Classifying Physiological States of *Polytrichum* Moss Based on Digital Images Using Machine Learning

Yuan Gao

March 21, 2018

Abstract

Mosses are widespread vegetations on the ground layer level in boreal forests. They play important roles in productivity, soil hydroclimate regulation and nutrient cycling in the ecosystem. The *Polytrichum* mosses are desiccation-tolerant and have two physiological states: a hydrated state and a desiccated state. The physiological features and growth rates of mosses differ in different states. Monitoring the physiological states of *Polytrichum* moss using near-surface remote sensing will be helpful in predicting the growth of mosses and assessing the vegetation condition in boreal forests. The initiative of this project is to classify the physiological states of the mosses based on digital images of moss canopies. In this project, we took images of moss canopies in fields and used OpenCV library to extract attributes that quantify the color and the structure of mosses from images. We then compiled a dataset with extracted attributes and use Weka Machine Learning Library to find ideal machine learning algorithms to do classification. The results showed that kNN classification algorithm had the best performance among the tested algorithms. The trained kNN model was used to predict images in the mixed state in multiple scales. The predictions were mapped back to the original images to compare with manual classifications of canopy images. On average, 66.4% of the area was predicted correctly. The median of this number was 74.1%. Overall, this model could provide a reasonable prediction of the physiological state of moss in images.

Contents

List of Figures

List of Tables

1 Introduction

Mosses are small flowerless green plants that can be found widely in nature. They are major components of plantations in many ecosystems. The biomass of mosses may even greater than other plantations in wet habitats such as fens and bogs [1]. Biomass measures the total mass of all living organisms within the species community, and is also an important measurement of species richness. In boreal forests, moss populations do not have a large biomass. However, they still have important influences on the environment, including productivity and thermal environment of the soil, nutrient cycling in the ecosystem and even attach influence on other plants in the environment [9]. The *Polytrichum* genus, also known as hair-cap moss, is commonly found in moist, shaded and cool areas in fields, forests or woodlands [8]. One of the common habitats of hair-cap moss species is boreal forests. Touw and Rubers's work tells that the species *Polytrichum formosum* grows in temperate zones of Northern Hemisphere, as well as in Africa and New Zealand. It is commonly found in coniferous and deciduous forests [14]. *Polytrichum juniperinum* is widely spread in North America. This species commonly distributed from the alpine zone in the New England mountains to dry scrub oak forests [2]. Like other moss species, *Polytrichum* moss also has a significant influence on the local ecosystem. Besides contributing to the nutrient cycling and regulating soil climate, they can impact regeneration of other plant species. In boreal and temperate coniferous forests, the presence of *Polytrichum* moss can promote the germination of white spruce seeds [10], but reduce seedling growth of the *Pinus sylvestris* tree species [13].

The *Polytrichum* genus is desiccation tolerant. It can withstand drought-like conditions and return to normal growth state after rehydration. The physiological states of mosses changes during this dehydrating and rehydrating process. Under normal hydrated conditions, *Polytrichum* moss should have a dark green appearance and narrow pointed leaves grow along the stem [8]. When the plant is hydrated, leaves spread outward from the stem to approximately 90 degrees. Figure 1 gives an example of the appearance of the moss under hydrated condition. When the plant is dehydrated, leaves curl up and the plant is tubular-looking [8] as shown in Figure 2. The metabolism of the moss also changes along its physiological state. As mosses dry out, their growth is inhibited and the rate of photosynthesis decreases. The metabolism rate goes back to normal when the moss is rehydrated [11]. Hydrated state is the physiological state in which the moss obtains enough water from the environment and maintains its normal productivity and growth rate. In dehydrated state, the moss suffers from a water loss and reduces its metabolism rate and growth rate. Hydrated and desiccated states are the only physiological states appears in the desiccation tolerance *Polytrichum* genus during the desiccation and recovery process.

Due to the significant impact of the moss population on its neighboring environment, knowing the growth condition of the moss can provide valuable information in evaluating the healthiness of the boreal forest ecosystem. For *Polytrichum* moss species, the time that an organism stays at each physiological states is related to its growth rate. In short, having information on the physiological state of the moss population in the ecosystem will assist the assessment of the environment. Traditionally, there are three approaches for monitoring the phenology of vegetation. The first is direct human observation, which is commonly used to observe a limited number of organisms in a small geographic area. The second way is satellite remote sensing, which can monitor the vegetation on a large scale but low resolution. The last method is to use near-surface remote sensing. This method uses radiometric instruments or imaging sensors to monitor the phenological change of vegetation to understand its temporal variation in phenology [12]. For our question, the third approach is the most suitable in terms of the scale. A population of mosses is too large to be manually monitored by human and is too small to use satellite images.

After images or recordings are collected, the next step is image process and analysis. In theory, a person can go through the collected images and observe the change in phenology of the canopy. However, this is not applicable in reality since there will be numerous monitoring files to view. A computational program is needed to automate the image analysis process. Since the moss species only has two physiological states, the program should be able to classify a piece of moss canopy into one of the two states. Machine learning is an effective empirical method for doing data-driven classification problems. Machine learning algorithms recognize patterns in a training dataset and apply the pattern relationship to make predictions on a similar dataset. Previous research has demonstrated that machine learning is a useful tool to use when a large number of data are available but the theoretical knowledge is incomplete [6]. In published studies, machine learning has been using in predicting the classes of vegetation using remote sensing data [3].

Inspired by the method used in previous research, I developed a program to identify the physiological states of hair-cap moss population using near-surface sensing images of the canopy.

2 Related Work

2.1 Near-surface Remote Sensing

Previous studies have applied the near-surface remote sensing technique to monitor the changes in leaf phenology in deciduous broadleaf forest [4] and to record spatial and temporal variation in the phenology of forest canopies [5]. A study on mosses in the Antarctic polar region used a remote sensing imaging method to access the fitness of moss-beds [7]. In a study that utilized near-surface remote sensing to study the spatial and temporal variation in canopy phenology, researchers set up networked digital webcams to

Figure 1: An image of a patch of hydrated *Polytrichum* moss, outside Cooperstown, NY.

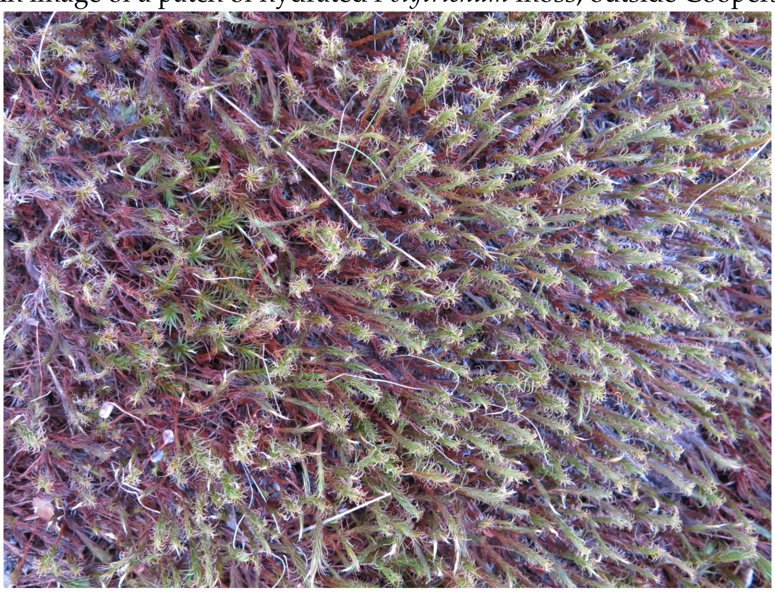

Figure 2: An image of a patch of dehydrated *Polytrichum* moss, outside Cooperstown, NY.

monitor the top of the tree canopies during 12:00 to 14:00 every day throughout the year. Then they used a MATLAB script to analyze the digital image files. Based on the color information extracted from the images, researchers generated the seasonal changes in the canopy phenology [12].

2.2 Machine Learning Applications

In previous research, machine learning has been used to accomplish supervised classification of remote sensing data. Predicting the land cover or vegetation classes are typical questions of this domain. Machine learning algorithms that are commonly used included random forest (RF), support vector machine (SVM) and artificial neural networks (ANN) in these studies. Naive Bayes (NB) and k-nearest neighbors (kNN) were also applied in geological mapping using remote sensing data. In these studies, spectral reflectance imagery served as inputs and manually classified classes served as training data. The classification process was separated into three main stages: pre-processing data, training machine learning model and evaluating predictions [3].

3 Project Design

3.1 Planning

Based on the research in section 1, near-surface remote sensing approach was ideal for the problem. However, given the scale of this project, it was not applicable to set up webcams in the field and conduct actual remote sensing on moss canopies. To simplify the problem, I used a set of near-surface images of moss canopies as the data source instead of remote sensing recordings. Since hair-cap moss species grows abundant within the New York State, I was able to collect images of moss canopies locally. Initially, the plan of this project was to produce a strategy for classifying each image of moss canopy into purely binary states, either hydrated or desiccated. Thus, in the initial image collection stage, the number of images in the mixed state was significantly lower than the number of images in hydrated or desiccated states. However, as the project moved forward, another piece was added to the original plan, mapping the physiological state of the canopies that were in mixed states. Since one patch of mosses can contain both hydrated and desiccated states, it will be more useful to be able to identify the exact locations of hydrated (or desiccated) mosses under the mixed state. Thus I decided to include the mixed state into the classification problem.

Term	Aims/Outcomes
Spring	Background research
	Topic selection
Fall	Collected moss canopy images
	Explored image processing methods for extracting attributes
	Conducted preliminary trial using extracted attributes
Winter	Further explored image processing tools and determined attributes to be used in classification
	Process images and compile the training dataset
	Evaluated the performance of different algorithms
	Classified mixed images manually
	Wrote evaluation pipeline program

Table 1: A brief timeline of the completion time of each part of the design.

3.2 Timeline

This project was completed in three terms, from Spring 2017 to Winter 2018. The execution of the project can be divided into the following steps:

- Data Collection and preprocessing
- Model training
- Prediction evaluating

See the timeline of this project in Table 1.

4 Methods

4.1 Image Collection

An image set of moss canopies was required in this project. Since no such image set is publicly available, all images were collected by ourselves. A regular digital camera was used to take images of mosses in fields. Images of hair-cap mosses outside Cooperstown, NY and in Peebles Island State Park, NY in fall. In total, 446 images were collected from the two sites. Among these images, 196 are in hydrated states, 200 are in desiccated states and 50 are in mixed states. Each image is taken 25 cm above the ground surface and covered an area approximately $24cm \times 18cm$. Two people were needed during the image collecting process. When one person took the image, another person held a cardboard as a sunshade to make sure the lighting of the images was consistent.

4.2 Attribute Selection

After obtaining the image set, we then came up with a list of attributes that are distinguishable between the two physiological states to serve as descriptors. The differences between the hydrated state and the desiccated state are easy to tell by human eyes. Desiccated mosses have a brownish color and their leaves all curl up attaching to the stem (Figure 2). Hydrated mosses are green with star-shaped leaves (Figure 1). By examination, the color of the canopy, number of visible shoots and the structure of shoots are the most distinguishable features. We wanted to quantify these features in a computational way. OpenCV, a library provides functions on computer vision, and ImageJ, an open source image processing program, were used for extracting quantifiable measurements from images.

ImageJ was first used to experiment various image processing algorithms. Since ImageJ has a GUI that provides good visualizations of the algorithm output after each run and the visual differences between two physiological states are distinctive in human eyes, ImageJ was the ideal program to use to evaluate the appropriateness of each image processing program. The immediate visualization feedback helped me to adjust the parameters used and determine the choice of algorithms. After a series of experimentations, I selected the following attributes to be the descriptors of the canopy image: number of shoots, the average size of shoots, the average circularity of shoots, the average aspect ratio of shoots and percentage of green area present in an image.

The following are the detailed methods used to extract the above attributes from an image:

- **percentage of green area coverage:** The image is first converted from RGB (Red, Green, Blue) color space to HSV (Hue, Saturation, Value) color space. Unlike RGB color model, HSV color model is less impacted by noises caused by different lighting condition, as shades and tones are separated from the color it resembles in HSV model. Hue channel represents the pure color information of the image (Figure 3) The value of hue in HSV ranges from 0 to 360 °. In ImageJ, the range of green is converted approximately from 63 to 96. Here, a green pixel was defined as a pixel that had a hue value that lies within this green value range. The number of green pixels was computed. Finally, the percentage of the green area in the image was calculated.

- **number of shoots:** In moss canopies, shoots of mosses usually cluster together. However, particle counting image processing algorithms generally work ideally on images without overlapping between individual particles. Thus, I first ran a watershed algorithm to segment the overlapping particles. Watershed is an algorithm used to perform segmentation of grayscale images. The idea of the algorithm simulates water flooding. It considers the image as a landscape and places water in every regional minimum of its relief. When the level of water rises, there must be a point that different water sources meet each other. That meeting point is the place to put the dam or the place to do the segmentation. I then used the built-in function in ImageJ called "Analyze Particles" to detect regions that belong to the same entity, individual shoot structures in this case. The number of such particles was computed after running the algorithm. Figure 4 is an example of the output after running Analyzing Particles.

- **structure of shoots:** The shape and structure of shoots were described using their area, circularity and aspect ratio. These measurements were also calculated by the function Analyzing Particles. The function calculated the measurements for each particle that was identified in the image. I then took the average value of all particles to represent the entire image.
  - Area: describe size of the shoot
    $Area =$ area of selection in square pixels
  - Circularity: describe the integrity of the shape of the shoot
    $Circularity = 4\pi \frac{area}{perimeter^2}$
  - Aspect ratio: describe if the shoot is round or elongated
    $AspectRatio = \frac{majoraxis}{minoraxis}$

  Here are other functions that I have experimented in the preliminary trial, but decided to discard in the final program:

- **dominant color in the image** The average RGB value of the image extracted using k-means color clustering algorithm. K-means clustering groups each of the n data points into k separate clusters. Each data point is assigned to a cluster with the nearest mean. The mean of each cluster is its centroid. In this case, the k-means clustering was used to find the mean RGB color of the image. However, the k-means clustering algorithm took a long running time to generate the results. Regarding the efficiency of the entire program, this attribute was not included in the final set of attributes. Instead, the percentage of green area coverage was used to include color information in the attribute set.

4.3 Machine Learning Algorithm Selection

Weka Machine Learning Library was used to train various models and determine which algorithm was the most suitable to accomplish the classification. The Weka library contains implemented machine learning algorithms and a collection of tools for selecting attributes and comparing algorithms. Weka Library provided different types of implemented classifiers, including trees, rules, Bayes, lazy and functions, which were all

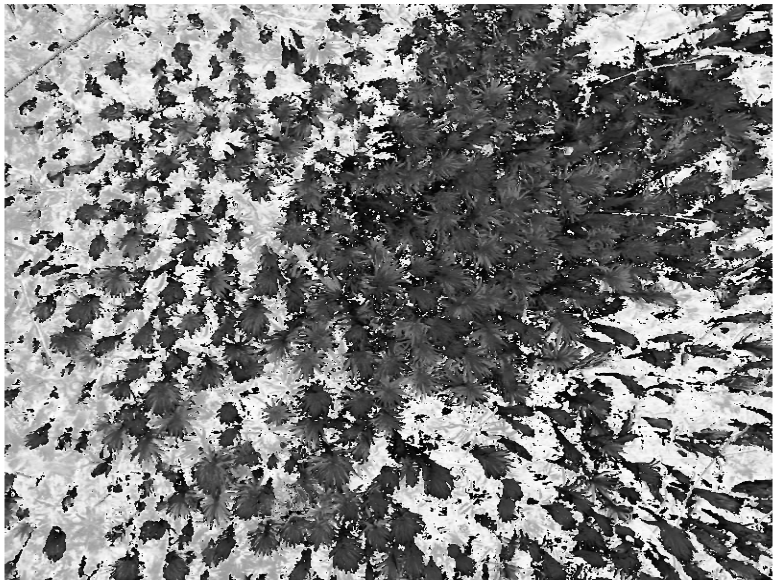

Figure 3: An image of the hue channel of a moss canopy image that contains both hydrated and desiccated areas.

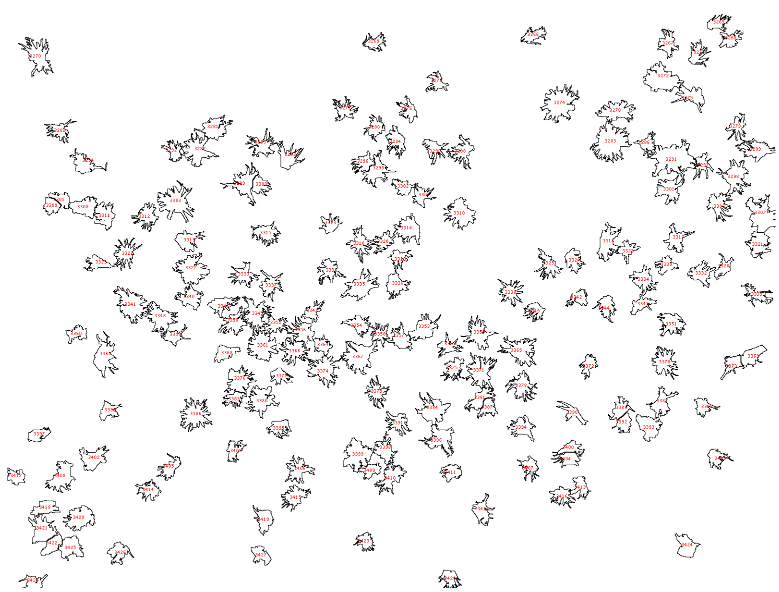

Figure 4: Sample result of analyzing particles of a hydrated moss canopy image.

Algorithms	Percent Correct Classification	Significance (v/ /*)
ZeroR	33.56	
OneR	73.17	v
kNN	95.25	v
Logistic	90.25	v
J48	92.45	v
JRip	92.10	v
Naive Bayes	88.54	v

Table 2: Results of paired t-test on classification accuracy (10-folds cross-validation) of selected algorithms with ZeroR as the benchmark.

commonly used algorithm types. At least one algorithm was chosen from each type to be a candidate. The selected classification algorithms were OneR, J48, JRip, Naive Bayes, kNN and Logistic Regression. All of which were commonly used machine learning algorithms that can be applied to almost any data classification problem.

5 Results

Weka's experimenter was used to evaluate the performance of the chosen algorithms. The input dataset in this experiment contained extracted image attributes mentioned in the subsection 4.2. As mentioned in subsection 4.1, the class of the instances in this dataset was not balanced. There were around 200 images in each of the desiccated and hydrated states, but there were only 50 images for the mixed state. To balance the number of instance in each class, I up-sampled the minority class, i.e. mixed state, to contain the same number of instances as the majority classes.

The 10-fold cross-validation was used to check the accuracy of the classification algorithms. Then I conducted two-tailed paired t-tests the prediction to evaluate the performance of the algorithms. In the first test, the classification accuracies of the selected algorithms were compared with the classification accuracy of ZeroR. ZeroR classifier predicts the majority class and ignores predictors. The performance of Zero classifier is commonly used as a benchmark for evaluating the prediction of other classification algorithms. The results of this experiment are displayed in Table 2. The significance level is set to be 0.05. Significance column uses annotation "v" to indicate a specific result is statistically better or "*" to indicate if it is statistically worse.

According to the result of the t-test, all selected algorithms can predict statistically significantly better than zeroR. Thus I ran another experiment to compare the performances of the 6 chosen algorithms. Since k-nearest neighbor had the best performance in the last test, it was used to be the baseline classifier. Again the level of significance was 0.05. The results in Table 3 show that the classification accuracies of the rests

Algorithms	Percent Correct Classification	Significance (v/ /*)
kNN	95.25	
OneR	73.17	*
Logistic	90.25	*
J48	92.45	*
JRip	92.10	*
Naive Bayes	88.54	*

Table 3: Results of paired t-test on classification accuracy (10-folds cross-validation) of selected algorithms with kNN as the benchmark.

Rank	Attributes	Information Gain Value
1	Percentage of Green Area Coverage	0.919
2	Circularity	0.877
3	Average Size of Particles	0.844
4	Number of Particles	0.435
5	Average Aspect Ratio of Particles	0.422

Table 4: Attribute selection output using information gain based feature selection.

of algorithms are statistically significantly worse than kNN.

I also applied information gain based feature selection algorithm to evaluate predictor attributes. This algorithm calculates the information gain for each attribute in the dataset. The result of evaluation ranges from 0 (no information) to 1 (maximum information). Attributes with a higher information gain value make a greater contribution to the classification stage and vice versa. The results showed that percentage of green area coverage contributes the most to the prediction model, followed by the average circularity of particles and the average size of particles. Nevertheless, the information gain values for all attributes were greater than 0.4, indicating all attributes played an important role in the model. Detailed outputs of the attribute evaluator are shown in Table 4.

6 Evaluation

6.1 Manual Classification

For evaluating the performance of the machine learning algorithms, I first classified all images in mixed state manually. I adapted a freehand masking demo of MATLAB to allow the user freehand circle the hydrated regions on the canopy images. Then a binary colored image was generated based on the drawing (Figure 5). On the left is the original image of the moss canopy. After I manually circled the area of mosses that I determined as hydrated, the program automatically generated a binary image that used green to represent the hydrated region and yellow to represent the desiccated region. I classified all 50 mixed images

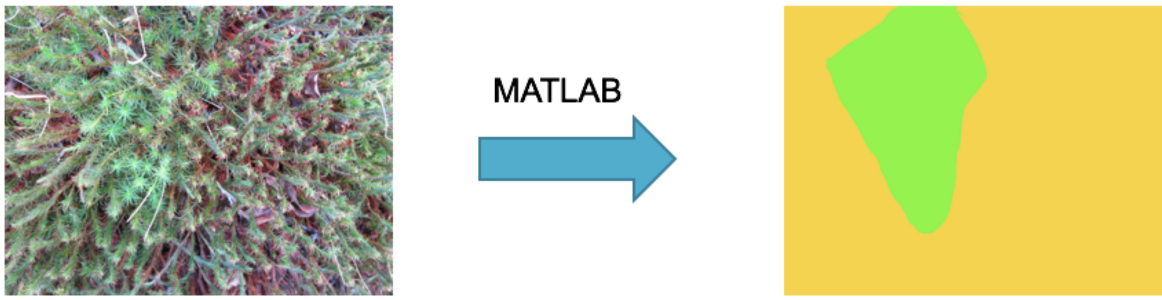

Figure 5: Sample input (left) and output (right) image of the MATLAB program.

in this way and the generated images were used to evaluate the accuracy of the machine learning outputs.

6.2 Machine Learning Classification

To evaluate the accuracy of the model, the results of classification needed to be compared with manual classified outputs. The kNN model trained in section 5 only provided a numerical classification accuracy. To draw a connection between the model and the manual classified images, I built a pipeline to generate binary images that represent the classification outcomes of the kNN model.

All images in mixed states were classified using the trained kNN model. Based on the classification result of the image, the pipeline program decided its next operation: if the prediction was either hydrated or desiccated, then the program generated a blank image with the same dimension as the original image but filled with solid color. If the predicted state was desiccated, the image was filled with yellow. If the predicted state was hydrated, the filled color was green. The color here matched the color used in subsection 6.1. When the model predicted the image to be in the mixed state, the program sliced the current image into four pieces. The sliced images were then processed to obtain values of prediction attributes. Since the size of the image was different from the original image, the Number of Particles attribute was standardized to keep the consistency between the training and test data. The same model was used to predict the class of the newly sliced images. The same operation was done as the last iteration. This slicing and predicting cycle continued until all the images were classified as either hydrated or desiccated. Figure 6 visualizes the workflow of this pipeline. As the sliced images being classified, new images filled with solid yellow or green color were generated. After all of the predictions were done, the generated images were merged back based on their original location index. The final outcome mapped all prediction decision onto the initial moss canopy images.

Figure 7 shows an example of the final image produced by the pipeline and its counterpart manual classified image. Just by eyeballing the two images, one can see the similarity between them. The calculated percentage overlap between the manual classified image and the pipeline produced image was 78.3%, proven the pipeline output was very close to the manual classification result for this image. Further analysis showed that more likely for hydrated mosses to be misclassified as desiccated ones than the other way around. Figure 8 has a high overlap percentage for about 94%. However, there were also cases that the prediction outputs were not ideal. In Figure 9, only 32.8% of the area was predicted correctly. The model failed to identify the majority of the hydrated regions.

Overall, the images generated by the pipeline displayed an ideal consistency with manual classifications. On average, the prediction and the manual output had a 66.4% overlap. The median of the overlapping percentage was 74.1%, higher than the mean value, which indicated the presence of extremely poorly classified images. For all hydrated regions, 34.8% was classified correctly, while 79.5% of the desiccated regions were classified correctly. The result showed the model had a tendency of classifying hydrated mosses into the desiccated state. Although the percentage of overlapping areas was not that high, the model was able to identify the general location of the hydrated mosses. The location of the green region in generated images was very consistent with the green region in manual classification images.

One possible explanation of having a lower classification accuracy in predicting hydrated regions was the bias in doing manual classification. The manual images were generated by freehand drawing on the border of hydrated regions. It was possible that on the edges, some desiccated regions were also circled into the drawing and were classified as hydrated. Maybe if the drawing were done by circling the border of desiccated moss, the results would be different.

7 Future Work

Currently, the manual classification was done by only one person. There might be a potential bias in terms of the criterion of defining the two physiological states. To improve the objectiveness of the classification, each image should be classified by more students. Further analysis can test the performance of the model using either the union of the selected regions or the intersection of all selected regions, and see if this choice makes a difference.

The program already had the ability to classify the physiological states of moss canopies by digital images. In the future, we can extend this program to classify states of mosses in video recordings, so that we will be able to perform actual remote sensing on mosses in the field.

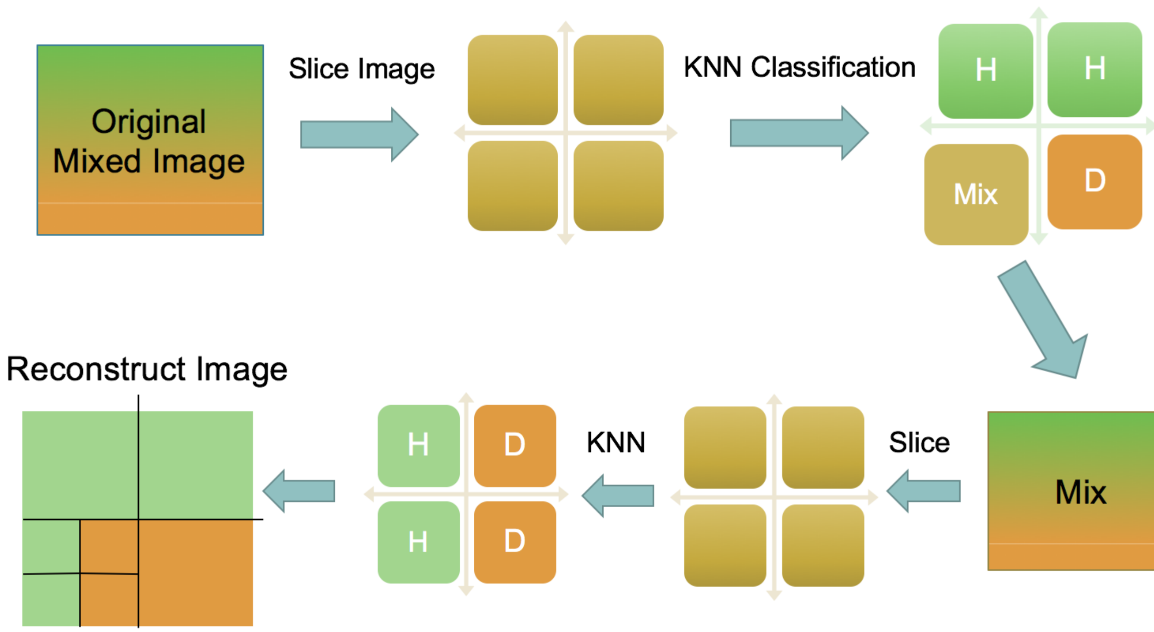

Figure 6: Visualization of the pipeline process.

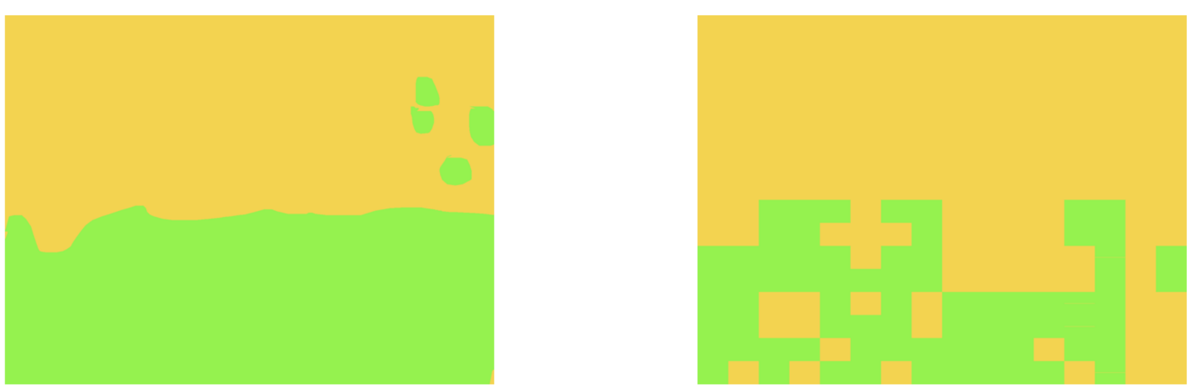

Figure 7: Sample pipeline output 1. Left: manual classification Right: assembled image based on model prediction

Figure 8: Sample pipeline output 2. Left: manual classification Right: assembled image based on model prediction

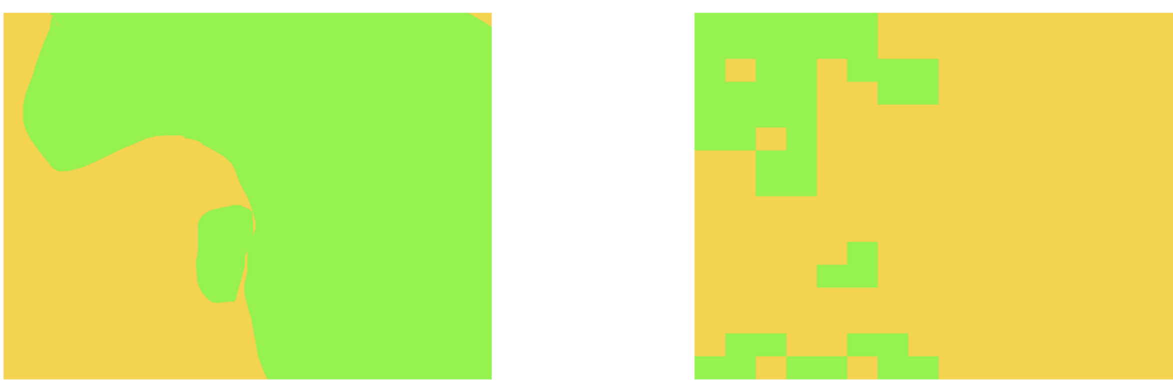

Figure 9: Sample pipeline output 3. Left: manual classification Right: assembled image based on model prediction

References

[1] D. PAULI A. BERGAMINI, M. PEINTINGER, and B. SCHMID. “Relationships between productivity, number of shoots and number of species in bryophytes and vascular plants”. In: *Journal of Ecology* 89 (2001), pp. 920–929.

[2] F. A. Bazzaz, Dominick J. Paolillo, and R. H. Jagels. “Photosynthesis and Respiration of Forest and Alpine Populations of Polytrichum juniperinum”. In: *The Bryologist* 73.3 (1970), p. 579. DOI: 10.2307/3241495.

[3] Matthew J. Cracknell and Anya M. Reading. “Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information”. In: *Computer and Geosciences* 63 (2013).

[4] Dominique Guyon et al. “Near-surface remote sensing observations for monitoring deciduous broadleaf forest species phenology”. In: *2014 IEEE Geoscience and Remote Sensing Symposium* (2014). DOI: 10.1109/igarss.2014.6946950.

[5] Qifei Han. “Remote sensing-based quantification of spatial variation in canopy phenology of four dominant tree species in Europe”. In: *Journal of Applied Remote Sensing* 7.1 (May 2013), p. 073485. DOI: 10.1117/1.jrs.7.073485.

[6] David J. Lary et al. “Machine learning in geosciences and remote sensing”. In: *Geoscience Frontiers* (2016).

[7] Zbynek Malenovsky et al. “Antarctic moss stress assessment based on chlorophyll content and leaf density retrieved from imaging spectroscopy data”. In: *New Phytologist* 208.2 (2015), 608?624. DOI: 10.1111/nph.13524.

[8] Karl B. McKnight. *Common mosses of the Northeast and Appalachians*. Princeton University Press, 2013.

[9] W. C. Oechel and K. Van Cleve. “The Role of Bryophytes in Nutrient Cycling in the Taiga”. In: *Forest Ecosystems in the Alaskan Taiga: A Synthesis of Structure and Function*. Ed. by K. Van Cleve et al. New York, NY: Springer New York, 1986, pp. 121–137.

[10] William C. Parker, Stephen R. Watson, and Daryll W. Cairns. “The role of hair-cap mosses (Polytrichum spp.) in natural regeneration of white spruce (Picea glauca (Moench) Voss)”. In: *Forest Ecology and Management* 92.1-3 (1997), 19?28. DOI: 10.1016/s0378-1127(96)03956-4.

[11] Michael C. F. Proctor, Roberto Ligrone, and Jeffrey G. Duckett. "Desiccation Tolerance in the Moss Polytrichum formosum: Physiological and Fine-structural Changes during Desiccation and Recovery". In: *Annals of Botany* 99.1 (July 2006), pp. 75–93. DOI: 10.1093/aob/mcl246.

[12] Richardson et al. "Near-surface remote sensing of spatial and temporal variation in canopy phenology". In: *Ecological Applications* 19 (), pp. 1417–1428.

[13] Babs Stuiver et al. "The Impact of Moss Species and Biomass on the Growth of Pinus sylvestris Tree Seedlings at Different Precipitation Frequencies". In: *Forests* 5.8 (June 2014), 1931?1951. DOI: 10.3390/f5081931.

[14] M. Van Der Velde et al. "The reproductive biology of Polytrichum formosum: clonal structure and paternity revealed by microsatellites". In: *Molecular Ecology* 10.10 (2001), pp. 2423–2434. DOI: 10.1046/j.0962-1083.2001.01385.x.

Appendices

A Pipeline Implementation

I implemented the pipeline program in Python. OpenCV library was used to process images and extract attributes. Scikit learn machine learning library was used to train the kNN model and make predictions. Image slicer tool in Python was used for splitting images. Here is a code sample from the pipeline program:

```python
def classify_image(model, iteration, test_data):

    dry, wet, mixed = make_prediction(model, test_data)

    gen.generate_image(dry, 'dry', iteration)
    gen.generate_image(wet, 'wet', iteration)

    if len(mixed) > 0:
        new_img_dir = slicer.slice_image(mixed, iteration)
        new_csv_dir = img_processor.batch_processing(new_img_dir, iteration)
        new_data = pd.read_csv(new_csv_dir)
        print("finish iteration: ", iteration)

        next_iter = iteration * 4
        classify_image(model, next_iter, new_data)


''' start the pipeline '''
def pipeline(training_dir, test_dir, filename):
    training_data = pd.read_csv(training_dir)
    model = train_model(training_data)
    iteration = 1

    test_data = pd.read_csv(test_dir + str(iteration) + '.csv')
    classify_image(model, iteration, test_data)

    assembler.assemple()
    evaluator.evalutation_output(filename)
```

Figure 10: Sample code