Samuel Levin
7 May 2019
GISC6317
Dr. Chastain

Machine Learning Detection of Archaeological Features from Thermal Imagery

Introduction

This project uses Python based image processing principles and machine learning libraries to classify archaeological features in thermal imagery. Thermal infrared remote sensing has vast potential for the detection of subsurface archaeological features. Using a multitemporal thermal dataset collected over the summer of 2018, these methods are applied to classify possible archaeological features at Picuris Pueblo, New Mexico.

Intended Audience/User

This analysis methodology was developed primarily for use by archaeologists interested in exploring the subsurface potential of sites. However, these methods may be adapted to serve other industries where subsurface materials are of interest and the appropriate environmental conditions exist.

Expected Inputs

- Multi-temporal, radiometric thermal dataset
- Polygon Area-Of-Interest feature classes
- Labeled training data locations

Output

- A series of Jupyter notebooks summarizing the analysis
- Feature engineered raster bands enhancing thermal effects
- Classified rasters indicating areas with potential for subsurface features

Usage

The included Jupyter Notebook files (.ipynb) must be opened and run using a Jupyter environment. Jupyter notebooks provide a convenient medium for sharing the code, commentary, and many intermediate outputs. Jupyter is installed by default with ArcGIS Pro.

In addition, a YAML file is included, which can be used to clone the Conda environment used for this analysis, including all dependencies and packages. This can be accomplished using the following command in a Python command prompt interface:

```
conda env create -f environment.yml
```

The Jupyter notebooks can be run by pointing the `arcpy.env.workspace` variable at the start of each notebook to the path of the input GDB (without outputs). Each block of code in the notebooks can be run individually, or the entire notebook may be run at once. New data will be saved to the GDB as the analysis is run, including extracted areas of interest and final classified output rasters. Commentary is included throughout, describing the objectives of each processing step.

NOTE: Some notebooks may take a long time (> 1 hour) to run depending on the power of the computer. For this reason, the outputs are provided in the second GDB.

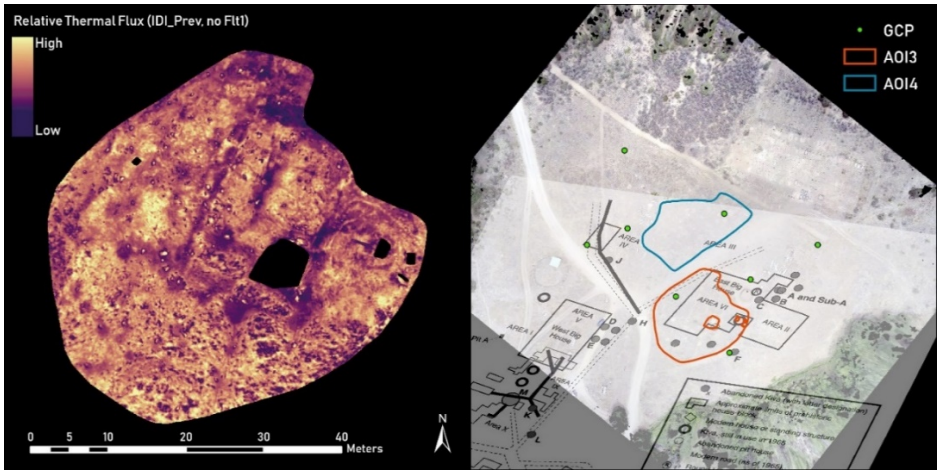
Analysis and Methodology

The Jupyter notebooks work through exploratory data analysis of the raw dataset, feature engineering of new features, extracting training data and areas of interests, through final processing using machine learning models.

The ESRI `arcpy` module is used for translating the georeferenced raster data into Python native data formats. All analysis including exploratory data analysis, feature engineering, and classification is conducted using open-source Python libraries. This analysis relies heavily upon the `NumPy` library, providing multi-dimensional array objects foundational to scientific processing. Extensive data manipulation of `NumPy` arrays is practiced throughout. All charts including histograms, raster visualizations, and scatterplots are achieved using the `Matplotlib` and `Seaborn` plotting libraries. Tables are managed using the `Pandas` library. Finally, `Scikit-learn` provides a framework for implementing the support-vector machine model.

The IDI feature engineering process works by enhancing the thermal trends of features in multitemporal imagery. Through image scaling and differencing, areas that are changing temperature more or less than the surrounding soil matrix are emphasized. These might indicate thermal effects caused by the varied thermal inertia of subsurface architectural features. Using the five min-max scaled observations and four varieties of IDI relative thermal flux, nine raster bands were used as input for supervised classification analysis.

Labeled feature training data are extracted from locations with a distinct IDI signature and from areas coincident with known features from 1960s excavations.

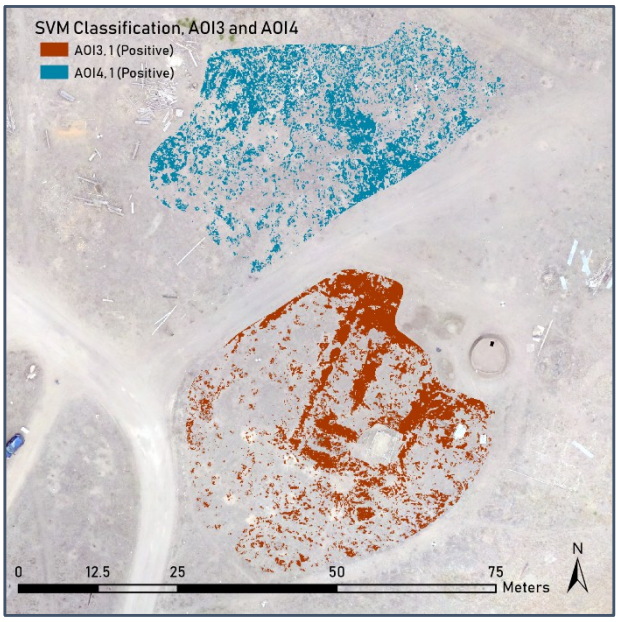


A support-vector machine classifier and decision tree classifier class are employed to identify the locations of architectural features. This classification problem has two goals. The first goal is to train the SVM classifier to identify known features with high precision and recall (>90% accuracy). This is conducted in AOI3, coincident with the 1960s excavations. The second goal is to use the models trained at AOI3 to classify potential subsurface features in AOI4, an area of unknown subsurface potential.

Results

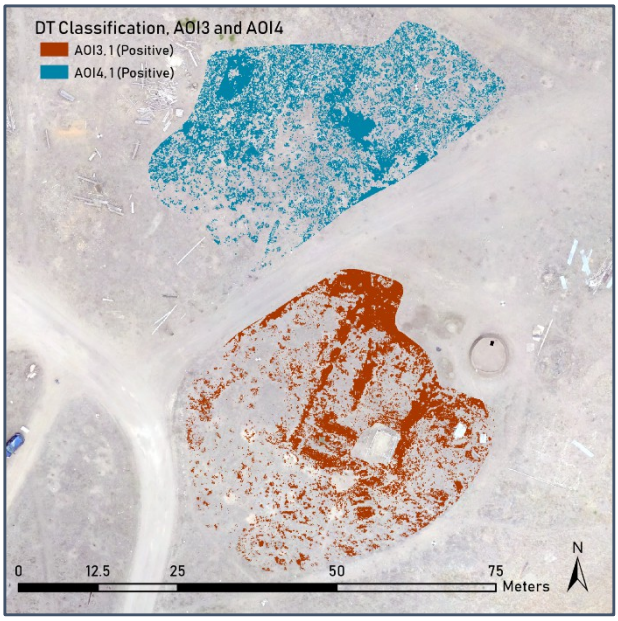
The trained SVM model achieves an overall classification score (F1 score) of 94% among the labeled testing data samples. The model confidently classifies the locations of architectural features in AOI3, demonstrating the efficacy of SVM analysis. However, applying this trained model to AOI4 is less successful. The classified output is characterized by significant noise, with only vague spatial patterning of positively classified subsurface features.

		Predicted Class		
		0 (Neg.)	1 (Pos.)	Recall
True Class	0 (Neg.)	3862	243	0.94
	1 (Pos.)	168	2523	0.94
	Precision	0.96	0.91	F1: 0.94



Similarly, the trained Decision tree model achieves an overall classification score (F1 score) of 91% among the labeled testing data samples, slightly less successful than the SVM model. Again, the model captures the locations of architectural features in AOI3, but yields significant noise in the AOI4 classification.

		Predicted Class		
		0 (Neg.)	1 (Pos.)	Recall
True Class	0 (Neg.)	3765	340	0.92
	1 (Pos.)	254	2437	0.91
	Precision	0.94	0.88	F1: 0.91



The modest success of these models could be due to several factors. First, the possibility that there are simply not any subsurface features in this area. Second, that the models trained on AOI3 is overfit to these data and does not translate well to AOI4. If they can be generalized, even if at the expense of overall classification score, it may achieve improved results in the AOI4 classification.

Future development of the project will continue in pursuit of the stated goals. Results might be improved through refinement of model parameters, improved labeled training data, additional feature engineering, or application of different machine learning models. While support-vector machines and decision trees have been explored in the current project iteration, convolutional neural networks (via `TensorFlow`) might also be well suited to this classification problem, as they are able to consider the contextual setting of each pixel relative to its neighbors. Nevertheless, the success of the project in pursuit of its primary goal indicates vast potential for thermal remote sensing and machine learning methods in archaeological and geospatial research.