*k*NN vs. SVM: A Comparison of Algorithms

Dale Hamilton, **Ryan Pacheco**, **Barry Myers**, and **Brendan Peltzer**,
Department of Math and Computer Science, Northwest Nazarene University, Nampa, Idaho

Abstract—Support Vector Machines (SVM) and *k*-Nearest Neighbor (*k*NN) are two common machine learning algorithms. Used for classifying images, the *k*NN and SVM each have strengths and weaknesses. When classifying an image, the SVM creates a hyperplane, dividing the input space between classes and classifying based upon which side of the hyperplane an unclassified object lands when placed in the input space. The *k*NN uses a system of voting to determine which class an unclassified object belongs to, considering the class of the nearest neighbors in the decision space. The SVM is extremely fast, classifying 12 megapixel aerial images in roughly ten seconds as opposed to the *k*NN which takes anywhere from forty to fifty seconds to classify the same image. When classifying, the *k*NN will generally classify accurately; however, it generates several small misclassifications that interfere with final classified image that is outputted. In comparison, the SVM will occasionally misclassify a large object that rarely interferes with the final classified image. While both algorithms yield positive results regarding the accuracy in which they classify the images, the SVM provides significantly better classification accuracy and classification speed than the *k*NN.

Keywords: *k*-nearest neighbor, support vector machine, remote sensing, small unmanned aircraft system, burn extent, biomass consumption

INTRODUCTION

This paper explores whether a SVM or *k*NN classifier provides higher accuracy when mapping effects with very high spatial resolution using small unmanned aircraft system (sUAS) imagery. Support Vector Machine (SVM) and *k*-Nearest Neighbor (*k*NN) are two machine learning algorithms mentioned in the literature that outperformed other algorithms (Yang and Liu 1999) (Zhang and Ma 2008) when mapping wildland fire effects on low resolution satellite imagery (Brewer et al. 2005) (Zammit et al. 2006) (Liu et al. 2006). This study compares the classification accuracy of the two algorithms when mapping wildland fire post-fire effects with very high spatial resolution imagery acquired with a sUAS.

Wildlands provide habitat for around 6.5 million species according to the United Nations Environment Program (Mora et al. 2011). In the United States and elsewhere, wildlands provide resources for energy and material in addition to offering recreational and spiritual opportunities for humans. Wildlands also provide irreplaceable ecosystem services including clean water, nutrient cycling and pollination, in addition to habitat for millions of animals. Large expanses of the wildlands in the U.S. have evolved with fire and depend on periodic wildfires for health and regeneration (Aplet and Wilmer 2010).

Decades of fire suppression have led to the current departure of wildlands from the fire return interval characteristically experienced prior to European settlement. As a result, wildlands in the western U.S. are experiencing a much higher incidence of catastrophic fires (Wildland Fire Leadership Council 2015). Fire impacts millions of hectares of

In: Hood, Sharon; Drury, Stacy; Steelman, Toddi; Steffens, Ron, tech. eds. The fire continuum—preparing for the future of wildland fire: Proceedings of the Fire Continuum Conference. 21-24 May 2018, Missoula, MT. Proc. RMRS-P-78. Fort Collins, CO: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. 358 p.

American wildlands each year, with suppression costs approaching two billion dollars annually (National Interagency Fire Center 2018) (Zhou et al. 2005). High intensity wildland fires contribute to post fire erosion, soil loss, flooding events and loss of timber resources. This results in negative impacts on human communities, wildlife habitat, ecosystem resilience, and recreational opportunities. Additionally, wildland fires across the U.S. claim more lives than any other type of natural disaster, resulting in the loss of ten to twenty firefighters per year (Zhou et al. 2005).

Effective management of wildfire and prescribed fires is a critical dimension of maintaining healthy and sustainable wildlands. A quantitative understanding of the relationships between fuel, fire behavior, and the effects on human development and ecosystems can help land managers develop nimble solutions to U.S. wildfire problems. Remotely sensed imagery is commonly collected to assist in assessing the impact of the fire on the ecosystem (Eidenshink et al. 2007). The knowledge gained from remotely sensed data enables land managers to better understand the effects a fire has had on the landscape and develop a more effective management response facilitating ecosystem recovery and resiliency.

Burn Severity and Extent

The term "wildland fire severity" can refer to many different effects observed through a fire cycle, from how intense an active fire is burning, to the response of the ecosystem to the fire over the subsequent years. This study investigates direct or immediate effects of a fire such as biomass consumption as observed in the days and weeks after the fire is contained (Keeley 2009). Therefore, this study defines burn severity as the measurement of biomass (or fuel) consumption (Key and Benson 2006).

Identification of burned area extent within an image can be achieved by exploiting the spectral separability between burned organic material (black ash and white ash) and unburned vegetation (Hamilton et al. 2017; Lentile et al. 2006). Classifying burn severity can be achieved by separating pixels with black ash (low fuel consumption) from white ash (more complete fuel consumption), relying on the distinct spectral signatures between the two types of ash (Hudak et al. 2013). In forested biomes, low severity fires can also be identified by looking for patches of unburned vegetation within the extent of the fire. If a patch is comprised only of tree crowns, the analysis can infer that the vegetation is a tree which the fire passed under, and classify the pixels as low intensity surface fire (Scott and Reinhardt 2001). If the patch of vegetation contains herbaceous or brush species, then the patch is actually an unburned island within the burned area and can be classified as unburned.

Utilization of sUAS for Image Acquisition

An unmanned aircraft system (UAS) is an aircraft that is not controlled by a pilot on board the aircraft (Calkins 2017). A UAS may either be controlled by a pilot on the ground or by an autonomous flight control systems that executes a previously designated flight path. New advances in small UAS (sUAS) capabilities enable the acquisition of imagery with a spatial resolution of centimeters and temporal resolution of minutes (Laliberte et al. 2010). This hyperspatial imagery, which enables objects to be represented in the image by multiple pixels (Sridharan and Qiu 2013), results in a huge increase in the quantity of data associated with a scene. When sUAS are used for remote sensing, a set of images are taken through the course of a flight. Photogrammetry software such as Pix4D is used to stitch the set of images into a single image or orthomosaic in which perspective distortions from multiple images are resolved (Küng et al. 2011). Additionally, the photogrammetry software uses the latitude and longitude embedded in the image by the global navigation satellite system (GNSS) receiver onboard the sUAS to georeference the orthomosaic, enabling a geographic information system (GIS) to be able to identify the physical location of every object within the georeferenced orthomosaic (Rosnell and Honkavaara 2012).

Mapping Wildland Fire Effects with sUAS

Fire ecology enables managers to study temporary environmental changes by accounting for the pronounced change that wildland fire effects on an ecosystem. The emerging field of ecoinformatics promises to provide the methodologies and tools needed to acquire, analyze, and manage the growing amounts of complex ecological data available from

the immense volume of data available in hyperspatial sUAS imagery. The combination of sUAS and ecoinformatics provides increasing amounts of actionable knowledge regarding wildfire management.

In order to fully utilize high resolution imagery to effectively map burn severity, it is necessary to identify black ash, white ash, surface vegetation and canopy vegetation within hyperspatial orthomosaics generated from imagery captured with sUAS. Toward this end, machine learning and image processing tools were developed which facilitate pixel-based identification of these classes of interest for mapping burn severity from hyperspatial imagery.

ALGORITHM COMPARISON: SVM VERSUS *K*NN

The recent advances in small unmanned aircraft systems (sUAS) technology promise to provide wide availability of hyperspatial imagery to users who previously did not have the ability to generate remotely sensed imagery on their own. The copious amounts of easily obtained data resulting from the acquisition of hyperspatial imagery warrant investigation into development of methods, analytic tools and metrics which enable the extraction of information and knowledge from imagery captured at much higher resolution than was previously possible. The affordability of sUAS is facilitating the dissemination of knowledge previously unattainable from lower resolution data. This paper describes a study investigating which of two machine learning algorithms—a Support Vector Machine (SVM) or *k*-Nearest Neighbor (*k*NN)—will enable mapping of burn severity with higher accuracy (Han et al. 2012; Russell and Norvig 2010).

Support Vector Machine

When classifying an image, the SVM creates a hyperplane, dividing the input space between classes, classifying based upon which side of the hyperplane an unclassified object lands when placed in the input space. The algorithm performs a pixel-based classification, labeling each pixel in the image with the postfire effects class determined by the classifier. SVM classifiers have been successfully used for image classification, including for mapping burn severity from medium resolution satellite imagery (Zammit et al. 2006) (Zhang and Ma 2008) (Liu et al. 2006) (Salimi et al. 2017). Hyperspatial imagery is defined as having a spatial resolution less than one decimeter (Hamilton, 2018). Classification starts with the training of an SVM via user-generated training data.

Training of an SVM starts with the labeling of regions in the image based on user observation of the pixels within the region. When classifying burn extent, user selected training pixels are labeled as to whether they burned or not. For training the SVM for mapping biomass consumption, or the reduction of vegetative fuels, the user labels pixels as to whether they represented black ash indicative of incomplete combustion or white ash resulting from more complete combustion.

As each training pixel is loaded from an image, the pixel values for the red, green and blue bands are placed into elements in a vector, producing a three-dimensional feature space. If necessary, any additional data for the pixel, such as texture, is included into an additional element on the vector. Assuming a binary classification, the vector is labeled with one of two classes, either +1 or -1. Each of the training pixels is placed onto an array X. The class labels for each training pixel are placed into corresponding elements of a parallel array Y. Training the SVM involves identifying a decision boundary, which separates the sets of training pixels on X. The SVM computes this class decision boundary as a hyperplane (Han et al. 2012). Two-dimensional data will be separated by a hyperplane which is shown as a line, as shown in figure 1. As the dimensionality of the data increases to three, the separating hyperplane is represented as a plane. Additional increases in data dimensionality results in separating hyperplanes that are one less dimension than the data. Unfortunately, simply selecting a separating hyperplane may not generalize well, with some candidate hyperplanes lying close to some of the training vectors (fig. 1).

SVM generalizability is improved, reducing the probability of misclassifications, by identifying the optimal separating hyperplane, which is the separating hyperplane that is the furthest from any training vectors, while still correctly separating the training vectors. Twice the distance from the hyperplane to the

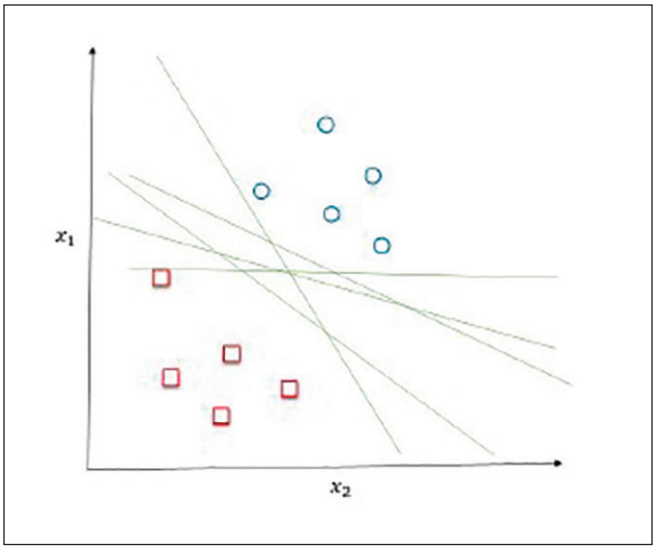

Figure 1–Hyperplanes (shown as green lines) separating classes represented by blue circles and red squares. X_1 and X_2 represent the two dimensions represented in the SVM.

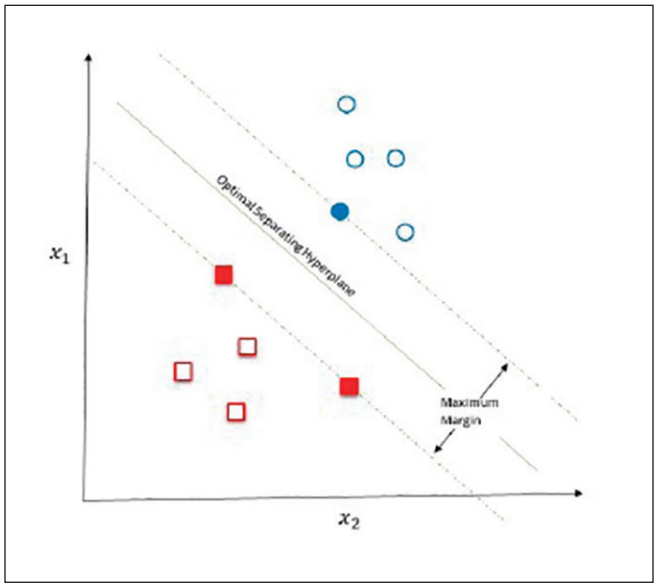

Figure 2–Support Vector Machine showing linearly separating hyperplane. The support vectors shown as solid shapes define the maximal separating margin between the two classes.

nearest training vector is referred to as the margin. The optimal separating hyperplane contains the maximum margin between training vectors of the two classes, resulting in the greatest separation between the classes. Some texts refer to the optimal separating hyperplane as the maximum margin hyperplane in reference to it having the maximum margin (Han et al. 2012). The hyperplanes that are found at the edges of the margin are parallel to the optimal separating hyperplane. The training vectors that lie on the hyperplanes on the margin edges are the support vectors, shown as the solid squares and circles as shown in figure 2. These support vectors on the edge of the optimal hyperplane margin support the placement of the optimal hyperplane. Once the optimal hyperplane has been located, none of the training vectors other than the support vectors need to be retained. SVM is a parametric model due to the need to retain the training vectors which comprise the support vectors. While it's possible that all the training vectors would need to be retained, in practice only a small number of training vectors need to be retained as support vectors, sometimes a small constant times the number of dimensions (Russell and Norviq 2010) (Hamilton, 2018).

Once the optimal hyperplane is located, pixels with unknown class are placed into a vector within the decision space, after which the pixel's class can be determined by calculating which side of the optimal hyperplane the vector lies on. Assuming there exists an optimal hyperplane that completely separates both sets of vectors on X, the classes of data are linearly separable. If the data is nearly linearly separable, with a small number of training pixels being outliers, it is possible to separate the data by allowing a minimal amount of error, resulting in the establishment of a soft margin (Russell and Norvig 2010).

Determination of Optimal Hyperplane Placement

Determining the location of the optimal separating hyperplane and maximizing the margin is an optimization problem for which there are multiple solutions. Separating hyperplanes in an SVM are defined by the equation:

$$\boldsymbol{W} \cdot \boldsymbol{X} + b = 0 \qquad (1)$$

where $\boldsymbol{W}$ is a weight vector, namely $\boldsymbol{W} = \{ w_1, w_2, \ldots w_n \}$; with n is the dimensionality of the decision space (e.g. for a color image, $n = 3$), b is the intercept (Russell and Norviq 2010) or bias (Han et al. 2012) and $\boldsymbol{X}$ is the set of support vectors.

The optimal separating hyperplane can be found by searching the decision space for $\boldsymbol{W}$ and b using gradient descent optimization, searching for parameters that maximize the margin while correctly

classifying the training vectors (Russell and Norviq 2010). Alternately, Equation 1 can be rewritten in an alternative dual representation as:

$$argmax_{\propto} \sum_j \alpha_j - \frac{1}{2}\sum_{j,k} \alpha_j \alpha_k y_j y_k (\boldsymbol{x}_j \cdot \boldsymbol{x}_k) \quad (2)$$

subject to the constraints $\alpha_j \geq 0$ and $\Sigma_j \alpha_j y_j = 0$ where α_j are Lagrangian multipliers. Equation 2 is a constrained convex quadratic equation for which there are software packages that can determine an optimal solution. (Han et al. 2012) (Russell and Norviq 2010). Once the vector α has been calculated, we can derive $\boldsymbol{W}$ from Equation 1 with $\boldsymbol{W} = \Sigma_j \alpha_j x_j$. Because Equation 2 is convex, there is a single global maximum. Additionally, the weights αj associated with each of the training vectors are zero except for the support vectors, which are closest to the optimal separating hyperplane (Russell and Norviq 2010).

Recalling that the class labels associated are represented as +1 and -1, the weight vector $\boldsymbol{W}$ can be adjusted so the hyperplanes on the edges of the margin (which are parallel to the optimal separating hyperplane) are

$$H_a : \mathbf{W} \cdot \mathbf{X}_i + b \geq 1 \text{ for } Y_i = +1 \quad (3)$$

and

$$H_b : \mathbf{W} \cdot \mathbf{X}_i + b \leq 1 \text{ for } Y_i = +1 \quad (4)$$

Any training vector on or above H_a belongs to class +1 while any training vector that falls on or below H_b will belong to class -1. Combining Equations 3 and 4 results in

$$Y_i (\mathbf{W} \cdot \mathbf{X}_i + b) \geq 1, \forall I \quad (5)$$

Any training vectors that lie on either H_a or H_b are the support vectors, denoted in Figure 2 as solid circles and squares (Han et al. 2012).

The support vectors are easily identifiable once the vector $\boldsymbol{\alpha}$ has been calculated, being the subset for which the weights α_j are greater than 0. The optimal separating hyperplane is defined, tuples with unknown label can be labeled with the following equation:

$$h(\boldsymbol{x}) = \text{sign}(\Sigma_j \propto_j y_j (\boldsymbol{x} \cdot \mathbf{X}_j) - b) \quad (6)$$

where $\boldsymbol{x}$ is a vector with unknown class. $h(\boldsymbol{x})$ will return either a class label of -1 or +1, the predicted class of $\boldsymbol{x}$. Only support vectors have $\boldsymbol{\alpha}_j > 0$, therefore the trained SVM can reduce runtime by restricting j so Equation 5 runs on support vectors (Russell and Norviq 2010).

Determining Hyperplane for Training Data That is Not Linearly Separable

The inability to locate an optimal separating hyperplane without error is an indicator that the classes of training vectors are not linearly separable. If the data is nearly linearly separable, with a small number of training pixels being outliers, it is possible to separate the data by allowing a minimal amount of error, resulting in the establishment of a soft margin (Russell and Norviq 2010). While optimizing the separating hyperplane, the individual vector error is calculated for each training vector that is falls on the wrong side of the hyperplane, labeled as ε_i, subject to $\varepsilon_i \geq 0, \forall i$. The misclassification error is the summation of all the individual vector errors for all of the training vectors, giving a misclassification error for the associated separating hyperplane. A parameter, referred to as C, is added allowing tuning of an SVM controlling the weight accorded to the misclassification error, resulting in the term of $C \Sigma_i \varepsilon_i$ being added to the margin width (Cortes and Vapnik 1995) as shown in Equation 7.

$$\text{argmax}^{\propto \Sigma_j \alpha_j - \frac{1}{2}\Sigma_{j,k} \propto_j \propto} (\boldsymbol{x}_j \cdot \boldsymbol{x}_k) + c \Sigma_i \varepsilon_i \quad (7)$$

Both the misclassification error and margin width are optimized together, maximizing the margin while also minimizing misclassification error. The resulting decision function is an optimal separating hyperplane with a soft margin (Cortes and Vapnik 1995). Figure 3 shows an optimal separating hyperplane that best generalizes the decision boundary between linearly inseparable classes.

Data should only be mapped to a higher dimensional decision space if the data is not already linearly separable. Fortunately, it is simple to determine whether the training data is linearly separable. By running all the training data through a trained SVM, the training data is completely linearly separable if the SVM labels all the training pixels with the same label as specified by the user. In the event that the user and SVM labels are not in agreement for all the training pixels, it is possible to calculate the degree of linear separability of the training data as

$$\sigma = \frac{Correctly\ predicted\ support\ vectors}{|x|} * 100 \quad (8)$$

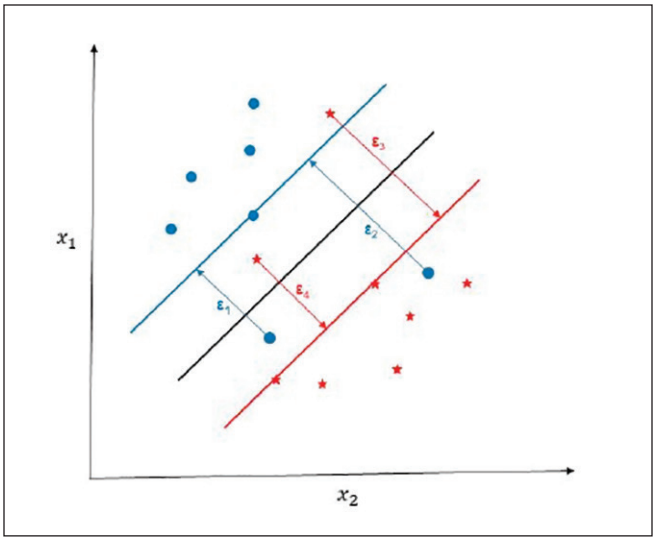

Figure 3–Optimal separating hyperplane defining decision boundary between linearly inseparable blue circles and red stars.

SVM Implementation

For this study, we used the SVM implementation that is available in OpenCV (OpenCV 2017), which used the LibSVM implementation which is very widely used by machine learning practitioners (Chang and Lin 2011). When users selected training pixels, it was common for there to be very small spatial spectral clusters of pixels in the training regions that did not match the class label assigned by the user. To compensate for this noise, the SVM used a soft margin when searching for the optimal separating hyperplane. Consideration was given to what value of C (defined in section 2.3) should be used while searching for the optimal separating hyperplane using a soft margin. Assessment of C allowed the determination of the most effective weight to be assessed for the error resulting from noisy training pixels. Additionally, we investigated the degree of linear separability of the training data in order to determine whether the original decision space was adequate for classification or if it was necessary to map the decision space to a higher dimensionality. In the situations where the original space was not adequately linearly separable, the RBF, Chi Squared and Histogram Intersection Kernels were evaluated.

k-Nearest Neighbor

The SVM is an eager learner, determining the decision boundary from the training data before considering any pixels with unknown class. Conversely, the k-Nearest Neighbor (kNN) is a lazy learner, just storing training pixels and waiting until it is given an unknown pixel before determining the decision criteria for the pixel.

*k*NN learns by analogy, comparing an unknown pixel to its most closely neighboring pixels in the decision space. Each pixel is described by n attributes, the proximity between pixels in the decision space being determined by the similarity between their attributes. Attribute similarity is defined in terms of a distance metric such as Euclidean distance. The Euclidean distance between points $X_{j1}, X_{j2},\ldots X_{jn}$) and $\boldsymbol{X}_j = (X_{j1}, X_{j2},\ldots X_{jn})$ is defined as

$$\text{Dist}\ (x_i x_j) = \sqrt{\Sigma_{k=1}^{n}\left(x_{ik} - x_{jk}{}^2\right)}. \qquad (9)$$

*k*NN works best if the attributes are numeric values that have been normalized (Han et al. 2012). When working with pixels from a color image, each pixel has attributes for the red, green and blue bands. Each value represents the spectral reflectance recorded in the associated spectra for that pixel. Color images acquired with the digital cameras on board sUAS have radiometric resolution of 8 bits, resulting in integer values between 0 and 255. Normalized values are very important for not allowing one attribute to dominate the distance metric (Han et al. 2012).

Classifying an unknown pixel is conceptually simple for a *k*NN classifier. Once the unknown pixel is located in the decision space, the k nearest pixels in the decision space are located. The unknown pixel is assigned the most common class occurring between the nearest neighbors. If the *k*NN is performing a binary classification, the pixel is assigned based on a simple majority between the two classes of training pixels (Russell and Norviq 2010).

Tuning *k*

The selection of a value for k is very important when classifying with a *k*NN. As k is increased, the decision boundary is smoothed, resulting in increased generalization (Russell and Norviq 2010). Typically, k is selected through experimentation, iteratively

running the classifier while increasing *k*, assessing classification accuracy at each iteration. The *k* value from the iteration resulting in the highest classification accuracy can be selected for use in that decision space (Han et al. 2012).

A related approach uses an iterative *n*-fold cross-validation using a single set of training pixels for both training and assessing accuracy for each value of *k* evaluated.

Implementation of the *k*NN Algorithm

An investigation of the OpenCV *k*NN implementation (OpenCV 2017) found that it was not capable of achieving the runtime efficiency required for this project. The classification of a 12 MP sUAS image was found to take about 10 minutes with the *k*NN. Image acquisition over a burned area that is hundreds of hectares in size was found to require multiple flights by a sUAS, with hundreds of photos taken. The largest orthomosaic created as part of this study exceeded two gigapixels (2,000 MP) in size. Classification of that orthomosaic with the OpenCV implementation of the *k*NN would have required ten hours. Consequently, a *k*NN was built around a customized n-D tree. Run time optimizations on the *k*NN were achieved by:

1. Pruning the tree so that training pixels with identical attributes were stored on a single node, which tracked how many training pixels existed with those attributes.
2. Balancing the tree once all the training pixels were loaded.
3. Implementing a recursion-less search of the tree for classifying unknown pixels.
4. Parallelizing classification of pixels with unknown class.

These customizations were successful in reducing classification runtime, resulting in a reduction of the time required to classify a 12MP image from 10 minutes with the OpenCV *k*NN implementation to 14 seconds with the custom *k*NN implementation computer with a dual-core CPU, a 42-fold decrease in runtime. Running the *k*NN on a server with even more physical cores will result in an even larger decrease in time required for classification due to the speedup resulting from dividing the classification of unknown pixels between even more processors (Cormen 2009).

Algorithm Comparison Hypothesis

In testing whether an SVM or *k*NN classifier can map wildland post-fire effects more accurately, a null hypothesis (H_0) was specified along with an associated alternate hypothesis (H_1). If H_0 was rejected, then H_1 would be accepted in its place. For this experiment, the independent variable was the algorithm selection between SVM and *k*NN. The dependent variable was burn severity mapping accuracy. The null and alternate hypotheses were:

H_0: Support Vector Machines and *k*NN have equal accuracy when mapping burn severity classes using hyperspatial color imagery.

H_1: Support Vector Machines have different accuracy than *k*NN when mapping burn severity classes using hyperspatial color imagery.

H_0 was tested by comparing algorithm accuracy, which was calculated from confusion matrices generated by validating an image classification by each of the algorithms against user labeled pixels. A two-tailed Student's t-test established the statistical significance of the accuracy results on the difference in accuracy between the SVM and *k*NN classifications. A p-value below a significance level of 0.05 rejected H0 in favor of H_1 which established that the mean accuracy of the SVM is different from the *k*NN. Once H_0 was rejected in favor of H_1, the more accurate algorithm was identified by selecting the algorithm that was found to have the highest accuracy results. The statistical analysis was again conducted with a single-tailed Student's t-test establishing the statistical significance of the accuracy results from the classifier with the higher accuracy against the classifier with the lower accuracy as shown in Algorithm 1.

```
For each burn image
        classify with SVM & kNN
        calculate accuracy (SVM & kNN)
                AE = (TBu +TU) / px
                AT = (TBl + TW) / Bu

For {AE, AT}    //Extent and AshType Accuracy
        t-test ← SVM vs kNN accuracies
```

Algorithm 1–SVM vs *k*NN experiment workflow. A = accuracy, E = extent, T = true, Bu = burned, U = unburned, px = pixels, Bl = black ash and W = white ash, A_E = extent accuracy and A_T = ash type (black vs. white) accuracy.

Algorithm Comparison Experiment Methodology

In order to assess the accuracy of both the SVM and *k*NN for mapping burn extent and biomass consumption, a set of orthomosaics acquired with sUAS over fires across the Boise National Forest (BNF) and Bureau of Land Management—Boise District (BLM-B) were hierarchically classified using both SVM and *k*NN classifiers. The first stage classified on burn extent, segmenting the image into burned and unburned regions. The second stage classified the burned regions of the image by black ash (indicative of partial combustion) versus white ash (indicative of more complete combustion).

Creation of Training Data

Training pixels were selected from post-fire imagery acquired with a sUAS over burned areas across the BNF and BLM-B. Training pixels were selected by the user identifying regions in the image which consisted of a set of homogeneous pixels, for which the user provided a class label. This process was aided by the Training Data Selector (TDS), a graphic tool developed through this research effort which assisted the user in identifying, selecting and labeling homogeneous regions of the image. Figure 4 shows an example of user labeled regions from the TDS denoting burn extent, where pixels in the green polygons are unburned pixels and red polygons denote black ash pixels. Figure 5 shows user labeled regions denoting burn severity, where pixels in the

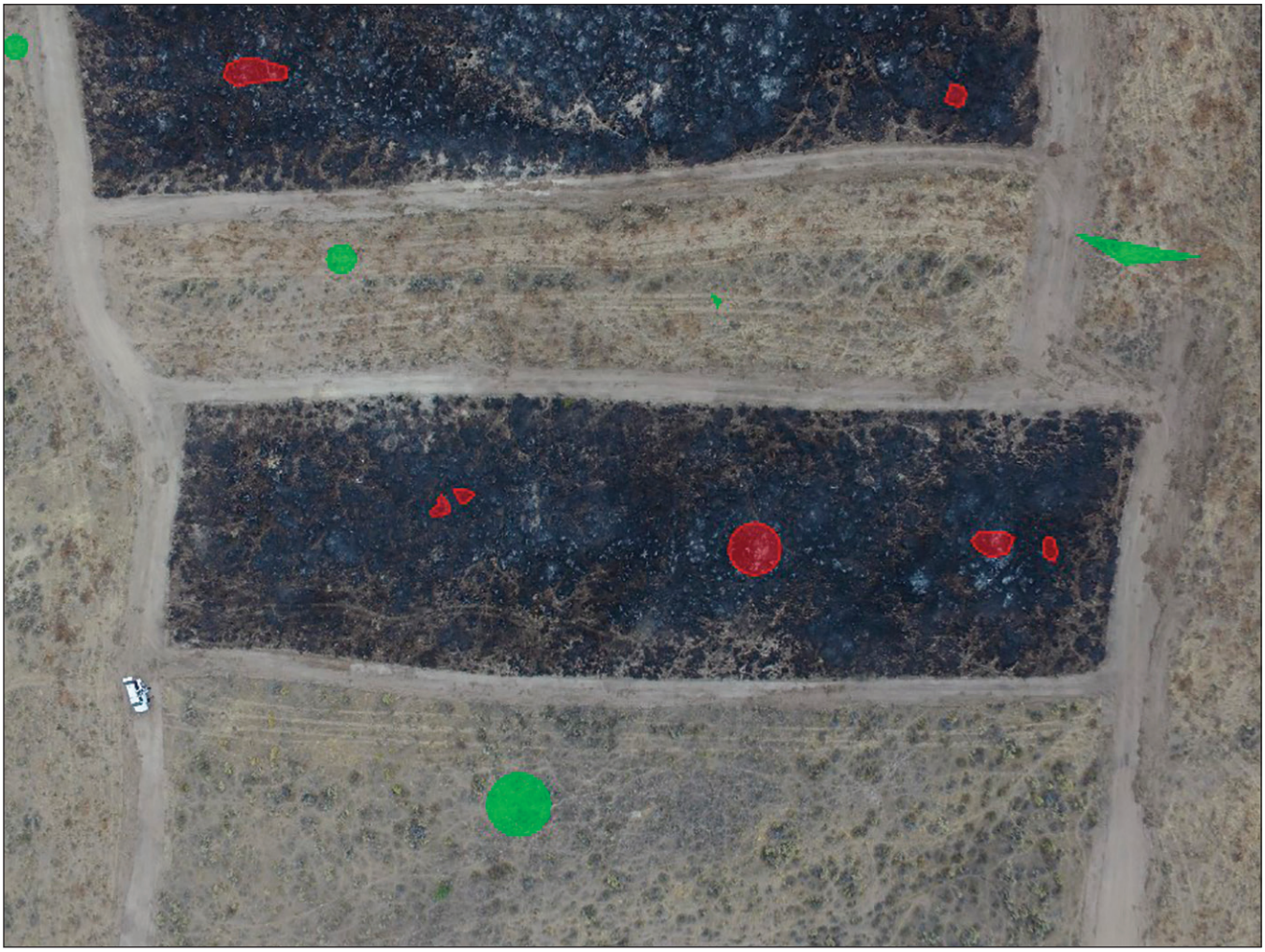

Figure 4–Training pixel regions for training burn extent as denoted by the user using the Training Data Selector. Green polygons are unburned pixels, red polygons denote black ash pixels.

blue polygons are labeled as white ash while the red polygons denote black ash.

When designating training regions in the image, it was advantageous to have approximately the same number of training pixels in each of the classes in order to ensure that the training data contained balanced amounts of both classes, which resulted in more accurate classifications. This criterion applied more to the *k*NN than the SVM, but since we were using the same training regions for both classifiers, the training data had to accommodate the implementations of both algorithms. It was also discovered that restricting the size of the training sets to under 1,000 pixels assisted the implementations of both algorithms in not over fitting the decision boundary to the data, providing for greater generalization. In order to allow the training regions to cover an adequate range of variability within a class, the TDS provided the user with the ability to select a sampling of pixels within a region. When the training pixels were extracted from the regions, only a user specified percentage of the pixels in the regions were placed in the training pixel set and labeled with the user specified class for the associated training region. Typically, when extracting pixels from a training region, only one to two percent of the pixels within the training regions were retained in the training sets by the TDS. This subsampling of the training regions provided for adequate representation of the variation within the training region while reducing the size of the training data.

Image Classification With SVM and *k*NN

The objective of classifying a post-fire image is to determine the extent of the burn as well as the level of biomass consumption within the burned area. To achieve that end, the image was first classified into burned and unburned regions. The classification of burn extent was facilitated by the spectral separability between burned and unburned vegetation as demonstrated previously in this research effort (Hamilton et al. 2017). Toward that end, the user selected burned and unburned regions within the training image using the TDS (fig. 5). Pixels within these regions were used for training the classifiers. The image was then classified into burned and unburned training pixels by either SVM or the *k*NN. With both classifiers, iterative 5-fold cross-validation (Han et al. 2012) was used to determine the optimal classifier parameter values as shown in table 1. Once the image was classified into burned and unburned pixels, a size filter was applied to the classified output using image processing morphological functions including dilation,

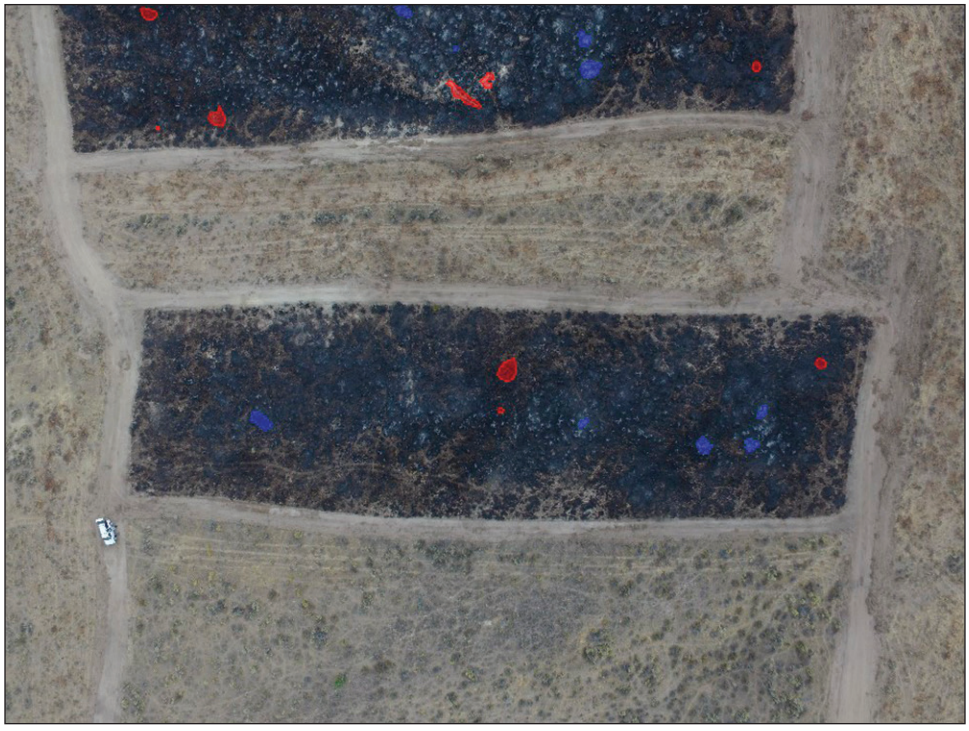

Figure 5–Training pixel regions classifying biomass consumption as denoted by the user using the Training Data Selector. Blue polygons are white ash, red polygons denote black ash.

Table 1—Optimal classifier parameter values.

Algorithm	Parameter	Classification type	Optimal value
SVM	Kernel	Burn extent	None (linear)
		Ash type	None (linear)
	C	Burn extent	0.1
		Ash type	0.1
*k*NN	*k*	Burn extent	3
		Ash type	3

erosion and the detection of very small connected components (Gonzalez and Woods 2008). Dilation is the process of gradually enlarging the boundaries of foreground pixels, while erosion has the opposite effect of eroding the boundaries of foreground pixels. These functions were used to remove spatial clusters of homogeneous pixels that were sub-object in size, filtering out clusters of pixels that are smaller than the objects being classified. For example, clusters of unburned vegetation smaller than a sagebrush that were surrounded by burned vegetation were merged into the surrounding black ash class. These misclassified clusters were commonly caused by small patches of herbaceous vegetation that were too sparse to carry adequate fire, or to adequately produce enough charred vegetation to be detected. In unburned regions of the image, misclassifications were commonly caused by shadows. (Misclassifications due to shadows were found to be reduced by conducting image acquisition flights as close to solar noon as possible, or by flying on cloudy days and slowing the shutter speed.) The size filter was also found to smooth the boundary between the burned and unburned classes in the classified image.

After cleaning the burn extent classification, the image was hierarchically classified with the burned region classified by biomass consumption using a binary classification between white ash (high consumption) and black ash (low consumption) using training pixels created as described above. An iterative five-fold cross validation was used to determine the optimal classifier parameter values as shown in table 1.

Validation of Classified Output

Accuracy was assessed on each of the classification types (burn extent, ash type and vegetation type) for both the SVM and *k*NN. Validation regions were created for each classification type in the same way as the training data was selected. The resulting validation pixel sets were loaded into a tool that compared the user specified class against the class predicted by the classifier, creating a confusion matrix (Han et al. 2012). Accuracy was calculated from the confusion matrix as discussed in Section 2.8.

Establishment of Statistical Significance

Statistical significance was assessed between the SVM and *k*NN classification accuracy results using the Student t-test as described in Section 2.8. H_0 states that there is not a difference in the accuracy with which the fire effects land cover classes can be classified using either SVM or *k*NN. Since we are using a hierarchical classification, we assessed the difference between accuracy results for the two algorithms with each of the classification types. This allowed us to have a finer scale assessment, allowing us to evaluate the difference in accuracy for both algorithms with each classification type. The two-tailed t-test p-value for a classification type fell below the previously stated significance level of 0.05, so we rejected H_0 (no difference in classification accuracy) in favor of H_1 (there is a difference in classification accuracy). We accepted the alternate hypothesis that one algorithm results in a more accurate classification, so a one-tailed t-test with a significance level below 0.05 was used to determine which algorithm was more accurate. Results of the algorithm comparison experiment are shown in the next section.

ALGORITHM COMPARISON

Assessment of the benefits of mapping wildland fire post-fire effects using Support Vector Machine (SVM) versus *k*-Nearest Neighbor (*k*NN) classifiers was accomplished by training an SVM and *k*NN with the same training regions, then classifying on the same post-burn orthomosaic, allowing for the comparison between classification outputs from both algorithms from a common training set. The assessment of both algorithms was explored over a set of burn orthomosaics from 16 fires, the results of which were

tested with a two-tailed t-test. The resulting p-values fell below the significance level, thus rejecting the null hypothesis that mean accuracy was the same for both algorithms in favor of the alternate hypothesis that the algorithms classify burn extent and biomass consumption with different accuracies. An observation of the SVM and *k*NN results showed that the SVM had a higher mean accuracy for both burn extent and biomass consumption. A one-tailed, paired t-test was then conducted to establish the statistical significance of the accuracy results to show that burn extent and biomass consumption can be classified with higher accuracy using the SVM algorithm.

Algorithm Comparison Accuracy Results

Evaluation of the capability with which the SVM and *k*NN were able to classify burn extent and biomass consumption was accomplished by assessing the accuracy of the output classifications of both classifiers. Accuracy is calculated as the number of samples correctly predicted by a classifier divided by the total number of samples (Han et al. 2012). To assess accuracy, both the SVM and the *k*NN were trained on a set of burn extent (burned and unburned) and biomass consumption (black ash and white ash) training regions within an image (fig. 6). The classifiers first performed a burn extent classification, labeling the pixels in the image as either burned or unburned. The image regions classified as burned were then classified by biomass consumption, labeling burned pixels as either white ash or black ash as shown in figure 7 (a-c).

Validation data sets for each of the images were selected as user labeled regions of pixels within the image, then the pixels from each validation dataset

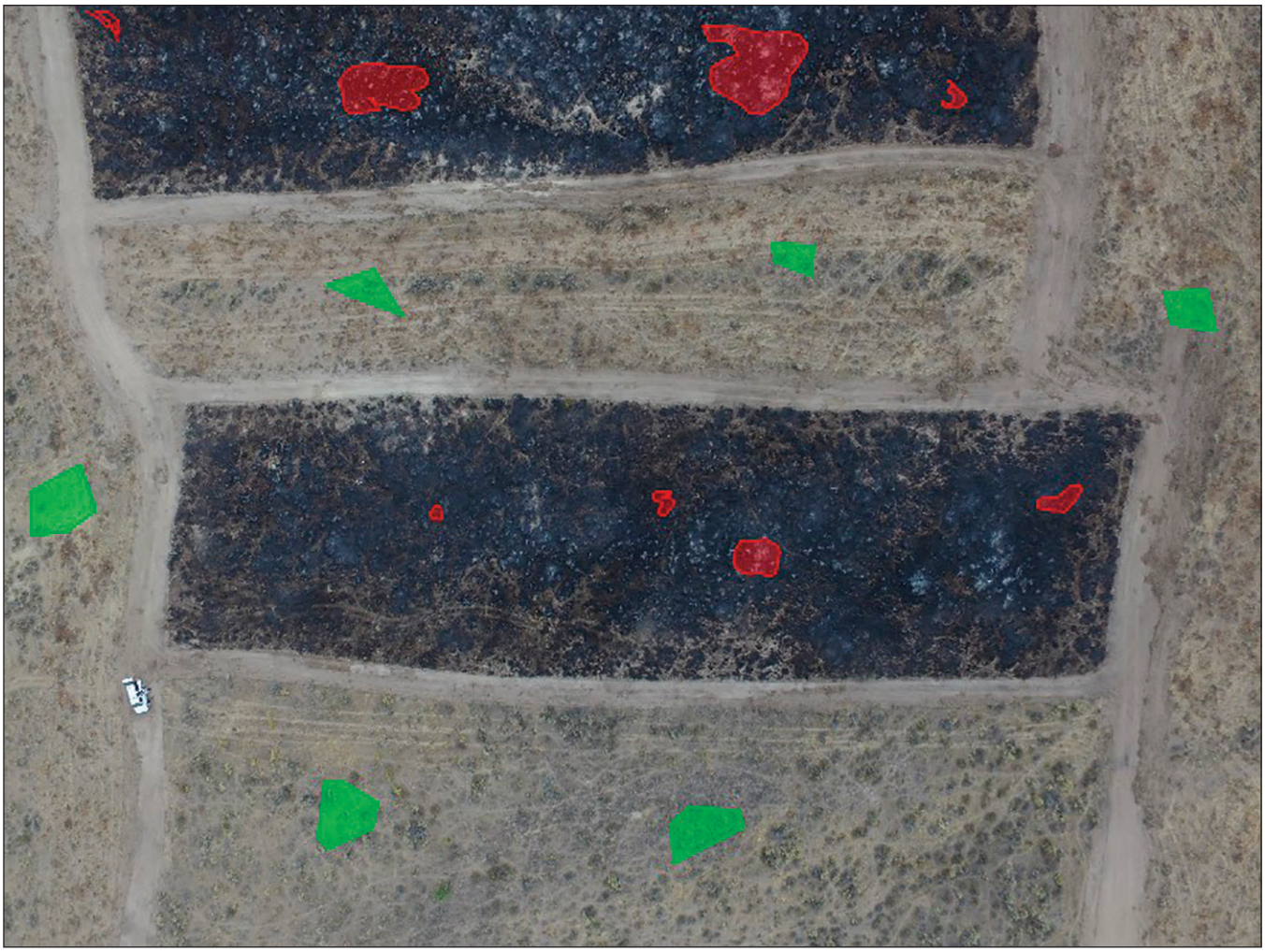

Figure 6–Example of validation data used for measuring classifier accuracy of figures 7(a) and 7(b).

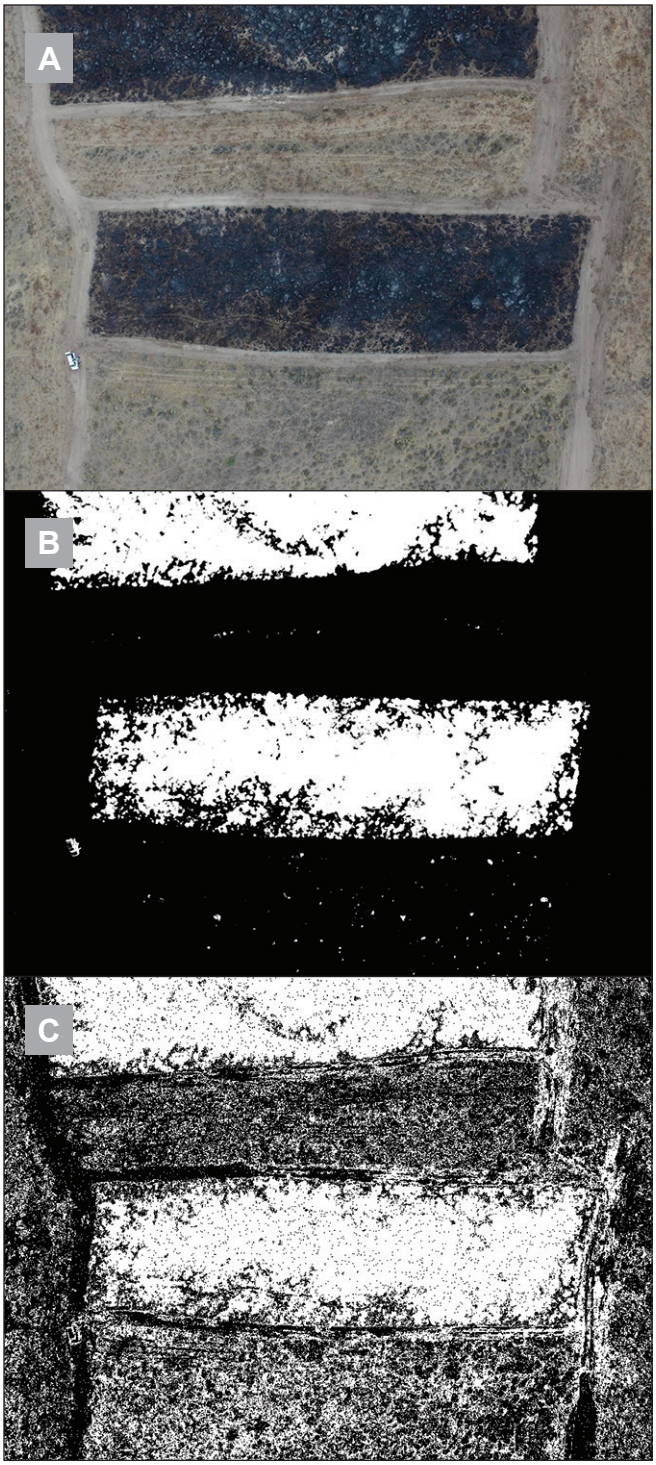

Figure 7–(a) Unclassified image of Reynolds Creek Prescribed burn; (b)–SVM classification result; (c)–NN classification result.

were run through the SVM and then again through the *k*NN. Accuracy for each validation data set was calculated for each algorithm, determining the percentage of validation pixels from a classifier which were classified the same as were labeled by the user. User labeling of validation data was based on visual observation of the image by the user, supplemented with ground observations recorded during image acquisition flights with the sUAS. Accuracy was calculated as the number of pixels correctly predicted by the SVM divided by the total number of pixels in the validation data set multiplied by 100. In order to obtain a more complete assessment of the accuracy of a set of classifier inputs, accuracy was evaluated based first on burn extent (ash vs unburned pixels) followed by assessment of biomass consumption (black ash versus white ash) accuracy. Accuracy results for each of the validation sets for burn extent and biomass consumption classifications from both classifiers on each fire are shown in table 2.

Classification accuracy for both burn extent and biomass consumption was averaged for both the SVM and *k*NN, then multiplied by 100. The resulting Mean Classification Accuracy are listed in table 3 for both SVM and *k*NN.

Table 2—SVM vs. *k*NN burn extent and biomass consumption classification accuracy.

	Burn extent		Biomass Consumption	
Fire	**SVM**	***k*NN**	**SVM**	***k*NN**
Jack	99.8	96.7	99.1	96.9
Northside	95.7	86.1	98	92.5
Reynolds Creek	99.59	79.51	98.7	96.85
Kane Fire	98.96	91.52	96.66	48.83
Deer Flat	99.99	72.62	99.86	78.06
Hoodoo 1	99.29	71.68	95.22	94.07
Hoodoo 2	89.23	50.36	99.25	92.88
Lucky Peak	97.61	74.62	99.85	96.17
mm107 (clip)	97.72	88.76	99.03	89.8
Camp	99.48	94.98	99.74	97.53
Oyhee plot	88.75	79.9	90.57	89.43
Immigrant (clip)	99.48	90.19	95.66	83.96
Elephant	95.08	91.44	99.15	96.68

Table 3—SVM vs. *k*NN mean classification accuracy.

Algorithm	Burn extent	Biomass consumption
SVM	96.97	97.75
*k*NN	82.18	88.74

A comparison of the Mean Classification Accuracy between the SVM and *k*NN showed that SVM had higher accuracy for both the burn extent and biomass consumption classifications.

Algorithm Comparison Accuracy Statistical Significance

The statistical significance of increased accuracy between the SVM and *k*NN across the validation sets for the burn images was established by using two-tailed paired t-tests. The null hypothesis is that the SVM and *k*NN classify burn extent and biomass consumption with equal accuracy. By contrast, the alternate hypothesis is that the SVM and *k*NN do not classify with equal accuracy. The t-test was run on the accuracy results for both the SVM and *k*NN, first testing burn extent, then biomass consumption. The significance level that the t-test passed is 0.05, which gives it 95 percent certainty to reject the null hypothesis in favor of the alternate hypothesis.

The burn extent accuracy tests rejected the null hypothesis with a p-value of 0.0005.

Likewise, the biomass consumption accuracy tests rejected the null hypothesis with a p-value of 0.031. In both cases, the null hypothesis was rejected, supporting the alternate hypothesis which shows that the SVM and *k*NN do not classify either burn extent or biomass consumption with equivalent accuracy.

The SVM was shown in table 3 to have a higher Mean Classification Accuracy for both burn extent and biomass consumption. Additionally, table 2 shows that the SVM had higher accuracy for each of the fires. To establish the statistical significance of the increase in accuracy over the *k*NN for both burn extent and biomass consumption, the accuracy values of both the SVM and *k*NN were run through a one-tailed t-test to establish that the SVM has a measurable increase in accuracy over the *k*NN.

The burn extent accuracy one-tailed t-test had a p-value of 0.0003, showing that the SVM has a measurable increase in mean accuracy over the *k*NN when classifying burn extent. Likewise, the biomass consumption one-tailed t-test had a p-value of 0.014, showing that the SVM had a measurable increase in accuracy over the *k*NN when classifying biomass consumption. This analysis shows that SVM classifies burn extent and biomass consumption with greater mean accuracy than *k*NN.

In addition to mapping post-fire effects with greater accuracy, the SVM implementation in this study also had a shorter execution time when classifying images, running in 20 to 25 percent of the time required by the *k*NN implementation to classify the same image. The comparison of the Support Vector Machine (SVM) and *k*-Nearest Neighbor (*k*NN) algorithms showed that both classifiers mapped burn extent and biomass consumption from hyperspatial sUAS imagery with high accuracy. The SVM outperformed the *k*NN in classifying burn extent as well as biomass consumption with the SVM mapping burn extent with average accuracy of 96.81 percent and biomass consumption with average accuracy of 97.74 percent on the fires studied. While the *k*NN did not map fire effects as accurately as the SVM, it still averaged 81.37 percent accuracy for burn extent and 88.74 percent for biomass consumption. The SVM outperformed the *k*NN, classifying burn extent and biomass consumption with higher accuracy when mapping post-fire effects from hyperspatial imagery acquired with sUAS.

CONCLUSION

The comparison of the Support Vector Machine (SVM) and *k*-Nearest Neighbor (*k*NN) algorithms showed that both classifiers mapped burn extent and biomass consumption from hyperspatial sUAS imagery with high accuracy. The SVM outperformed the *k*NN in classifying burn extent as well as biomass consumption with the SVM mapping burn extent with average accuracy of 96.81 percent and biomass consumption with average accuracy of 97.74 percent on the fires studied. While the *k*NN did not map fire effects as accurately as the SVM, it still averaged 81.37 percent accuracy for burn extent and 88.74

percent for biomass consumption. The inclusion of Second Order Entropy as a classifier input along with the three color bands was found to increase burn extent mapping accuracy with an SVM to 94.4 percent from 91.71 percent accuracy achieved using only color imagery. Likewise, the inclusion of Second Order Entropy increased the biomass consumption mapping accuracy to 83.8 percent from the 77.3 percent accuracy achieved using only the color bands as input to the SVM. While the inclusion of Second Order Entropy was able to increase fire effects mapping accuracy, that increase in accuracy must be weighed against the run-time computational complexity of calculating Second Order Entropy.

FUTURE WORK

Improve the SVM algorithm in both efficiency and accuracy. Apply research method to identifying prostate cancer. Apply research method to identifying and mapping archeological items of interest. Gather more data for testing and validating results. More data is required for determining statistical significance for white ash.

ACKNOWLEDGMENTS

This publication was made possible by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under Grant #P20GM103408 and the National Aeronautic and Space Administration through the Idaho Space Grant Consortium. We acknowledge the guidance and mentorship of Dr. Gregory Donohoe, Professor Emeritus of Computer Science at The University of Idaho as well as Dr. Eva Strand, Assistant Professor of Landscape Fire Ecology at The University of Idaho. We also acknowledge the assistance of undergraduate research assistants in the Computer Science program at Northwest Nazarene University including Zach Garner, Nicholas Hamilton, Llewellyn Johnston and Greg Smith.

REFERENCES

Aplet, G.H.; B. Wilmer, B. 2010. Potential for restoring fire-adapted ecosystems: Exploring opportunities to expand the use of wildfire as a natural change agent. Fire Management Today. 70(1): 35–39.

Brewer, C.K.; Winne, J.C.; Redmond, R.L.; Opitz, D.W.; Mangrich, M.V. 2005. Classifying and mapping wildfire severity. Photogrammetric Engineering and Remote Sensing. 71(11): 1311–1320.

Calkins, A.T. 2017. Unmanned aircraft systems (UAS) and photogrammetrics as a tool for archaeological investigation in 19th Century historic archaeology. Thesis. Reno, NV: University of Nevada. 122 p.

Chang, C.-C.; Lin, C.-J. 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2(3): 27.

Cormen, T.H. 2009. Introduction to algorithms. Cambridge, MA: MIT Press. 1292 p.

Cortes; C.; Vapnik, V. 1995. Support-vector networks. Machine Learning. 20(3): 273–297.

Eidenshink, J.C.; Schwind, B.; Brewer, K.; Zhu, Z.-L.; Quayle, B.; Howard, S.M. 2007. A project for monitoring trends in burn severity. Fire Ecology. 3(1): 3–21.

Gonzalez, R.; Woods, R. 2008. Digital image processing. Upper Saddle River, NJ: Pearson Prentice Hall. 945 p.

Hamilton, D.; Bowerman, M.; Collwel, J.; Donahoe, G.; Myers, B. 2017. A spectroscopic analysis for mapping wildland fire effects from remotely sensed imagery. Journal of Unmanned Vehicle Systems. DOI:10.1139/juvs-2016-0019

Han, J.; Kamber, M.; Pei, J. 2012. Data mining: Concepts and techniques, third edition. Waltham, MA: Morgan Kaufmann Publishers. eBook.

Hudak, A.T.; Ottmar, R.D.; Vihnanek, R.E.; Brewer, N.W.; Smith, A.M.; Morgan, P. 2013. The relationship of post-fire white ash cover to surface fuel consumption. International Journal of Wildland Fire. 22(6): 780–785.

Keeley, J.E. 2009. Fire intensity, fire severity and burn severity: A brief review and suggested usage. International Journal of Wildland Fire. 8(1): 116–126.

Key, C.H.; Benson, N.C. 2006. In: Lutes, D.C.; Keane, R.E.; Caratti, J.F.; Key, C.H.; Benson, N.C.; Sutherland, S.; Gangi. L.J., eds. FIREMON: Fire effects monitoring and inventory system. General Technical Report RMRS-GTR-164-CD. Fort Collins, CO: USDA Forest Service, Rocky Mountain Research Station: LA-1-LA-51.

Küng, O.; Strecha, C.; Beyeler, A. [et al.]. 2011. The accuracy of automatic photogrammetric techniques on ultra-light UAV imagery. Presented at: UAV-g 2011: Unmanned Aerial Vehicle in Geomatics, Zürich, CH, September 14-16, 2011.

Laliberte, A.S.; Herrick, J.E.; Rango, A.; Winters, C. 2010. Acquisition, orthorectification, and object-based classification of unmanned aerial vehicle (UAV) imagery for rangeland monitoring. Photogrammetric Engineering and Remote Sensing. 76(6): 661–672.

Lentile L.B.; Holden, Z.A.; Smith, A.M.S.; Falkowski, M.J.; [et al.]. 2006. Remote sensing techniques to assess active fire characteristics and post-fire effects. International Journal of Wildland Fire. 15(3): 319–345.

Liu, D.; Kelly, M.; Gong, P. 2006. A spatial–temporal approach to monitoring forest disease spread using multi-temporal high spatial resolution imagery. Remote Sensing of Environment. 101(2): 167–180.

Mora, C.; Tittensor, D.P.; Adl, S.; Simpson, A.G.; Worm, B. 2011. How many species are there on earth and in the ocean? PLoS Biology. 9(8): p. e1001127, 2011.

OpenCV, vol. 3.2, no. Computer Program, 2017. Available online at: www.opencv.org.

Rosnell, T.; Honkavaara, E. 2012. Point cloud generation from aerial image data acquired by a quadrocopter type micro unmanned aerial vehicle and a digital still camera. Sensors. 12(1): 453–480.

Russell, S. Norvig, P. 2010. Artificial intelligence: A modern approach. Third edition. Upper Saddle River, NJ: Prentice Hall. eBook.

Salimi, A.; Ziaii, M.; Amiri, A.; Zadeh, M.H.; Karimpouli, S.; Moradkhani, M. 2017. Using a Feature Subset Selection method and Support Vector Machine to address curse of dimensionality and redundancy in Hyperion hyperspectral data classification. Egyptian Journal of Remote Sensing and Space Sciences. 21(1): 27-36.

Sridharan, H.; Qiu, F. 2013. Developing an object-based hyperspatial image classifier with a case study using WorldView-2 data. Photogrammetric Engineering and Remote Sensing. 79(11): 1027–1036.

Yang, Y.; Liu, X. 1999. A re-examination of text categorization methods. SIGIR ’99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. New York and Berkeley, CA: Association for Computing Machinery: 42–49.

Zammit, O.; Descombes, S.; Zerubia, J. 2006. Burnt area mapping using support vector machines. Forest Ecology and Management. 234(1): S240.

Zhang, R.; Ma, J. 2008. An improved SVM method P-SVM for classification of remotely sensed data. International Journal of Remote Sensing. 29(20): 6029–6036.

Zhou, G.; Li, C.; Cheng, P. 2005. Unmanned aerial vehicle (UAV) real-time video registration for forest fire monitoring. Geosci. Remote Sens. Symp. 2005 IGARSS05 Proc. 2005 IEEE Int., vol. 3, 2005.