



**US Army Corps
of Engineers®**
Engineer Research and
Development Center

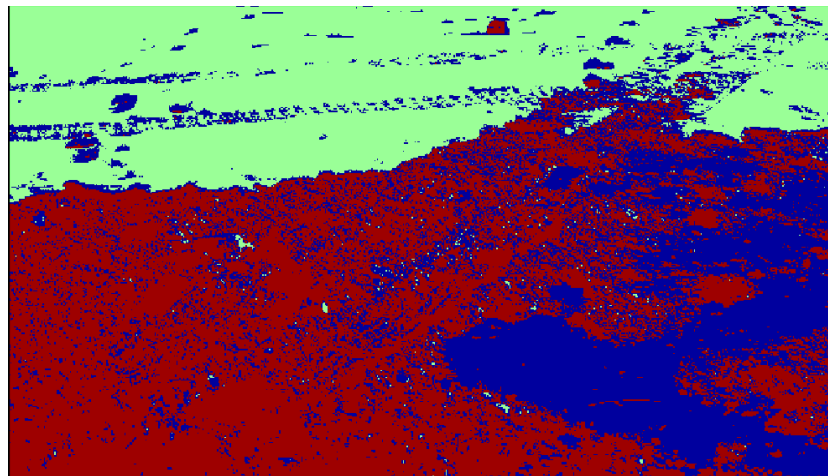


Cold Mobility Program

Automated Terrain Classification for Vehicle Mobility in Off-Road Conditions

Taylor S. Hodgdon, Anthony J. Fuentes, Jason L. Olivier,
Brian G. Quinn, and Sally A. Shoop

April 2021



The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdclibrary.on.worldcat.org/discovery.

To search for other technical reports published by ERDC, visit the ERDC online library at www.erdclibrary.on.worldcat.org/discovery.

Automated Terrain Classification for Vehicle Mobility in Off-Road Conditions

Taylor S. Hodgdon, Anthony J. Fuentes, Jason L. Olivier, Brian G. Quinn, and Sally A. Shoop

*U.S. Army Engineer Research and Development Center (ERDC)
Cold Regions Research and Engineering Laboratory (CRREL)
72 Lyme Road
Hanover, NH 03755-1290*

Final Report

Approved for public release; distribution is unlimited.

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under PE 0603463A, Project SBP402, Task A1070, "Machine Learning Analytics for
Cold Mobility," and PE 0602144A, Project BL7, Task SBL701

Abstract

The U.S. Army is increasingly interested in autonomous vehicle operations, including off-road autonomous ground maneuver. Unlike on-road, off-road terrain can vary drastically, especially with the effects of seasonality. As such, vehicles operating in off-road environments need to be informed about the changing terrain prior to departure or en route for successful maneuver to the mission end point. The purpose of this report is to assess machine learning algorithms used on various remotely sensed datasets to see which combinations are useful for identifying different terrain. The study collected data from several types of winter conditions by using both active and passive, satellite and vehicle-based sensor platforms and both supervised and unsupervised machine learning algorithms. To classify specific terrain types, supervised algorithms must be used in tandem with large training datasets, which are time consuming to create. However, unsupervised segmentation algorithms can be used to help label the training data. More work is required gathering training data to include a wider variety of terrain types. While classification is a good first step, more detailed information about the terrain properties will be needed for off-road autonomy.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Abstract	ii
Figures and Tables	iv
Preface	vi
Acronyms and Abbreviations	vii
1 Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Approach	2
2 Sensors and Field Sites	3
2.1 FLIR Duo Pro R	3
2.2 TerraSAR-X satellite	3
2.3 Velodyne lidar	3
2.4 Field sites	4
3 Machine Learning Results	5
3.1 Unsupervised algorithms	5
3.1.1 <i>K-means</i>	5
3.1.2 <i>Iterative Self-Organizing Data Analysis Technique (ISODATA)</i>	8
3.1.3 <i>Mean shift</i>	10
3.1.4 <i>Density Based Spatial Clustering of Applications with Noise (DBSCAN)</i>	11
3.2 Supervised algorithms	12
3.2.1 <i>QT Modeler and Metashape for elevation point-cloud data</i>	12
3.2.2 <i>ENVI supervised algorithms</i>	14
3.2.3 <i>Support vector machines</i>	16
4 Summary	19
5 Conclusions	20
6 Recommendations	22
References	23
Report Documentation Page	25

Figures and Tables

Figures

1	(A) The RGB image used in tandem with (B) the associated thermal image to perform the image segmentation, (C) the clustering results if two classes are specified (red: soil; blue: snow), and (D) the clustering results if three classes are specified (red: soil; blue: water, green: snow).....	6
2	(A) The high-resolution RGB image used to perform the segmentation, (B) the clustering results if two classes are specified with no smoothing filter used (red: wet tilled soil; blue: dry untilled soil), and (C) the clustering results with a 7×7 smoothing kernel applied prior to running the algorithm	7
3	(A) Multiband raster created using the high-frequency returns, low frequency returns, and HH datasets; and (B) the clustering results using the <i>K</i> -means algorithm in ENVI.....	8
4	(A) Multiband raster created using the VV, VH, and VH/VV datasets; (B) the clustering results using the <i>K</i> -means algorithm in ENVI; and (C) the clustering results using the ENVI ISODATA clustering algorithm.....	9
5	(A) Ideal segmentation with soil (<i>red</i>) and snow (<i>blue</i>) separated into two different clusters; (B) the clustering results using the same parameters as image A and a similar dataset (there is significant oversegmentation—four clusters); and (C) the clustering using the same parameters (there is under-segmentation—one cluster). Classes are shown as partially transparent over the RGB image	11
6	(A) Poor segmentation results using the DBSCAN algorithm with large epsilon and a large number of minimum samples (two colors represent segmented classes overlaid on the camera image). Note the segmented regions do not correspond to the actual terrain types we expected in this case. (B) Poor segmentation results with smaller epsilon and a smaller number of minimum samples resulting in four classes unrelated to terrain type but which may be related to rutting and terrain roughness or resulting shadowing of the image.....	12
7	(A) The unclassified point cloud from Range 4-3 at CEATS, collected on 13 March 2019, and (B) the same point cloud that has been classified using the classification algorithm in QT Modeler	14
8	(A) A classification attempt where four classes were specified and road points were selected from groomed snow trails and (B) the classification attempt just using snow and vegetation as class options. The <i>red</i> boxes indicate the extents of Range 6-6 (<i>top</i>) and Range 4-3 (<i>bottom</i>)	16
9	(A) A visual image collected using the FLIR Duo camera and (B) the corresponding thermal image.....	17
10	Scatter plot of training datasets using temperature and pixel intensity as training parameters. <i>Red points</i> show data points that are known to be ice, and <i>blue points</i> show data points that are known to be pavement. The black line separating the two classes is the hyperplane derived from the SVM algorithm. When using a test image, any pixel whose parameters fall on the right side of the hyperplane will be classified as pavement, and those that fall on the left side will be classified as ice	18

11 Proper classification of ice and pavement using the training datasets and hyperplane generated by the support vector machine. *The blue* region is pixels classified as ice, and the *red* region is pixels classified as pavement. Results are partially transparent superimposed over the original RGB test image.....18

Tables

1 Summary table of all examples outlined in this study. Classifications of “good,” “moderate,” and “poor” in the overall results column were generated by comparing the classified regions in the resultant image to visually distinguishable classes in the test images.....19

Preface

This study was conducted for the U.S. Army Corps of Engineers under PE 0603463A, Project SBP402, Task A1070, “Machine Learning Analytics for Cold Mobility,” and PE 0602144A, Project BL7, Task SBL701

The work was performed by the Terrestrial and Cryospheric Sciences Branch (Dr. John Weatherly, Chief), the Force Projection and Sustainment Branch (Dr. Dorothea Lundberg, Chief), and the Engineering Resources Branch (Dr. Caitlin Callaghan, Chief) of the Research and Engineering Division, U.S. Army Engineer Research and Development Center, Cold Regions Research and Engineering Laboratory (ERDC-CRREL). At the time of publication, Dr. David B. Ringelberg was Acting Division Chief. The Deputy Director of ERDC-CRREL was Mr. David B. Ringelberg, and the Director was Dr. Joseph L. Corriveau.

The authors acknowledge Ms. Michelle Michaels at ERDC-CRREL for technical review of this report and Dr. Sergey Vecherin for guidance on algorithm selection.

COL Teresa A. Schlosser was Commander of ERDC, and Dr. David W. Pittman was the Director.

Acronyms and Abbreviations

API	Application Programming Interface
CEATS	Camp Ethan Allen Training Site
CRREL	Cold Regions Research and Engineering Laboratory
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ENVI	Environment for Visualizing Images
ERDC	U.S. Army Engineer Research and Development Center
FLIR	Forward Looking Infrared
GNSS	Global Navigation Satellite System
HH	Horizontal-Horizontal Polarization
ISODATA	Iterative Self-Organizing Data Analysis Technique
QT Modeler	Quick Terrain Modeler
RGB	Red, Green, Blue
SAR	Synthetic Aperture Radar
SVM	Support Vector Machine
TerraSAR-X	X-Band Terra Synthetic Aperture Radar
TSX	TerraSAR-X
VH	Vertical-Horizontal Polarization
VV	Vertical-Vertical Polarization

1 Introduction

1.1 Background

Autonomous ground vehicle mobility is of great importance for successful military maneuvers. As such, there is growing interest in the U.S. Army for off-road autonomous vehicle capabilities. Autonomous vehicles allow for unmanned scouting missions through potentially hostile territory, significantly decreasing the risk of harm to the soldier. While there has been interest in this area for on-road autonomy, significant knowledge gaps exist for high-latitude regions where there are substantial seasonal impacts on terrain. To have successful autonomous mobility in these high-latitude regions, there must be accurate characterization of the terrain to provide look-ahead information. One way to provide this look-ahead information is by using machine learning algorithms in tandem with various sensors to group areas of the terrain into different classes. In fact, one of the most common machine learning applications is that of statistical segmentation (Jain et al. 1999; Sutton and Barto 1997; Dutton and Conroy 1996). This technique is used for determining which class or cluster a target data point belongs to based on one or more independent variables. In this analysis, we attempt to segment different types of terrain to provide look-ahead information for autonomous ground vehicles. To accomplish this task, we used several common machine learning algorithms. These algorithms fall into two subcategories: supervised and unsupervised. Unsupervised learning does not require any prior training datasets but can only segment an image into various unlabeled regions (Gentleman and Carey 2008). While these algorithms are quick to use and easy to implement, they are not able to provide classification labels to the segmented regions (Kotsiantis et al. 2007). Supervised algorithms are able to segment images and provide class labels for the segmented regions (Bhavsar and Ganatra 2012). The main drawback to supervised learning is that it requires vast amounts of labeled training data to effectively capture each class. The training data typically require a significant amount of user-dependent classification labeling (Singh et al. 2016), which is extremely time consuming if data were not collected in a restricted manner. As part of this analysis, we aim to see if using unsupervised algorithms can segment images for training datasets to later be used with supervised learning algorithms.

1.2 Objectives

The primary objective of this analysis is to compare various sensor datasets and machine learning algorithms to see which combinations would be effective for capturing different types of terrain. Secondarily, we wanted to test several common software packages to assess their effectiveness for analyzing these datasets.

1.3 Approach

To test this approach, we used a variety of sensors to collect the training and test datasets. Section 2 documents the sensors in detail. The datasets encompassed a wide range of terrain types, including snow, wet and dry soil, pavement, ice, and vegetation. We collected them as part of several field campaigns throughout the 2018–2019 field season. Therefore, each dataset was sampled under vastly different environmental conditions (i.e., temperature, lighting conditions, etc.). In addition to varied sensors and environmental conditions, we used several common supervised and unsupervised machine learning algorithms from Metashape, Quick Terrain Modeler (QT Modeler), ENVI (Environment for Visualizing Images), and the Scikit-Learn Python library (<https://scikit-learn.org>). The “Machine Learning Results” section documents each data and algorithm combination, where each example highlights a different aspect of the overall objective. Each example includes a pros and cons list associated with the sensor, algorithm, or software package used to process the data. Finally, in the Conclusions section we discuss the results and indicate which sensor and algorithm combinations are most fruitful for future analyses.

2 Sensors and Field Sites

2.1 FLIR Duo Pro R

The FLIR (forward-looking infrared) Duo Pro R camera produces two separate images—one that contains RGB (Red, Green, Blue) bands and one that stores the thermal information on a pixel-wise basis. We selected this FLIR camera as it provides in addition to the RGB imagery thermal information about the surface it is sampling. However, because of the offset between the two cameras and differing view angles, overlapping a radiometric thermal image with an RGB “true-color” image requires significant manual manipulation using FLIR proprietary software. Due to this manual manipulation, the algorithms were primarily tested on either the RGB image or the thermal image alone. Two tests used the manually overlapped images to see if including the thermal information as an extra parameter aided in the classification process.

2.2 TerraSAR-X satellite

TerraSAR-X (X-band Terra Synthetic Aperture Radar, TSX) is a high-resolution commercial Earth observation imaging radar. The satellite collects in the X-band of the microwave region of the electromagnetic spectrum with a 31 mm wavelength (frequency of 9.6 GHz). The satellite has the ability to collect imagery in like-polarized (vertical-vertical, VV) and cross-polarized (vertical-horizontal, VH) combinations. Furthermore, the TSX data can be provided as either single polarization, where VH or VV is captured, or dual polarization where VH and VV are combined, for example. The data can also be analyzed as a ratio of VH and VV. Depending on the collection mode, TSX can produce images with a 30 cm resolution. The StripMap images used in this analysis have a resolution of 6m.

2.3 Velodyne lidar

The Velodyne HDL-32E is a 32-beam, three-dimensional, high-definition lidar sensor that has a range of up to 100 m with an accuracy of approximately 2 cm. The lidar sensor has $+10^\circ$ and -30° vertical field of view relative to the horizontal plane of the sensor. The sensor is also coupled with a 360° horizontal field of view resulting from the rotation of the scanner. Lidar sensors are often paired with integrated inertial measurement units and Global Navigation Satellite System devices to convert the collected returns to the sensor’s local coordinate system, followed by a conversion of

the returns to a geospatial coordinate system. This configuration is less feasible under GPS-denied conditions. Because of this shortcoming, the HDL-32E employs a Kaarta Stencil 2-32 mobile mapping computer in place of an inertial measurement unit and Global Navigation Satellite System. The Kaarta Stencil 2-32 is a mobile mapping system that uses an RGB camera with a simultaneous localization and mapping framework to map the sensor position based on the identification of common landmark features across the scanned region. The resulting point clouds are georeferenced using ground control points with known geographic coordinates during postprocessing.

2.4 Field sites

For this analysis, the CRREL mobility team collected data from several field sites over the 2018–2019 season. These sites include the Chilean Andes near Lonquimay, Chile; the Camp Ethan Allen Training Site (CEATS) in Jericho, Vermont; Tullando Farm in Orford, New Hampshire; Keweenaw Research Training Center in Calumet, Michigan; and at the Cold Regions Research and Engineering Lab (CRREL) in Hanover, New Hampshire. The team selected each of these sites because they exhibited varied terrain types that could be leveraged to test the various sensors and machine learning algorithms.

3 Machine Learning Results

This section reviews the supervised and unsupervised algorithms used as part of this analysis. As mentioned in section 1.3, each example highlights a unique combination of a machine learning algorithm applied to a specific sensor dataset. In each example, we have provided the collection location and date, the machine learning algorithm being tested, and what sensors and parameters were used. For some of the examples, we utilized smoothing filters to remove noise in the datasets.

3.1 Unsupervised algorithms

3.1.1 *K*-means

K-means clustering is a type of unsupervised learning used when data is unlabeled (i.e., data without defined categories or groups) (Forgy 1965). It is also one of the most commonly used clustering algorithms as it is quick and generally provides good segmentation results. The goal of this algorithm is to find groups that cluster similar types of data values. The number of groups or clusters are represented by the variable *K*. The algorithm works iteratively to assign each data point to a *K* group based on the features that are provided (Hartigan and Wong 1979). Data points are clustered based on feature similarity (i.e., which cluster centroid they lie closer to). One drawback to this algorithm is that the number of clusters must be specified prior to running the algorithm; therefore, it requires the user to know a priori how many classes may be represented in an image.

Our analysis tested this algorithm on both the FLIR and TSX datasets. For the FLIR data, we used the *K*-means clustering algorithm from the Scikit-Learn Python library. We used ENVI software to perform the *K*-means clustering segmentation on the TSX datasets. Shown below are examples of the *K*-means algorithm results from a variety of test areas.

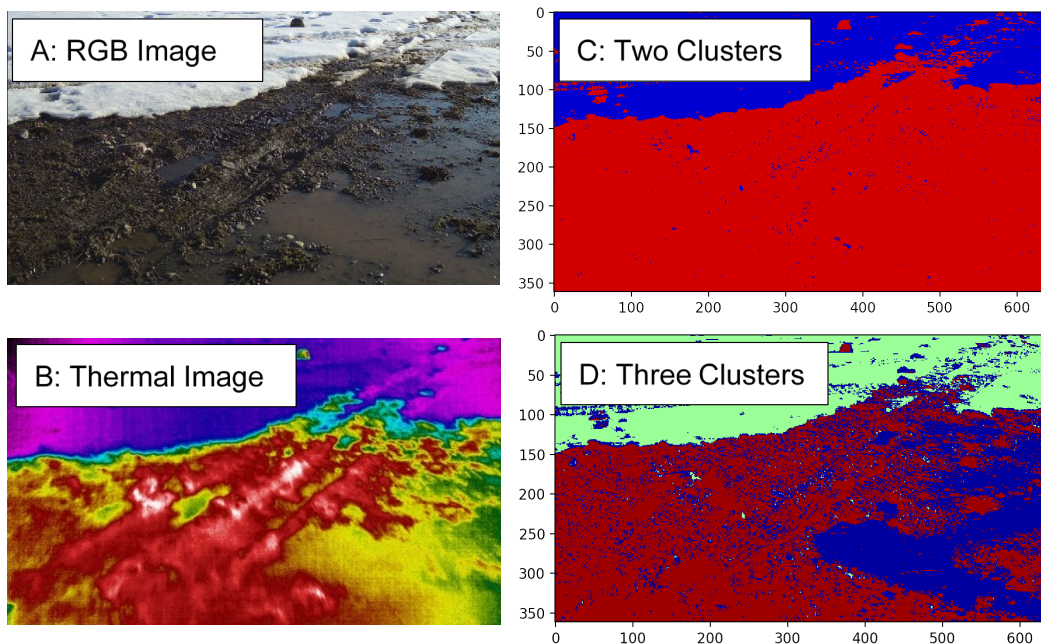
Example 1: Chile 2019 Collection—Snow vs. Soil vs. Water

- Sensor: Vehicle-mounted FLIR camera
- Parameters: Temperature and RGB intensity values
- Algorithm: *K*-means clustering (Scikit-Learn)
- Other comments: No smoothing or neighborhood information used

Pros: For this example, *K*-means performs a good segmentation of the image when using either two (Figure 1C) or three (Figure 1D) clusters. In this example, water is segmented from the surrounding soil even though they have similar RGB values (this is a scenario when using the thermal parameter is useful).

Cons: The number of clusters must be chosen prior to running the algorithm, indicating a need for a priori knowledge of the terrain. *K*-means does not provide class labels for the different segmented regions. The temperature parameter aids in distinguishing regions of similarity in an image; however, it requires manual manipulation beforehand to properly align the two images prior to segmentation.

Figure 1. (A) The RGB image used in tandem with (B) the associated thermal image to perform the image segmentation, (C) the clustering results if two classes are specified (red: soil; blue: snow), and (D) the clustering results if three classes are specified (red: soil; blue: water, green: snow).



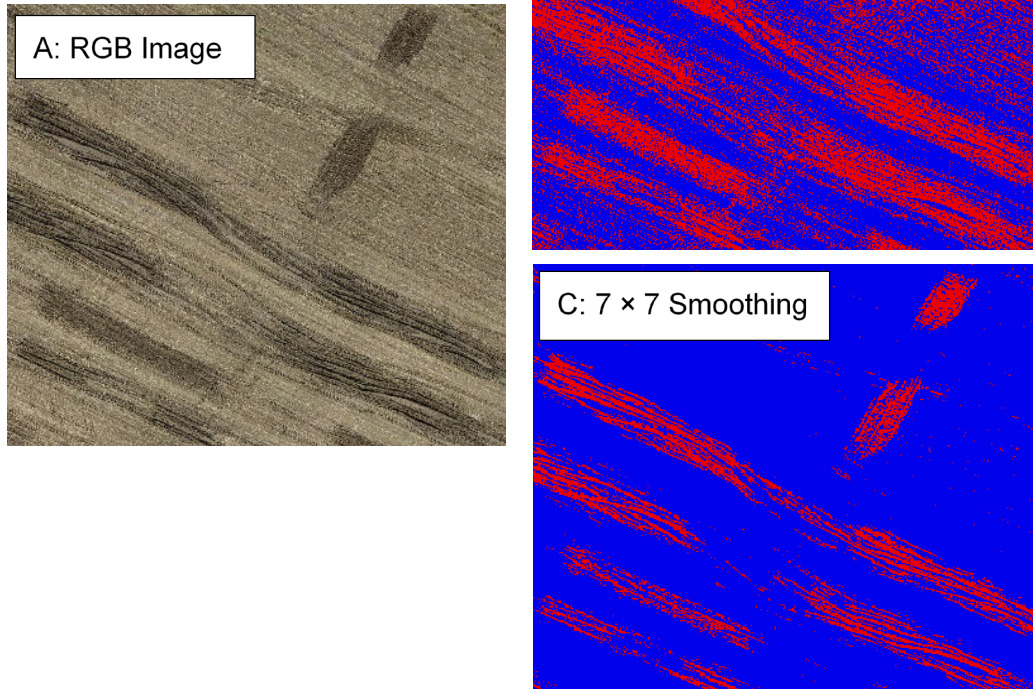
Example 2: Tullando Farm 2019—Wet vs. Dry Soil

- Sensor: RGB camera on an unmanned aerial drone
- Parameters: RGB intensity values
- Algorithm: *K*-means clustering (Scikit-Learn)
- Other comments: 7×7 Gaussian smoothing kernel used to homogenize noisy regions

Pros: The K -means algorithm performs a better segmentation when using kernel smoothing to homogenize noisy regions (Figure 2C). The segmentation results are robust even when using only RGB values (Figure 2B), and no preprocessing had to be performed on the images to align datasets since infrared was not used.

Cons: The number of clusters must be known prior to running the algorithm, once again requiring a priori knowledge of the terrain. This algorithm does not provide class labels for the different segmented regions. Determining a proper kernel size will vary from image to image depending on the sampled media and the sample resolution.

Figure 2. (A) The high-resolution RGB image used to perform the segmentation, (B) the clustering results if two classes are specified with no smoothing filter used (red: wet tilled soil; blue: dry untilled soil), and (C) the clustering results with a 7×7 smoothing kernel applied prior to running the algorithm.



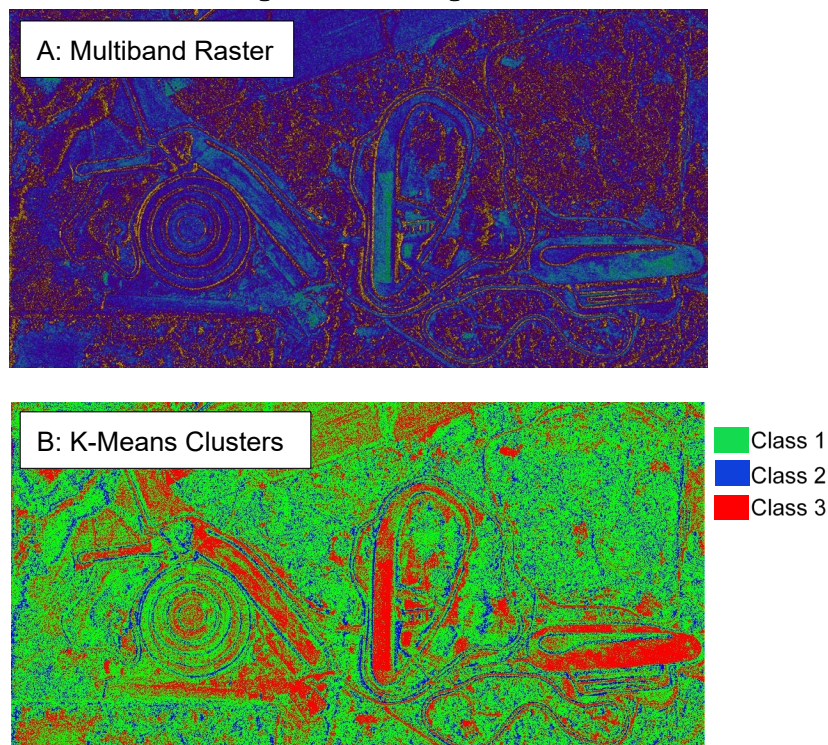
Example 3: Keweenaw Research Center 2020—Vegetation vs. Snow vs. Roads

- Sensor: TSX satellite
- Parameters: Multiband synthetic aperture radar (high frequency, low frequency, horizontal-horizontal polarization (HH))
- Algorithm: *K*-means clustering (ENVI)

Pros: The ENVI software is easy to use and has the ability to be automated with an application programming interface (API).

Cons: Similar to the two previous examples, the number of clusters must be chosen prior to running the algorithm, and it does not provide class labels for the different segmented regions (Figure 3). The ENVI software is proprietary and must be purchased to access the API.

Figure 3. (A) Multiband raster created using the high-frequency returns, low frequency returns, and HH datasets; and (B) the clustering results using the *K*-means algorithm in ENVI.



3.1.2 Iterative Self-Organizing Data Analysis Technique (ISODATA)

The ISODATA algorithm (Ball and Hall 1965) is one of the unsupervised classification algorithms in the ENVI framework. It calculates the means

of each class and then iteratively clusters the remaining pixels by using minimum distance techniques (i.e., which class mean the pixel falls closest to). Each iteration recalculates the means and reclassifies pixels with respect to the new mean.

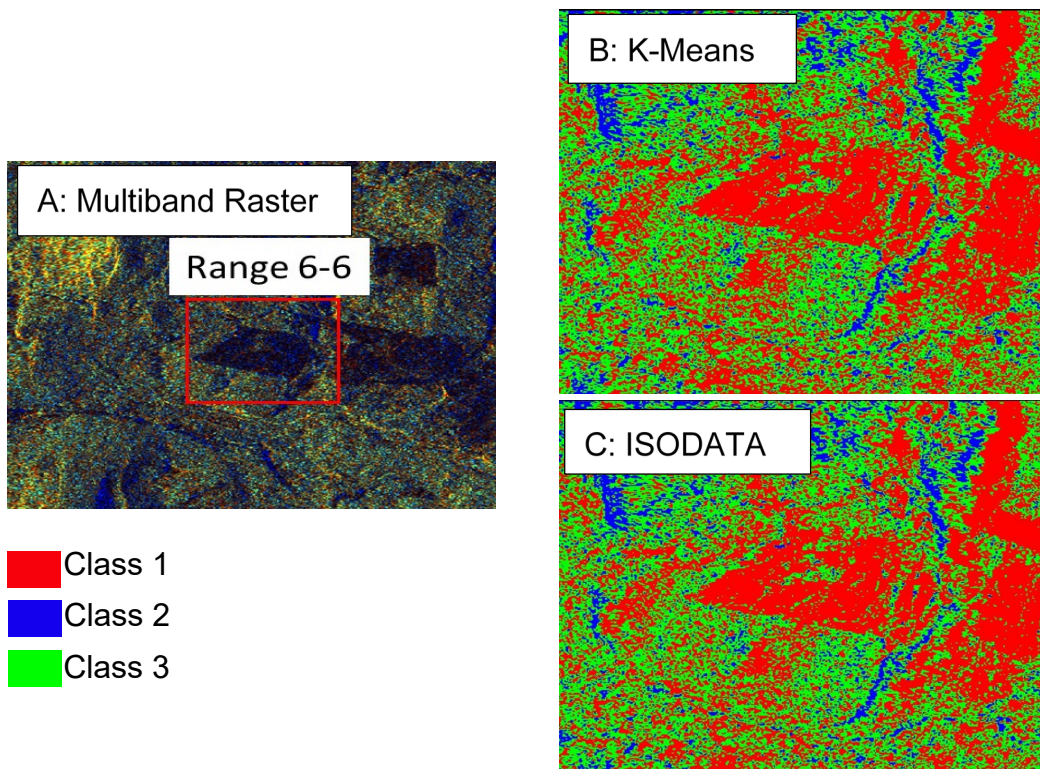
Example 4: CEATS 2019, Range 6-6

- Sensor: TSX satellite
- Parameters: Multiband SAR (VV, VH, VH/VV)
- Algorithm: ISODATA (ENVI) compared to K -means

Pros: Overall, ISODATA provided comparable results (Figure 4C) to the ENVI K -means segmentation (Figure 4B). ISODATA also had a quick segmentation time, which is crucial for real-time terrain characterization.

Cons: ISODATA had comparable segmentation results to K -means but was slower to execute. Also, prior to running, ISODATA requires a user-defined input of the number of expected classes.

Figure 4. (A) Multiband raster created using the VV, VH, and VH/VV datasets; (B) the clustering results using the K -means algorithm in ENVI; and (C) the clustering results using the ENVI ISODATA clustering algorithm.



3.1.3 Mean shift

Mean shift is a clustering algorithm that assigns data points to clusters iteratively by shifting points towards the mode (Cheng 1995). Given a set of data points, the algorithm assigns each data point towards the closest cluster centroid. The direction to the closest cluster centroid is determined by where most of the points nearby are located. During every iteration, each data point will move closer to the largest cluster of points. When the algorithm finishes, each point is assigned to a cluster. Unlike the *K*-means algorithm, mean shift does not require specifying the number of clusters in advance. However, the user needs to specify other parameters, such as the maximum distance apart points can be within a single cluster, which factors into the number of final clusters.

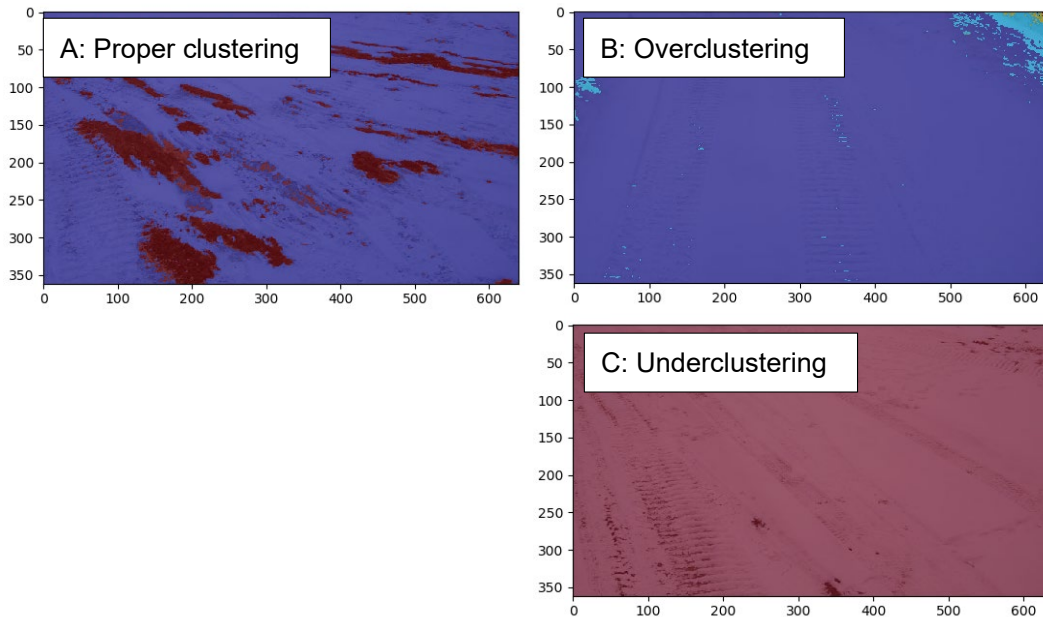
Example 5: CEATS 2019—Snow vs. Vegetation vs. Soil

- Sensor: Vehicle-mounted RGB camera
- Parameters: RGB intensity
- Algorithm: Mean shift (Scikit-Learn)

Pros: This algorithm does not require the user to specify the number of clusters prior to segmentation. The algorithm optimizes the number of clusters and works well on some images (Figure 5A).

Cons: User needs to specify other crucial parameters to calculate an appropriate probability kernel bandwidth, otherwise clustering will provide poor results (Figure 5B and C). The algorithm is not robust enough to properly segment images in a batch-processing manner. Therefore, processing needs to happen on an image-by-image basis, with the user adjusting the input values.

Figure 5. (A) Ideal segmentation with soil (*red*) and snow (*blue*) separated into two different clusters; (B) the clustering results using the same parameters as image A and a similar dataset (there is significant oversegmentation—four clusters); and (C) the clustering using the same parameters (there is under-segmentation—one cluster). Classes are shown as partially transparent over the RGB image.



3.1.4 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN clustering is another unsupervised clustering algorithm that divides the data points into a number of specific batches or groups such that the data points in the same groups have similar properties (Ester et al. 1996). The DBSCAN algorithm requires two parameters: epsilon and minimum points. Epsilon defines the radius of the neighborhood around a data point (i.e., if the distance between two points is lower or equal to epsilon, then they are considered neighbors). If the epsilon value is too small, then a large part of the data will not be grouped into clusters and will be considered outliers. If a large epsilon value is chosen, then the clusters will merge; and the majority of the data points will be in the same clusters. The minimum number of points defined specifies the number of data points that must fall within the epsilon radius to identify a new cluster.

Example 6: CEATS 2019—Snow vs. Soil

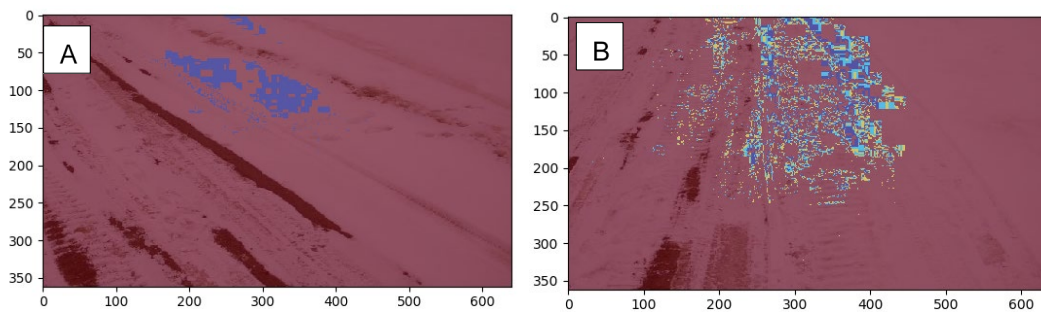
- Sensor: Vehicle-mounted RGB camera
- Parameters: RGB intensity

- Algorithm: DBSCAN (Scikit-Learn)
- Other comments: Varied epsilon and minimum samples for each image

Pros: This algorithm does not require the user to specify the number of clusters prior to segmentation.

Cons: The user needs to specify crucial parameters, otherwise clustering will provide poor results (Figure 6A and B). The algorithm is not robust enough to properly segment images in a batch-processing manner. Therefore, processing needs to happen on an image-by-image basis with the user adjusting the input values.

Figure 6. (A) Poor segmentation results using the DBSCAN algorithm with large epsilon and a large number of minimum samples (two colors represent segmented classes overlaid on the camera image). Note the segmented regions do not correspond to the actual terrain types we expected in this case. (B) Poor segmentation results with smaller epsilon and a smaller number of minimum samples resulting in four classes unrelated to terrain type but which may be related to rutting and terrain roughness or resulting shadowing of the image.



3.2 Supervised algorithms

This study used three supervised techniques to classify varying datasets. Two of these were pretrained on existing datasets from the ENVI and QT Modeler software, and the third used training data collected at CRREL.

3.2.1 QT Modeler and Metashape for elevation point-cloud data

The assignment of points within a lidar point cloud to either “Ground” or “Non-Ground” objects has been widely explored within the literature (Zhang et al. 2003, 2016; Pingel et al. 2013). These ground-point classification methods outlined in existing literature, while distinct in their methodology, all rely on determining an elevation threshold between a reference surface and a neighborhood of points that can account for the differences in spatial extent between different features (e.g., buildings, trees, and cars). In contrast to the well-described, open-source, ground-point

classification algorithms, the proprietary ground-point classification algorithms within QT Modeler and Metashape are black boxes with respect to their filtering methodologies.

The QT Modeler and Metashape software employ proprietary point-cloud classification algorithms that allow for the segmentation and labeling of different features (e.g., trees, ground, and buildings) recorded in lidar- or photogrammetry-derived point clouds. Unlike with the open-source Scikit-Learn Python library, these algorithms are not explicitly described or available within an API. The current lack of a coding interface in the ENVI software package reduces the applicability of these software-based algorithms for real-time classification tasks.

QT Modeler requires that all input data, including photogrammetry-derived point clouds, be in an LAS or LAZ (zipped LAS) file format. QT Modeler segments the image based on four predefined classes taken from the American Society of Photogrammetry and Remote Sensing classification: 0 = not classified, 2 = ground, 3 = low vegetation, and 6 = building. There is currently no functionality that allows for incorporating user-defined algorithms or further classification beyond these four classes. The strength of QT Modeler is that it is able to distinguish nonground objects (e.g., buildings and trees) from ground points within the point-cloud data. The point-cloud density must exceed one point per square meter for the algorithm to be effective. This can be challenging in environments with dense tree cover or other sheltering features as the number of return points will be much less dense than in open environments. Additionally, in the classifier prompt prior to running the algorithm, the user must select the specific classes that are known to be represented in the point cloud.

Metashape, like QT Modeler, uses a classification algorithm that segments a point cloud into predefined classes: ground, high vegetation, building, road, car, and man-made (e.g., infrastructure). The algorithm allows the user to define a confidence threshold between 0 and 1 that will assign points below the threshold as “unclassified.” The user may manually select objects or regions within the point cloud to be assigned to one of the predetermined classes. There is no functionality as of yet to create custom classes (i.e., wet ground, snow, etc.).

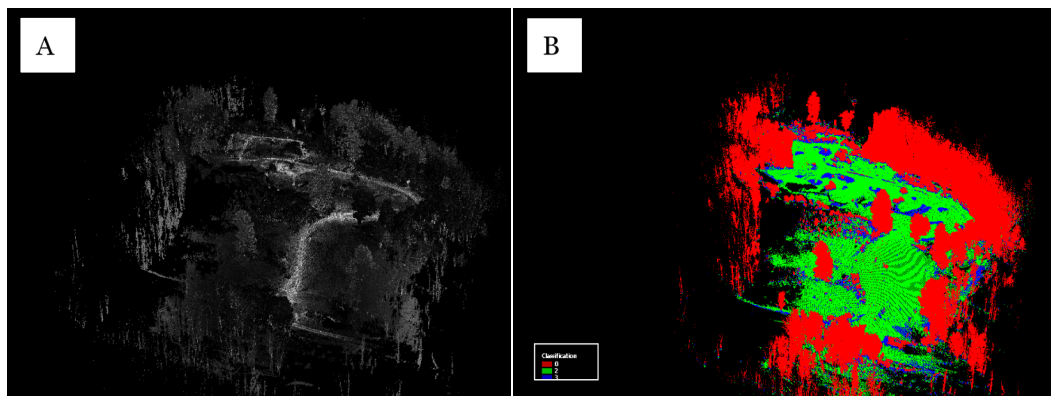
Example 7: CEATS 2019—Lidar Ground-Point Classification

- Sensor: Vehicle-mounted Velodyne HDL-32E lidar
- Parameters: Point-cloud elevation values
- Algorithm: QT Modeler point-cloud classification algorithm

Pros: This algorithm provides fast classification of ground, vegetation, and low vegetation classification (Figure 7). It is robust against topographic variation that often hinders other algorithms.

Cons: The proprietary algorithm is a black box and is not configurable with user-defined classes. It has no existing API that allows for scripting functionality to enhance classification performance.

Figure 7. (A) The unclassified point cloud from Range 4-3 at CEATS, collected on 13 March 2019, and (B) the same point cloud that has been classified using the classification algorithm in QT Modeler.



3.2.2 ENVI supervised algorithms

ENVI is a software application used to process and analyze geospatial imagery and bundles together a number of scientific algorithms for image processing. The software includes numerous algorithms to process imagery for terrain classification. This project used the generic ENVI image classification workflow for supervised classification. This analysis used imagery from 13 March 2019 for two areas at CEATS. We selected Range 4-3 and Range 6-6 as they were relatively open spaces where three classes were visible on panchromatic imagery: snow, vegetation, and trails.

This test used maximum likelihood classification in ENVI (Richards 2012). Maximum likelihood classification assumes that the statistics for each class in each dataset follow a gaussian distribution. With that assumption,

it calculates the probability that a given pixel belongs to a specific class. Each pixel is assigned to the class that has the highest probability. Supervised classification requires large amounts of labeled data to train the algorithm. It is advantageous to use larger areas to ensure the pixels used are fully representative of the terrain class, since pixels on the edge of two classes could confuse the algorithm. The training data used to complete the supervised classification included four classes: road, building, snow, and vegetation. Approximately 15,500 pixels were used for the training data (road: 252 points; building: 146; snow: 7956; vegetation: 7157) and were taken from the entire image. Because of the inability to distinguish roads and buildings on the images, the number of points to represent those classes was significantly lower than for snow and vegetation. The reduced number of road and building samples relative to the other classes likely reduced the ability of the algorithm to differentiate those two classes. Two different runs were conducted for supervised classification: the first run was conducted with the four classes where the road points were selected from groomed snow trails, and the second was a basic two-class snow versus nonsnow classification. The second run proved to be most useful because while large features could be identified in the scene, the image resolution did not allow for identification of smaller features or the ability to distinguish between snow and dirt or virgin and groomed snow.

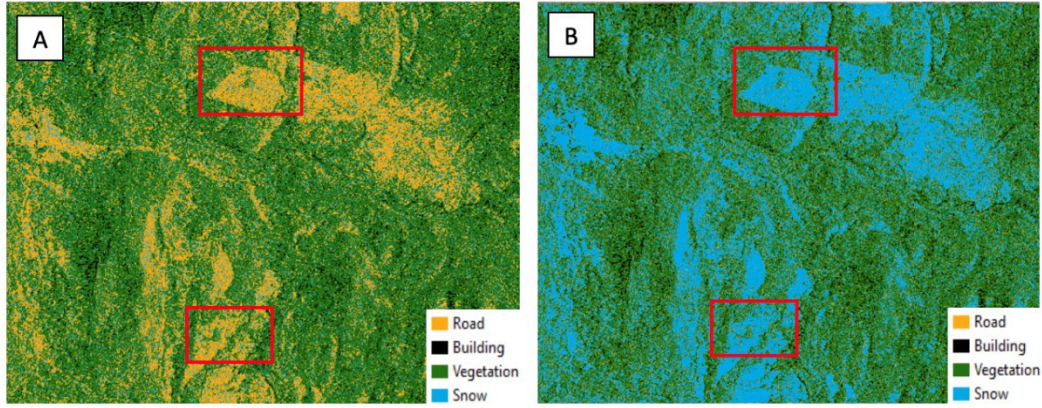
Example 8: CEATS 2019—TSX Imagery Classification

- Sensor: TSX satellite
- Parameters: Multiband SAR (VV, VH, VH/VV)
- Algorithm: ENVI maximum likelihood

Pros: Supervised classification allows the user to pick regions of their choice to represent a surface, allowing for the best possible classified image. This algorithm provides class labels for each pixel as opposed to just segmenting into various clusters. ENVI software is user friendly and provides an API for script processing.

Cons: Classes and regions need to be defined beforehand, making cognitive classification much more difficult. Manually labeling training data points is time consuming and labor intensive. Accurate training labels are essential for proper classification of images (Figure 8A and B).

Figure 8. (A) A classification attempt where four classes were specified and road points were selected from groomed snow trails and (B) the classification attempt just using snow and vegetation as class options. The *red boxes* indicate the extents of Range 6-6 (*top*) and Range 4-3 (*bottom*).



3.2.3 Support vector machines

The support vector machine (SVM) method is based on the concept of decision planes that separate data points belonging to different class members (Gunn 1998). These decision planes are constructed from the training datasets by determining the greatest distances between points of separate classes. In addition to the SVM method being suitable for a broad spectrum of machine learning classification problems, several other factors make it well suited for real-time classification problems. First, SVMs typically use simple decision boundaries, thus lessening the risk of overfitting (Wang 2005). Second, once the SVM has been trained and the decision plane constructed, this method is computationally inexpensive, which is necessary for real-time evaluation (Gunn 1998).

For simplicity, we elected to classify two different types of terrain: ice and pavement. However, this method could be adapted to incorporate several other classes. The input data used for the training datasets include infrared and visual imagery collected with a FLIR camera. Images were collected at different points throughout the winter 2019 season to account for effects of varied solar loading and ambient air temperature. Roughly 150 infrared images and 150 visual images were collected of each class through the sampling season. Images of pure ice and pure pavement were collected to make the classification of the training datasets simpler. Several images contacts between ice and pavement were sampled to be used as the test data for the analysis (Figure 9A and B).

This analysis used the Scikit-Learn Python library as it provides well-tested and readily available SVM functionality. A linear kernel was used along with a default penalty parameter of 1.0. All other optional parameters were set to their default values. Figure 10 shows the results of the test-image analysis (Figure 9A and B). Overall, the classification method worked well given the input data used for the analysis. The thermal signature of the different classes dominates the placement of the decision plane, indicating that the decision to classify a pixel as either ice or pavement depends heavily on this parameter (Figure 11). Ice is translucent and will often have the same RGB coloring as the material beneath it (Figure 9A). However, ice has a different thermal signature than pavement, so it is not surprising that the thermal parameter is crucial for distinguishing between the two classes (Figure 9B).

Example 9: Cold Regions Research and Engineering Laboratory—Ice vs. Pavement

- Sensor: Tripod-mounted FLIR camera
- Parameters: Temperature and RGB intensity values
- Algorithm: Support vector machine (Scikit-Learn)
- Other comments: No smoothing or neighborhood information used

Pros: Overall, this algorithm performs a good classification of images. Using the thermal information is key since RGB intensity values are indistinguishable between ice and pavement. The main benefit of SVM is the near-real-time classification after the algorithm has been thoroughly trained.

Cons: As with any supervised algorithm, SVM requires a lot of training data collected under a variety of environmental conditions. While the classification process itself is fast, it takes a long time to train the SVM algorithm.

Figure 9. (A) A visual image collected using the FLIR Duo camera and (B) the corresponding thermal image.

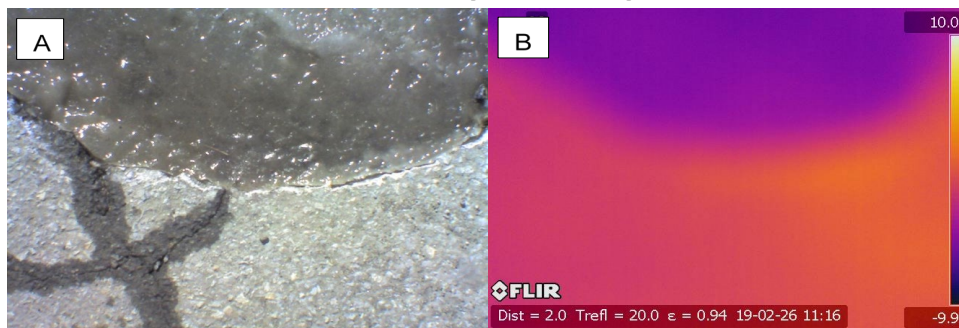


Figure 10. Scatter plot of training datasets using temperature and pixel intensity as training parameters. *Red points* show data points that are known to be ice, and *blue points* show data points that are known to be pavement. The black line separating the two classes is the hyperplane derived from the SVM algorithm. When using a test image, any pixel whose parameters fall on the right side of the hyperplane will be classified as pavement, and those that fall on the left side will be classified as ice.

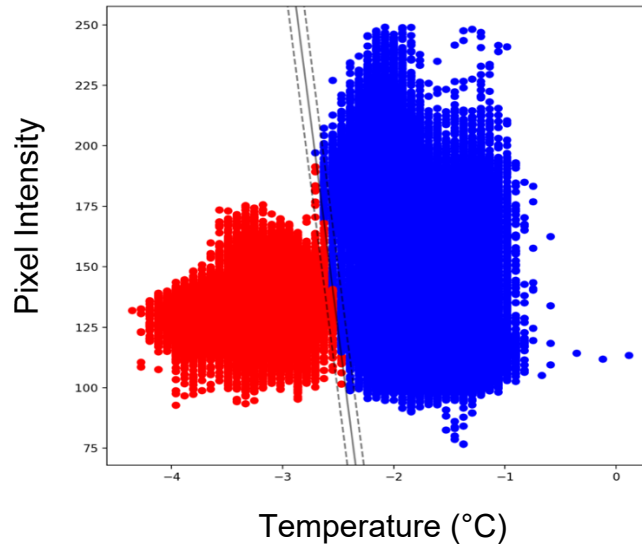
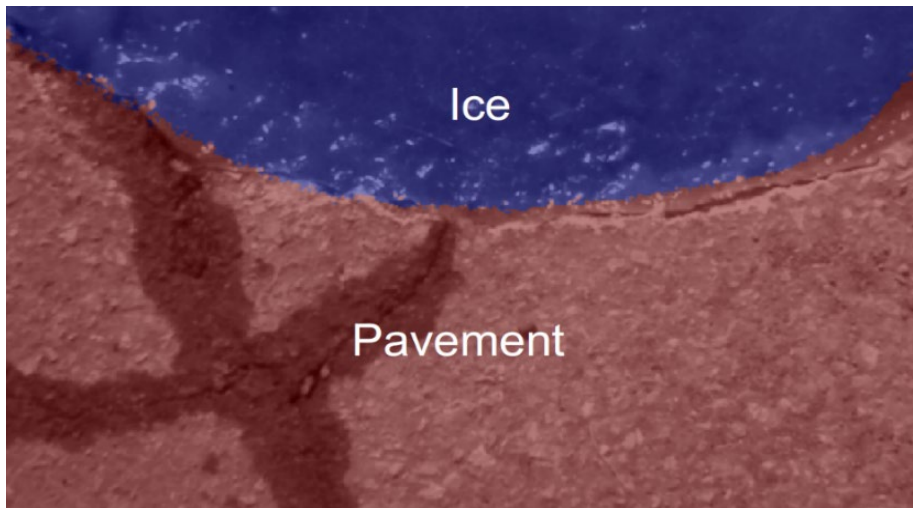


Figure 11. Proper classification of ice and pavement using the training datasets and hyperplane generated by the support vector machine. *The blue region* is pixels classified as ice, and the *red region* is pixels classified as pavement. Results are partially transparent superimposed over the original RGB test image.



4 Summary

The primary objective of this analysis was to see what combination of sensor and machine learning algorithms would be effective for capturing different types of terrain to inform vehicle operations. Three sensors were used for data collection (FLIR with visual camera, TSX, and lidar), and the analysis used a variety of supervised and unsupervised machine learning techniques. Each test example highlights the pros and cons of using that particular sensor and machine learning algorithm. Table 1 highlights all of the examples discussed in previous sections.

Table 1. Summary table of all examples outlined in this study. Classifications of “good,” “moderate,” and “poor” in the overall results column were generated by comparing the classified regions in the resultant image to visually distinguishable classes in the test images.

Example	Sensor	Algorithm	Software	Software Availability	Overall Results
1	FLIR	<i>K</i> -means	Scikit-Learn	Open source	Good
2	RGB	<i>K</i> -means	Scikit-Learn	Open source	Good
3	TSX	<i>K</i> -means	ENVI	Proprietary	Moderate
4	TSX	ISODATA	ENVI	Proprietary	Moderate
5	RGB	Mean shift	Scikit-Learn	Open source	Poor
6	RGB	DBSCAN	Scikit-Learn	Open source	Poor
7	Lidar	Proprietary	Metashape	Proprietary	Good
8	TSX	Max likelihood	ENVI	Proprietary	Moderate
9	FLIR	SVM	Scikit-Learn	Open source	Good

5 Conclusions

The aim of this study was to compare and contrast various sensors and machine learning algorithms to see which combinations were best suited for characterizing terrain. The following are the main takeaways from this analysis:

- Unsupervised algorithms can be used to segment the training data into different clusters but require a user to manually label the identified clusters.
- The most accurate unsupervised algorithm seems to be *K*-means, as it properly segmented the images. The mean shift and DBSCAN algorithms provided poor segmentation results.
- The main drawback with the *K*-means algorithm is that the number of clusters in each image must be known prior to running the algorithm. Thus, there will likely need to be close supervision of the results to ensure proper clustering.
- When using clustering algorithms with high-resolution data, it may be necessary to smooth the data prior to running the algorithm. This will help eliminate outliers from each cluster and eliminate highly speckled classification results
- One way to remove the need for segmenting the images is to collect training datasets of each desired class in a regimented way. For example, if using a camera to collect images of ice and snow, collect images of each class individually. That way all pixels in one image are known to belong to a single class, thus eliminating the need for segmenting and labeling.
- As demonstrated in Example 1 in section 3.1.1, the FLIR Duo camera proved to be a valuable sensor for identifying different types of terrain. However, to use both the thermal and visual information requires manually shifting the datasets so they overlap one another.
- Thermal information seems to be most useful when there are multiple classes with similar pixel intensities, such as water or ice, as they display the same color as the terrain type beneath them but have differing thermal signatures.
- QT Modeler and Metashape both provide decent unsupervised results; however, they allow for only the classes built into the software. So, incorporating classes such as snow, ice, and wet and dry soil is not possible.
- Similar to QT Modeler and Metashape, ENVI has predefined classes for its standard classification algorithm. However, it also allows for user-

defined classes and provides a simple framework for generating training datasets. The robustness of these algorithms strictly depends on how well the training data represent each assigned class.

- In general, incorporating more parameters aids in the classification process, as it adds more distinctive differences between each group.

The results of this analysis provide initial insight into which combinations of sensors and algorithms are useful for automated terrain classification. This work provides the initial step for ensuring successful off-road autonomous vehicle mobility in varied seasonal terrain.

6 Recommendations

While this work provides a preliminary assessment of the various sensors and algorithms that can inform autonomous vehicle platforms about terrain, there is still a significant amount of work to do. For classifying terrain, the unsupervised algorithms cannot be used as they only separate the data into clusters instead of labeled class regions. However, these unsupervised methods can segment data for training supervised machine learning algorithms, which would make the training step more automated and less time consuming. While classification of the terrain can provide some useful look-ahead information for autonomous vehicles, the vehicle must know the underlying mechanical properties of the terrain (i.e., soil strength, snow density, ice roughness, etc.). Thus, the next step is to identify which sensors and machine learning algorithms are able to address these properties. This area of research has large knowledge gaps and will require many follow-on studies to develop successful off-road autonomous vehicle capabilities.

References

- Ball, G. H., and D. J. Hall. 1965. *ISODATA, a Novel Method of Data Analysis and Pattern Classification*. Menlo Park, CA: Stanford Research Institute.
- Bhavsar, H., and A. Ganatra. 2012. "A Comparative Study of Training Algorithms for Supervised Machine Learning." *International Journal of Soft Computing and Engineering (IJSCE)* 2 (4): 2231–2307.
- Cheng, Y. 1995. "Mean Shift, Mode Seeking, and Clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (8): 790–799.
- Dutton, D., and G. Conroy. 1996. "A Review of Machine Learning." *Knowledge Engineering Review* 12:341–367.
- Ester, M., H. P. Kriegel, J. Sander, and X. Xu. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *KDD-96, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* 96 (34): 226–231.
- Forgy, E. W. 1965. "Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications." *Biometrics* 21 (3): 768–769.
- Gentleman, R., and V. J. Carey. 2008. "Unsupervised Machine Learning." In *Bioconductor Case Studies*, 137–157. New York: Springer.
- Gunn, S. R. 1998. *Support Vector Machines for Classification and Regression*. ISIS Technical Report. Southampton, UK: University of Southampton. <http://svms.org/tutorials/Gunn1998.pdf>.
- Hartigan, J. A., and M. A. Wong. 1979. "Algorithm AS 136: A K-Means Clustering Algorithm." *Journal of the Royal Statistical Society, Series C* 28 (1): 100–108.
- Jain, A. K., M. N. Murty, and P. Flynn. 1999. "Data Clustering: A Review." *ACM Computing Surveys* 31 (3): 264–323.
- Kotsiantis, S. B., I. Zaharakis, and P. Pintelas. 2007. "Supervised Machine Learning: A Review of Classification Techniques." *Emerging Artificial Intelligence Applications in Computer Engineering* 160:3–24.
- Pingel, T. J., K. C. Clarke, and W. A. McBride. 2013. "An Improved Simple Morphological Filter for the Terrain Classification of Airborne LIDAR Data." *ISPRS Journal of Photogrammetry and Remote Sensing* 77:21–30.
- Richards, J. 2012. *Remote Sensing Digital Image Analysis: An Introduction*. 5th ed. New York: Springer. <https://doi.org/10.1007/978-3-642-30062-2>.
- Singh, A., N. Thakur, and A. Sharma. 2016. "A Review of Supervised Machine Learning Algorithms." In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 16–18 March 2016, New Delhi, India, 1310–1315.

- Sutton, R. S., and A. G. Barto. 1997. *Introduction to Reinforcement Learning*. Cambridge, MA: MIT press.
- Wang, L. 2005. *Support Vector Machines: Theory and Applications*. New York: Springer. <https://doi.org/10.1007/b95439>.
- Zhang, K., S. C. Chen, D. Whitman, M. L. Shyu, J. Yan, and C. Zhang. 2003 “A Progressive Morphological Filter for Removing Nonground Measurements from Airborne LIDAR Data.” *IEEE Transactions on Geoscience and Remote Sensing* 41 (4): 872–882. <https://doi.org/10.1109/TGRS.2003.810682>.
- Zhang, W., J. Qi, P. Wan, H. Wang, D. Xie, X. Wang, and G. Yan. 2016. “An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation.” *Remote Sensing* 8 (6): 501.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) April 2021		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) FY20-FY21	
4. TITLE AND SUBTITLE Automated Terrain Classification for Vehicle Mobility in Off-Road Conditions				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT 0603463A and 0602144A	
6. AUTHOR(S) Taylor S. Hodgdon, Anthony J. Fuentes, Jason L. Olivier, Brian G. Quinn, and Sally A. Shoop				5d. PROJECT NUMBER SBP402 and BL7	
				5e. TASK NUMBER A1070 and SBL701	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center (ERDC) Cold Regions Research and Engineering Laboratory (CRREL) 72 Lyme Road Hanover, NH 03755-1290				8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/CRREL TR-21-4	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Headquarters, U.S. Army Corps of Engineers Washington, DC 20314-1000				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The U.S. Army is increasingly interested in autonomous vehicle operations, including off-road autonomous ground maneuver. Unlike on-road, off-road terrain can vary drastically, especially with the effects of seasonality. As such, vehicles operating in off-road environments need to be informed about the changing terrain prior to departure or en route for successful maneuver to the mission end point. The purpose of this report is to assess machine learning algorithms used on various remotely sensed datasets to see which combinations are useful for identifying different terrain. The study collected data from several types of winter conditions by using both active and passive, satellite and vehicle-based sensor platforms and both supervised and unsupervised machine learning algorithms. To classify specific terrain types, supervised algorithms must be used in tandem with large training datasets, which are time consuming to create. However, unsupervised segmentation algorithms can be used to help label the training data. More work is required gathering training data to include a wider variety of terrain types. While classification is a good first step, more detailed information about the terrain properties will be needed for off-road autonomy.					
15. SUBJECT TERMS Automated vehicles--Off-road operation, Computer algorithms, Machine learning, Military geography, Remote sensing, Trafficability					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)