET SUPPORT VECTOR MACHINE CONFUSION MATRIX (BALANCED)

| | Negative | ITC | Micro | Macro |
|---|---|---|---|---|
| Negative | 15 | 16 | 19 | 15 |
| ITC | 3 | 1 | 3 | 2 |
| Micro | 2 | 0 | 3 | 2 |
| Macro | 4 | 6 | 2 | 7 |

[2]The accuracies reported for Color Buckets and HOG feature extraction are average accuracies with 10-fold cross validation. The accuracy on the dev set is reported for SURF.

### B. HOG

We implemented SVMs on the feature set obtained using HOG and tried different kernels. We also implemented AdaBoost, RUSBoost and Subspace ensemble methods. SVMs, AdaBoost and Subspace all suffered from the class-imbalance problem described with color buckets. We implemented RUSBoost and trained it for 1000 iterations. The algorithm had a low test accuracy of 0.23.

TABLE V
RUSBOOST ON HOG FEATURES: CONFUSION MATRIX

| | Negative | ITC | Micro | Macro |
|---|---|---|---|---|
| Negative | 10 | 24 | 18 | 10 |
| ITC | 2 | 0 | 3 | 3 |
| Micro | 0 | 4 | 8 | 0 |
| Macro | 4 | 2 | 7 | 5 |

To provide a reference for comparison with the CNN, we performed binary classification on the HOG feature space with the polynomial kernel SVM. The CNN outperforms the SVM model by a large margin.

### C. SURF with Bag of Words

We first used the default vocabulary size of 500 and formed a bag of words using SURF features. SVMs, AdaBoost and Subspace we susceptible to the class-imbalance problem again. RUSBoost resulted in a more balanced confusion matrix but had poor classification accuracy. We then ran the algorithms with a larger vocabulary size of 5000. It resulted in marginal improvements of accuracy.

### D. Deep Learning

We implemented several different training strategies for training the ResNet. All of them uses the learning rate and momentum schedule as follows: initially have a learning rate f $10e-3$ and decay that by a factor of 10 every 500 epochs. For momentum, begin at 0.9 and decay that continually to reach 0.1 after 1000 epochs.

Using these parameters, we trained several classifiers with a couple different methods to compare their effectiveness.

*4-Class Classification:* We tried training a 4-class classifier on the full training set. This classifier suffered from the class imbalance similarly to what was seen with the Color Buckets SVM.

We also tried training this classifier on randomly balanced training sets and then on the full training set with a low learning rate and momentum. This improved the problem of the class imbalance.

This method was able to reach an accuracy of 46% on the ful

TABLE VI

4-CLASS CLASSIFICATION CONFUSION MATRIX (RESNET)

| | Negative | ITC | Micro | Macro |
|---|---|---|---|---|
| Negative | 26 | 3 | 21 | 20 |
| ITC | 2 | 0 | 0 | 0 |
| Micro | 3 | 0 | 4 | 3 |
| Macro | 2 | 0 | 0 | 16 |

*Hierarchical Classification:* We tried a hierarchical model in which we train two classifiers. One to perform binary classification between *Normal* and a derived class *Metastasis* that corresponds to some for of metastasis. We then take the slides classified as *Metastasis* and run them through another classifier which is trained to distinguish between the types of metastases: *itc*, *micro*, and *macro*. This approach did not fall victim to the class imbalance problem. We trained the binary classifier on the full 400 training examples but with the classes collapsed into 2. We then trained the second classifier on a 121 example subset of that training set where all examples were from one of the metastasis classes.

TABLE VII

POSITIVE METASTASIS CLASSIFICATION CONFUSION MATRIX (RESNET)

| | ITC | Micro | Macro |
|---|---|---|---|
| ITC | 5 | 0 | 0 |
| Micro | 1 | 8 | 2 |
| Macro | 1 | 3 | 10 |

TABLE VIII

POSITIVE METASTASIS CLASSIFICATION CONFUSION MATRIX (RESNET)

| | Negative | Positive |
|---|---|---|
| Negative | 55 | 16 |
| Positive | 10 | 20 |

These results combine to produce a model that is 57% accurate at predicting the class of the slide.

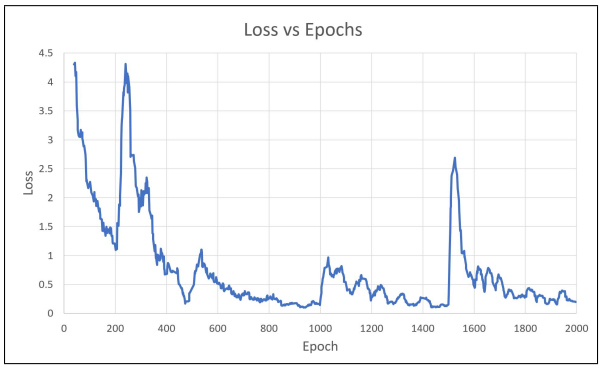


Fig. 7. ResNet Loss 4-Class Over Time

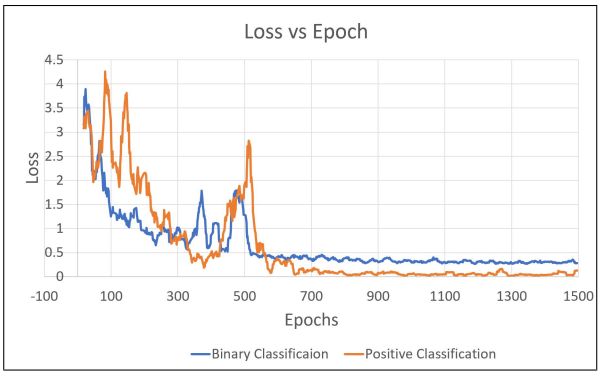


Fig. 8. ResNet Loss Binary and Positive Classifications

## VII. CONCLUSION / FUTURE WORK

Altogether, here we experimented with a variety of feature extraction methods and ran traditional classifiers on these features as well as performed deep learning experiments on the Camelyon Challenge data. After dealing with balancing issues in the dataset, we find deep learning methods to be the most promising in classifying the data. Given the success of deep learning methods in the classification of images, this result is as expected. Overall, as also seen in the most successful attempts at classifying the Camelyon Challenge dataset, our most successful method was hierarchical classification using a residual neural network.

As a result, we would plan to focus future work on residual networks using hierarchical classification. Specifically, our performance on the first step of this method (classifying slides as either containing metastases or not) does not perform as well as those seen in previous Camleyon Challenge submissions [2]. This difference was likely in large part the result of our lack of time and computational resources, which prevented us from moving lower than one step down the pyramid slides (since each step down leads to a significantly larger image). Therefore, if given more time and resources, our next experiment would be to move further down the pyramid.

Additionally, we were unable to test performance of our selective search algorithm and the bounding boxes it outputted. Though difficult to test, our methods depend largely on the performance of this algorithm. Consequently, if this project were pursued further, we suggest manually labeling bounding boxes in the slides, and comparing performance on selective search to the hand labeled data.

## CONTRIBUTIONS

Jonathan implemented the selective search pre-processing and analysis with convolutional neural networks. Michael implemented the ”Color buckets” feature extraction and ran SVMs and ensemble classifiers on the features. Ashwini implemented HOG feature extraction and SURF with bag of visual words and ran SVMs and ensemble classifiers on the features.

## REFERENCES


[1] CAMELYON Challenge: https://camelyon17.grand-challenge.org/home/

[2] Liu, Wei, Yaohua Wang, and Yu Chen. A Hierarchical and Ensemble Framework for Patiences PN-Stage Prediction. Camelyon Challenge, September 26, 2017.

[3] Zanjani, F. Ghazvinian, S. Zinger, and P.H.N de With. Automated Detection and Clasification of Cancer Metastases in Whole-Slide Histopathology Images Using Deep Learning. Camelyon Challenge, 2017.

[4] Akay, Mehmet Fatih. Support Vector Machines Combined with Feature Selection for Breast Cancer Diagnosis. Expert Systems with Applications 36, no. 2, Part 2 (March 1, 2009): 324047.

[5] Moorthy, Arjun, Arun Moorthy, Sujay Nair, and Suraj Nair. Camelyon 2017: Expanding Convolutional Fliters for Robust Metastasis Detection in Lympth Nodes. Camelyon Challenge, 2017.

[6] Fukuta, Keisuke, Daisuke Komura, Tatsuya Harada, and Shumpei Iskikawa. Breast Cancer Using Deep Texture Representation. Camelyon Challenge, 2017.

[7] J. Frost, T. Geisler, A. Mahajan. Monitoring Illegal Fishing through Image Classification, CS229, 2016.

[8] K.E.A. van de Sande, J.R.R. Uijlings, T. Gevers, and A.W.M. Smeulders. Segmentation as Selective Search for Object Recognition ICCV, 2011.