Automatic Classification of Felsic, Mafic, and Ultramafic Rocks in Satellite Images from Palmira and La Victoria, Colombia

Saulo Bosquez[1], Germán H. Alférez[4(✉)], Ana María Martínez Ardila[2], and Benjamin L. Clausen[3]

[1] Facultad de Ingeniería y Tecnología, Universidad de Montemorelos, Av. Libertad 1300 Poniente, Barrio Matamoros, 67530 Montemorelos, N.L., Mexico
1180800@alumno.um.edu.mx

[2] Department of Earth and Biological Sciences, Loma Linda University, Griggs Hall, 11065 Campus Street, Loma Linda, CA 92350, USA
anmartinez@llu.edu

[3] Geoscience Research Institute, 11060 Campus Street, Loma Linda, CA 92350, USA
bclausen@llu.edu

[4] School of Computing , Southern Adventist University , PO Box 370, Collegedale, TN 37315-0370, USA
harveya@southern.edu

Abstract. Manually inspecting and analyzing satellite images can lead to numerous errors and is quite time consuming. Our geological contribution is to offer a means for the automatic classification of areas with felsic, mafic, and ultramafic rocks via machine learning using satellite images from Palmira and La Victoria, Colombia. Specifically, this study focuses on two types of satellite images taken from the Earth Observation System (EOS), namely natural color (bands B04 B03 B02) and infrared color vegetation (B08 B04 B03). The following machine learning algorithms were used in this study: Random Forest, K-Nearest Neighbors, Support Vector Machines, Logistic Regression, and Multilayer Perceptron. The model generated with K-Nearest Neighbors performed best for classifying natural color images with an accuracy of 91%, a precision of 87%, and a recall of 88%. Random Forest was the best model for classifying infrared images with an overall accuracy of 83%, a precision of 31%, and a recall of 31%.

Keywords: Machine learning · Geology · Rock classification · Random Forest · K-Nearest Neighbors · Support Vector Machines · Logistic Regression · Multilayer Perceptron · GDAL · OGR · Satellite images · Infrared images

1 Introduction

The study of ophiolites in the Palmira and La Victoria regions of Colombia is indispensable for understanding the formation and accretion of land that make up western Colombia and the orogenesis of the West margin of South America during the Cretaceous and Cenozoic. According to [20], ophiolitic type and occurrence in the Colombian Andes have been defined via previous studies. Ophiolitic bodies that have been studied and characterized are: the ultramafic body of Los Azules, El Complejo Ofiolítico

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
K. Arai (Ed.): SAI 2022, LNNS 507, pp. 531–547, 2022.
https://doi.org/10.1007/978-3-031-10464-0_36

de Pácora, the mafic-ultramafic complex of Bolívar-Valle, and the Ginebra Ophiolitic Complex (GOC). For the Amaíme Complex and the Buga Batholith, and the GOC petrographic and geochemical studies have been done on the banded gabbros in the ophiolitic sequence, and they have been interpreted in conjunction with previously published geochemical analysis of other parts of the ophiolitic sequence [20]. The ultramafic rocks in the study area are represented mainly by pyroxenite and peridotite bodies. The mafic rocks consist mainly of basalt, gabbro, and diorite bodies. Gabbro exists in the isotropic and layered forms. The felsic rocks are represented by quartz diorite and tonalite. These studies have analyzed and classified rock types from a "micro perspective", i.e., geochemical analysis. However, the Palmira and La Victoria area have not been analyzed from a "macro perspective". In other words, through remote sensing, the analysis of satellite images from that area.

Given that classification and mapping of different types of ophiolitic rocks by using satellite images can be resource-intensive and time consuming, the contribution of this study is to analyze satellite images from Palmira and La Victoria, Colombia from a macro perspective, using the following machine learning models to analyze satellite images: Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Multilayer Perceptron (MLP). This study represents the first attempt for mapping an ophiolite zone via remote sensing and machine learning in the West margin of Colombia, which can potentially be applied in the other ophiolites of the Colombian Andes.

The algorithms were trained with two types of satellite imagery, namely natural color and infrared color vegetation images. On the one hand, a natural color band combination allows for ground features to appear in colors similar to their appearance to the human eye; this band combination provides good sediment information. On the other hand, infrared color vegetation is an extremely popular band combination and has numerous uses, including studying vegetation studies, monitoring drainage, identifying soil patterns and multiple stages of crop growth [1]. Images used in this research were sourced from the Earth Observation System[1], which is an open-source image site. This site provides different options such as band combination, cloud cover, and time frame. The models were evaluated in terms of accuracy, precision, recall, and F1 score.

This paper is structured as follows. The second section presents the underlying concepts of our approach. The third section presents related work. The fourth section presents the methodology. The fifth section presents the results. The sixth section presents the discussion. The last section presents the conclusions and future work.

2 Theoretical Foundation

Our approach is based on the following concepts (see Fig. 1).

Igneous Rocks

Igneous rocks result from cooling and solidification of magma. These rocks can be volcanic or plutonic, depending on whether they solidify quickly at the surface or slowly in the Earth's crust [23]:

[1] https://eos.com/products/landviewer/

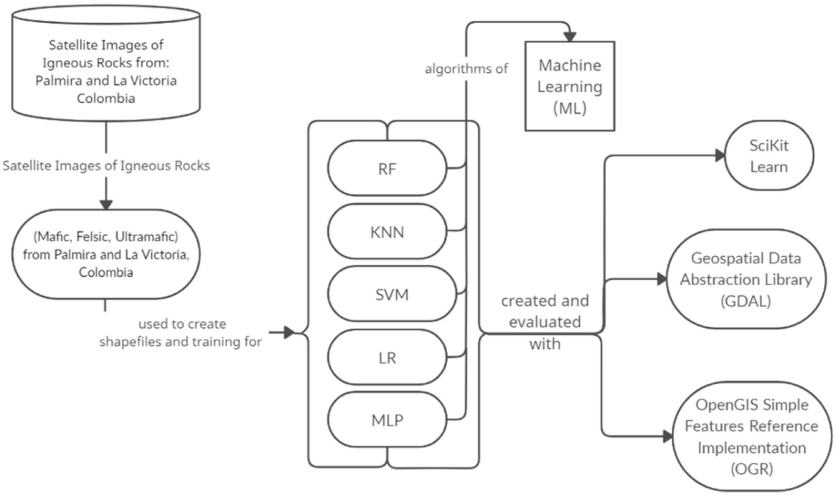

Fig. 1. Underpinnings of our approach

- In a widely accepted silica-content classification scheme, rocks with more than 65% silica are called felsic [9].
- Mafic rocks have a dark mineral content of 50 to 90% [13].
- Ultramafic rocks have a dark mineral content greater than 90% [13].

Machine Learning

Machine learning can be defined as the science (and art) of programming computers so they can learn from data. It is the field that gives computers the ability to learn without being explicitly programmed [14]. In this research work, we focus on supervised learning. In supervised learning, the training set that is given to the algorithm(s) includes the desired solutions (i.e., the classes). The classification algorithms that were used here are as follows:

- KNN is a machine learning technique that can be used for both regression and classification tasks. KNN examines the labels of a chosen number of data points surrounding a target data point, in order to make a prediction about the class that the data point falls into [17]. KNN is computed at the moment of prediction, and not when first fitting the data into the model. When a new data point arrives, the KNN algorithm starts by finding the nearest neighbors of this new data point. Once it has the values for neighbors of the data point, it uses them as a prediction for the new data point [19].
- RF combines the output of multiple decision trees to reach a single result. Due to its usability and flexibility, it has become a commonly used algorithm as it handles both classification and regression problems [8].
- SVM is a relatively simple supervised learning algorithm that is used for classification and/or regression, although it is preferred for classification. SVM finds a hyper-plane

that creates a boundary between the types of data. SVMs are a decent and straightforward example of a supervised classification technique for remote sensor data classification [7]. This technique is suitable for distinguishing the patterns as well as the objects that are utilized for pixel-based as well as object-based classification.

- LR models a relationship between predictor variables and a categorical response variable. LR helps estimate a probability of falling into a certain level of the categorical response given a set of predictors [22].
- Multilayer Perceptron (MLP) is the simplest kind of feed-forward networks. Within this algorithm, artificial neurons are arranged into a set of layers, each containing the exact same number of units. Every artificial neuron in one layer is connected to every unit in the next layer. Since this algorithm is built upon many layers, the first layer is called the input layer, the middle layers are called the hidden layers, and the last layer is called the output layer [15].

Satellite Images

EOS offers access to numerous image types such as: natural color, infrared color, and false color. It also offers filters such as: cloud cover, region specific, and satellite sensors. The band combinations chosen in this research work were natural color B04, B03, and B02 and infrared color (vegetation) B08, B04, and B03. The natural band allowed us to observe ground features as they would appear to the human eye. Healthy vegetation would appear as green. Unhealthy vegetation would be brown and yellow. The color infrared (vegetation) presents the standard false color composite. Vegetation appears in shades of red, while urban areas appear as cyan, and soils would vary from dark to light browns.

EOS's website offers a filter that allows control of the percentage of cloud cover in the images. This option is most beneficial to studies using satellite images since having the least amount of visual noise returns better results.

Underlying Technologies

Scikit-learn

Scikit-learn is a Python library that incorporates a wide range of state-of-the-art machine learning algorithms. The aim of this library is to provide machine learning to a more general audience using straightforward high-level language. One highlight of this library is the ease of use, performance, documentation, and application Programming Interface (API) consistency. Its dependencies are few and it is distributed under the simplified Berkeley Source Distribution (BSD) license, which encourages its use in both academic and commercial settings [18].

Quantum Geographic Information System

QGIS is a free, open-source, cross-platform compatible, and scalable GIS tool. Today, QGIS is geographic information processing software that is popular with many users. This software makes it possible to collect, store, process, analyze, manage, and present all types of spatial and geographic data comparable to other available, high-priced software [3].

Geospatial Data Abstraction Library
GDAL is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source License by the Open Source Geospatial Foundation. As a library, it presents a single raster abstract data model and single vector abstract data model callable for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing [12].

OpenGIS Simple Features Reference Implementation
OGR uses drivers to access files in different formats. OGR is a library focusing on vector data. It used to be a separate vector IO library inspired by OpenGIS Simple Features which was separated from GDAL; after the release of GDAL 2.0, GDAL and OGR components were integrated [12].

3 Related Work

Within the last two decades, satellite images have been beneficial to many studies in the geosciences. In this context, satellite image classification has not only become the appropriate choice for these types of studies, it has become the right choice [6]. This section presents relevant research works with different machine learning algorithms applied to satellite images. Table 1 summarizes the work presented in this section.

In [2], the authors present a detailed comparison of various techniques utilized in image processing with the goal of analyzing satellite images. One issue was that the images were affected by noise and other environmental conditions. In order to clear this visual noise, it was necessary to process the images so that they could be used for analysis. This research work indicates that satellite images are widely used in many real-time applications such as agriculture, land detection, navigation, and in geographical information systems. Some of the most popular machine learning based image processing techniques are presented, namely Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Cuckoo Search (CS) and PSO. The techniques are compared, and image processing limitations are described. The different metrics for performance evaluation in each of the image processing areas is studied.

In [11], a systematic analysis of satellite image-based land cover classification techniques was conducted. Accurate and effective techniques are required for classification to provide meaningful information regarding climate change, bio-diversity variation, and so on. The authors indicated that one of the most interesting research areas is satellite image-based land cover classification. Remotely sensed data obtained from remote sensors are capable of providing easily accessible data. Within this area, the authors categorized research works based on different classification, such as Fuzzy Random Forest, SVM, ANN, Bayesian Model, DT, and so on.

In [16], a system for land use mapping by machine learning is proposed. Governments, the private sector, research agencies, and community groups rely on land use mapping data for natural resource assessment, monitoring, and planning. Finding an effective mapping approach is thereby crucial for natural resource condition monitoring and investment, agricultural productivity, sustainability and planning, biodiversity conservation, natural disaster management, and biosecurity. The four machine learning algorithms used to classify satellite images for land use were: KNN, SVM, Convolutional Neural Network, and a Capsule Network. In addition, the implemented algorithms

Table 1. Comparison table of the articles described

Author	Area of study	Year	Models used	Results
Asokan et al. [2]	Machine learning based image processing techniques for satellite image analysis—a survey	2019	Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Cuckoo Search (CS) and PSO	GA and PSO suffer from the limitation that they get trapped in local minima. CS and PSO has shown better results in terms of closeness to optimal solution when compared to PSA or GA individually
Gavade et al. [11]	Systematic analysis of satellite image-based land cover classification techniques: literature review and challenges	2019	Clustering, SVM, Learning, Decision Tree, Neural Network, Hierarchical, Fuzzy, Spatial, Bayesian Model, Other Classifier	Accuracy from 50 selected papers: Clustering: 8% SVM: 14% Learning: 10% Decision Tree: 2% Neural Network: 8% Hierarchical: 10% Fuzzy: 20% Spatial: 6% Bayesian Model: 10% Other Classifier: 12%
Liao et al. [16]	ML-LUM: A system for land use mapping by machine learning algorithms	2019	KNN, SVM, CNN, CapsNet	Accuracy (Satellite Dataset): KNN: 54% SVM: 75% CNN: 98.09% CapsNet: 96%
Tangthaikwan et al. [21]	Multiclass support vector machine for classifying spatial data from satellite image	2017	MLP4, MLP36, SVM Standard, SVM Max	Accuracy: MLP4: 87.25% MLP36: 89.32% SVM Standard: 87.27% SVM Max: 90.89%

(*continued*)

Table 1. (*continued*)

Author	Area of study	Year	Models used	Results
Garosi et al. [10]	Assessing the performance of GIS-based machine learning models with different accuracy measures for determining susceptibility to gully erosion	2019	RF, SVM, NB, GAM	RF AUC: 92.4% SVM AUC: 90.9% NB AUC: 87.2% GAM AUC: 89.9%
Bérubé et al. [4]	Predicting rock type and detecting hydrothermal alteration using machine learning and petrophysical properties of the Canadian Malartic ore and host rocks, Pontiac Subprovince, Québec, Canada	2018	SVM	Overall Precision: 89% Overall Recall: 89% F1 Scores: Meta-sedimentary rocks: 73% Felsic-intermediate Intrusive rocks: 69% Mafic dykes: 93%
Chakouri [5]	Geological and mineralogical mapping in Moroccan central Jebilet using multispectral and hyperspectral satellite data and Machine Learning	2020	SVM	Accuracy: Hyperspectral: 93.05% Multispectral: 89.24%

were also modified for land use mapping in a Machine Learning Use Mapping system. This system is able to train models, predict classifications of satellite images, map the land use, display the land use statistical data and predict production yields.

In [21], a multiclass SVM is used to classify spatial data from satellite images. The image is pre-processed and classified using SVM with the Radial Basis Function (RBF) Kernel. A pixel-based classification method is performed according to the value of spectral pixels with a Multi-Spectral Scanner satellite image and the data used comes from a 3X3 square neighborhood. The research process consists of two stages. In the first stage, the RBF kernel is applied. In the second stage, the classification result is compared with other classification methods. The result with SVM got a higher accuracy compared to other methods.

In [10], the performance of machine learning models applied to GIS resulted in different accuracy values. This was used to determine susceptibility to gully erosion in the study area. With the help of extensive field surveys and GPS data, digital maps were prepared. Topographical attributes were provided from digital elevation models (DEM). The land use and normalized difference vegetation index (NDVI) marks were created by satellite imagery. The functional relationships between gully erosion and controlling factors were calculated using Random Forest, SVM, Naive Bayes, and generalized additive models. The results showed that the RF model had the highest amount of efficiency, Area Under the Curve (AUC), and lowest amounts of mean absolute error (MAE) and root mean square error (RMSE) compared with SVM, NB, and Generalized Additive Model (GAM).

In [4], machine learning was used on rock samples to predict rock type and hydrothermal alteration of the Canadian Malartic ore and host rocks. Various rock types are present in the Malartic District, mainly meta-sedimentary rocks, felsic intermediate intrusive rocks, and mafic dykes. With SVM, it was found that these two physical properties can be used to predict the rock type of a sample with an average precision and recall rate of 89%. The SVM classifier was extended to predict whether meta-sedimentary rocks, felsic-intermediate intrusive rocks, and mafic dykes had undergone hydrothermal alteration with average F1 scores of 73%, 69%, and 93%, respectively. The machine learning process used in this case study can be applied in advanced exploration stages, and the recovered rock samples can be used to update trained prediction models.

In [5], geological and mineralogical mapping was done for the Moroccan central Jebilet using machine learning and multispectral/hyperspectral satellite data. The Moroccan central Jebilet Massif is one of the main Paleozoic outcrops in Morocco. The massif is characterized by its location in an arid climate, significant mining potential and the absence of plant cover, which favors the use of spatial remote sensing for geological mapping. The classification with SVM allowed the mapping of the lithological units in the study area. The accuracy of the SVM classification of hyperspectral data is higher than that of multispectral data, which was demonstrated by the confusion matrix, notably an overall accuracy of 93.05% and 89.24%, respectively. The use of hyperspectral and multispectral images has been shown to be a good technique for the characterization of iron deposits and lithological units, which may help in mineral exploration engineering with reduced need for fieldwork and geochemistry.

4 Methodology

This section describes the steps followed in this research work.

4.1 Acquiring Raw and Infrared Satellite Images

To begin, several images of natural and infrared colors were required. The images utilized for this research were sourced from EOS. Using the provided shapefile[2] 6 images for the Palmira and La Victoria area in Colombia were searched for and downloaded.

[2] https://github.com/SBosq/MUFRocks/tree/main/Colombia_Geo.

Using the provided options within EOS, three filters were applied: source indicating passive sensors (day, night, or low resolution) or active sensors or terrain tiles or EOS storage files or high-resolution imagery, cloudiness indicating how much cloud cover is desired, and sensor indicating which satellite should the images come from. One of the most important filters was cloud cover. This filter allowed a search of images that were mostly or completely free of visual noise. Six different images were selected by hand and are available online[3]. Their image quality was approximately 1394 × 1060 (60 m/px), meaning that there is a total of 1,477,640 pixels in each image, and each individual pixel covers 60 m. These images were downloaded in May 2021, in TIFF format. GeoTiff (.tif/.tiff) format is the most common raster data file type suitable for storage, transfer, display, and printing of raster images. GeoTiff supports black-and-white, gray scale, pseudo color, and true color images. The image quality of the used images was 661 × 1374, meaning that there was a total of 908,214 in the images. Processing this cropped

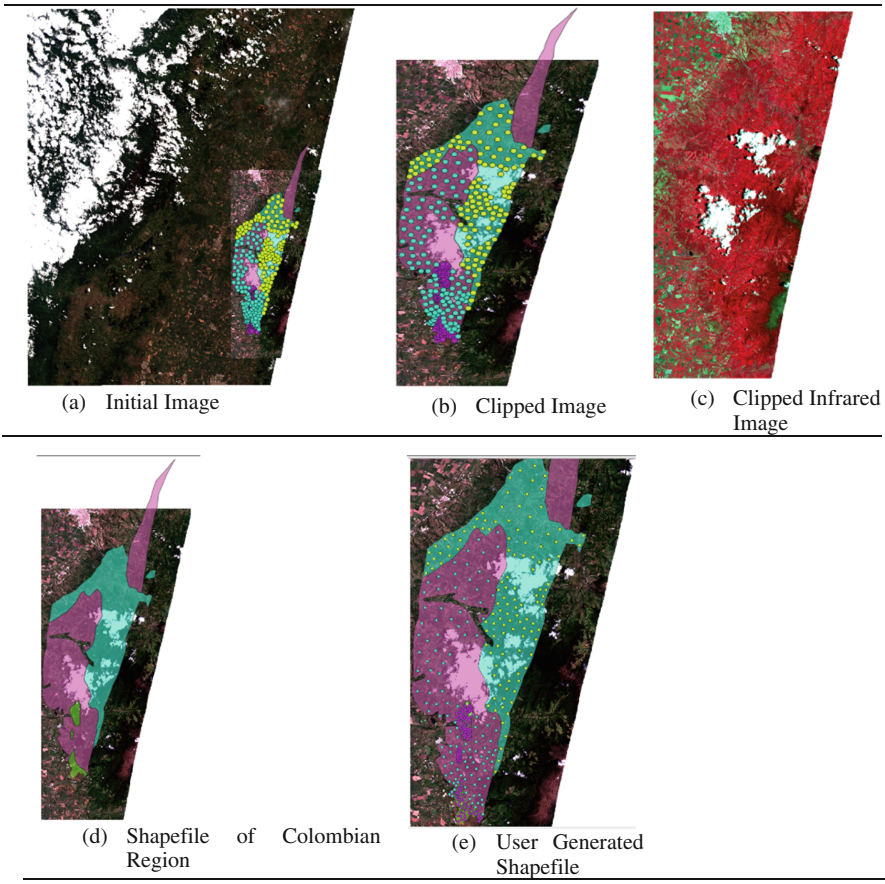

Fig. 2. Satellite images

[3] https://github.com/SBosq/MUFRocks/tree/main/InitialImages.

image took approximately 3.5 s. The images used in this work were not the full images that were downloaded from EOS, since processing the full downloaded image would have taken too much time, computing time, and resources. Figure 2(a) shows an example of one of the six initial satellite images that were downloaded. After the shapefile was fitted into the satellite image, the initial image was bigger than the focus area. Figure 2(b) shows the newly clipped satellite image of the focus area. The shapefile and its contents nicely fit into the clipped area. Figure 2(c) shows the focus area in a different band combination, this time focusing on the infrared aspect of the image. Figure 2(b) shows the focus area labeled. The first teal colored data points belong to the mafic classification. The purple data points belong to the ultramafic classification. Lastly, the yellow data points represent the felsic classification. Figure 2(e) provides a closer look at the clipped satellite image of the focus area with the shapefile (Figure 2(d)) superimposed on top of it.

Next, the images were renamed according to their band combination, whether natural color or infrared. All 6 images were then imported to QGIS in order to begin image segmentation and analysis.

4.2 Preparing and Processing the Data

In this step, the images were prepared and processed, one at a time. To this end, a Python script was created. This script is available online[4] and is shown in Listing 1. Line 1 from Listing 1 shows that the "colombia_fn" variable stores the cropped natural color image of our area of interest. The format used was TIFF. The reasoning behind this is that the files can be large in size and may contain high detail, and the image layers are merged when they are saved. Line 2 highlights the "segments_fn" variable, which stores the final segmented and clipped image in.tif format. Notice that the format is spelled differently in lines 1 and 2. However, it is important to state that there is no other difference between these formats than their spelling. In line 4, GDAL is used in the creation of the variable called "driverTiff" to utilize the necessary driver to translate the data found in the aforementioned .tiff and .tif files.

Next, our dataset is opened as read-only using the GDAL open function and loaded into the "colombia_ds" variable in line 5. Finally, the number of bands found in our image is counted and stored within the "nbands" variable in line 6.

In line 7, an empty list named "band_data" is created to store the data for all the bands found within the image. In line 11, the parameters for the loop that will read the band data are established. Since the bands found within the "colombia_ds" variable start at 1 and not 0, it is important to state that the range will begin at 1 and end at nbands +1 in order to avoid an indexing error. Lines 12–14 show the creation of a local variable within the for-loop called "band". This variable will store the individual data from each band as a Numpy array. Since the amount of data found within each band may be large in quantity, the utilization of a Numpy array is required. Specifically, a Numpy array provides better speed and consumes less memory space than Python lists. The array for the individual raster band is then appended to the empty "band_data" list. Line 15 shows the data from this list being stacked depth-wise and being reassigned

[4] https://github.com/SBosq/MUFRocks/blob/main/OSGDAL/main.py.

into the "band_data" variable. This step is important because it arranges the data into a single data structure containing rows, columns, and bands. In Listing 1, line 16 creates a new variable to store the normalized data, from values 0 to 1, from the "band_data" list using the function "rescale_intensity". In order to take into account, the amount of time that the segmentation of our image takes. Line 18 shows the creation of the "seg_start" variable that takes the time before the segmentation starts and stores it. In line 19, image segmentation is initialized using Simple Linear Iterative Clustering (SLIC) and stored in the "segments" variable. SLIC is a modern approach for segmenting superpixels, and it requires very little computing power. In this same line, the hyperparameters passed are "img", "n_segments", and "compactness". The variable "img" contains the normalized data from "band_data". Next, the number of segments, represented by "n_segments" was assigned the value of 68,250 after trial and observation. During this step it was realized that the image spanned more than the focus area, thus the image was downsized to an appropriate size. Once the image was clipped, it was stored in a local file, read, and stored into the "segments_fn" variable from line 2. The final hyperparameter found in this line is "compactness", which refers to the size of the segments to be created; this was set at 0.1. Once an image was segmented, the amount of time that the segmentation took is displayed on screen, as observed in line 21.

Listing 1: Image Processing

```
1   colombia_fn = 'Cropped_Colombia_Area_3.tiff'
2   segments_fn = \
3       'C:/temp/eosImages/segments_final.tif'
4   driverTiff = gdal.GetDriverByName('GTiff')
5   colombia_ds = gdal.Open(colombia_fn)
6   nbands = colombia_ds.RasterCount
7   band_data = []
8   print('bands', colombia_ds.RasterCount, 'rows',
9         colombia_ds.RasterYSize, 'columns',
10        colimbia_ds.RasterXSize)
11  for i in range(1, nbands + 1):
12      band = \
13          colombia_ds.GetRasterBand(i).ReadAsArray()
14      band_data.append(band)
15  band_data = np.dstack(band_data)
16  img = exposure.rescale_intensity(band_data)
17
18  seg_start = time.time()
19  segments = slic(img, n_segments=68250,
20                  compactness=0.1)
21  print('segments_complete', time.time() – seg_start)
```

Listing 2 presents the creation of the "segments_ds" variable that contains the same parameters as the "colombia_ds" variable. Listing 2, lines 1–4 show the "segments_ds"

variable, which is created and assigned a newly created.tiff file using the "Create" function from the GDAL "GetDriverByName" variable. The hyperparameters used in the "Create" function are the "segments_fn" variable. This variable is created in line 2 of Listing 1; the raster x size from the original.tiff file, as well as the raster y size from the.tiff file (loaded in line 1, Listing 1), the number of bands, and the data type. The data type utilized was GDT_Float32 to accommodate varying formats and element sizes found within the dataset. In line 5, the GDAL "SetGeoTransform" function is used to convert the map coordinates from "colombia_ds" to pixel coordinates. Line 7 utilizes the GDAL "SetProjection" function to identify "colombia_ds"'s projection information and assign it to the newly created.tiff file. Line 9 writes the data from "segments" into "segments_ds" as an array using GDAL's "WriteArray" function; line 10 closes the newly created.tiff file by simply equaling it to None.

Listing 2: Image Processing

```
1   segments_ds = driverTiff.Create(
2       segments_fn, Colombia_ds.RasterXSize,
3       colombia_ds.RasterYSize, 1,
4       gdal.GDT_Float32)
5   segments_ds.SetGeoTransform(
6       colombia_ds.GetGeoTransform())
7   segments_ds.SetProjection(
8       colombia_ds.GetProjectionRef())
9   segments_ds.GetRasterBand(1).WriteArray(segments)
10  segments_ds = None
```

Listing 3 presents the creation of a dataframe using the resulting feature names and values from the created shapefiles, as well as splitting the dataframe into two files, one for training and the other for testing the ML models. Listing 3, line 1 shows the "gdf" variable being created and assigned the truth data gathered from the shapefile; this data is read using a geodataframe. In line 3, the unique values from the geodataframe column 'RockTypes' is assigned to the "class_names" variable. The purpose of line 5 is to assign an integer value to each of the "class_names" variable. This is because rasters are not able to store strings. The Numpy function used here is ".arange", which creates an array from zero to n depending on the size of the variable passed. In our work, the "class_names" variable is used and increased by 1 so that all the values are not stored in an index value lesser than their original place. In line 7, a Pandas dataframe is assigned to the "df" variable. Also the dataframe is passed a dictionary, which contains the columns 'RockType' and 'id'. This dataframe is then converted to a CSV file and saved under the file name 'RType_lookup.csv' in line 9. Line 11 adds a column to the shapefile, via mapping, using dict(zip(class_names, class_ids)), which contains the IDs assigned in the CSV file. Lines 15 and 16 split the shapefile into two separate shapefiles, one reserved for training and the other used to test the model. Specifically, line 15 splits the data into two subsets. The split containing 70% of the data is used to train the models. The remaining 30% of the data is used to evaluated the models. Line 16 is then assigned the remaining thirty percent of the geodataframe that was loaded with the "Rock_data.shp"

shapefile. Lines 19 and 20 save these two files into separate shapefiles, train.shp and test.shp, respectively.

Listing 3: DataFrame Creation

```
1   gdf = gpd.read_file(
2         'C:/Users/saulo/Documents/Rock_data.shp')
3   class_names = gdf['RockTypes'].unique()
4   print('class_names', class_names)
5   class_ids = np.arange(class_names.size) + 1
6   print('class_ids', class_ids)
7   df = pd.DataFrame(
8       {'RockTypes': class_names, 'id': class_ids})
9   df.to_csv('C/temp/eosImages/RType_lookup.csv')
10  print('gdf_without_ids', gdf.head())
11  gdf['id'] = gdf['RockTypes'].map(
12        dict(zip(class_names, class_ids)))
13  print('gdf_with_ids', gdf.head())
14
15  gdf_train = gdf.sample(frac=0.7)
16  gdf_test = gdf.drop(gdf.train.index)
17  print('gdf_shape', gdf.shape, 'training_shape',
18          gdf_train.shape, 'test', gdf_test.shape)
19  gdf_train.to_file('C:/temp/eosImages/train.shp')
20  gdf_test.to_file('C:/temp/eosImages/test.shp')
```

5 Results

In this research, we used the following supervised-learning algorithms: RF, KNN, SVM, LR, and MLP to classify ultramafic, mafic, and felsic rocks. The source code of the algorithms that were used for training and evaluating the models is available online[5]. Table 2 summarizes the results in terms of accuracy values for the ultra mafic, mafic, and felsic rock types. Accuracy is the ratio of the number of correct predictions made to the number of all predictions made.

$$Accuracy = \frac{Correct\ Predictions}{Predictions\ Made}$$

In the KNN model, the number of neighbors was specified as four after several testings. In the SVM model, the RBF kernel was applied. In the RF model, 100 trees were used in the random forest. In the LR model, the "liblinear" optimizer was chosen because of the small training dataset. In the MLP model, the "lbfgs" optimizer was chosen because it has better performance and convergence than the other MLP optimizers.

[5] https://github.com/SBosq/MUFRocks/tree/main/neighbortest.

Table 2. Class accuracy

	Natural color image	Infrared color image
	[ULTRA MAFIC, MAFIC, FELSIC]	[ULTRA MAFIC, MAFIC, FELSIC]
RF	[76%, 87%, 83%]	[82% 80% 81%]
KNN	[86%, 95%, 93%]	[46% 59% 52%]
SVM	[56% 70% 79%]	[0% 47% 40%]
LR	[58%, 44%, 71%]	[0% 48% 0%]
MLP	[78%, 81%, 83%]	[26% 51% 36%]

As one can see from Table 1, natural color images were successfully able to be classified and yielded better results, whereas there are instances where the infrared rock datasets were not able to be processed by the models created. Also, precision, recall and F-score values were evaluated for each model.

Precision is the number of correct positives results, divided by the number of positive results predicted.

$$Precision = \frac{TP}{TP + FP}$$

Recall is the number of correct positive results, divided by the number of all relevant samples (all the samples that should be classified as positive).

$$Recall = \frac{TP}{TP + FN}$$

F-score is the harmonic mean between precision and recall. This number, which is in the [0,1] range, indicates how precise the classifier is (precision) and how robust it is (recall). The greater the F1 score, the better the overall performance of the model.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Table 3 presents the classification report in terms of precision, recall, and F1 score for natural color images and infrared color images. The best overall results for both image types used the RF model, while the best results for each image were KNN and RF for raw images and infrared images, respectively.

Table 3. Classification report models in terms of precision, recall, and F1 score

RF							
	Natural color image				**Infrared color image**		
	Precision	Recall	F1 score		Precision	Recall	F1 score
Ultramafic	0.77	0.89	0.83	Ultramafic	0.83	0.71	0.77
Mafic	0.90	0.73	0.81	Mafic	0.79	0.82	0.81
Felsic	0.79	0.82	0.81	Felsic	0.82	0.91	0.86
KNN							
	Natural color image				**Infrared color image**		
	Precision	Recall	F1 score		Precision	Recall	F1 score
Ultramafic	0.86	1.00	0.92	Ultramafic	0.30	0.36	0.33
Mafic	0.94	0.71	0.81	Mafic	0.50	0.60	0.54
Felsic	0.82	0.93	0.87	Felsic	0.36	0.22	0.27
SVM							
	Natural color image				**Infrared color image**		
	Precision	**Recall**	**F1 score**		**Precision**	**Recall**	**F1 score**
Ultramafic	0.47	0.26	0.33	Ultramafic	0.00	0.00	0.00
Mafic	0.34	0.56	0.42	Mafic	0.46	0.99	0.63
Felsic	0.86	0.76	0.81	Felsic	0.36	0.01	0.01
LR							
	Natural color image				**Infrared color image**		
	Precision	Recall	F1 score		Precision	Recall	F1 score
Ultramafic	0.54	1.00	0.70	Ultramafic	0.00	0.00	0.00
Mafic	0.67	0.06	0.12	Mafic	0.50	1.00	0.66
Felsic	0.85	0.89	0.87	Felsic	0.00	0.00	0.00
MLP							
	Natural color image				**Infrared color image**		
	Precision	Recall	F1 score		Precision	Recall	F1 score
Ultramafic	0.78	0.96	0.86	Ultramafic	0.32	0.05	0.08
Mafic	0.94	0.61	0.74	Mafic	0.51	0.90	0.65
Felsic	0.87	1.00	0.93	Felsic	0.38	0.14	0.20

6 Discussion

According to Table 2, for natural color images the KNN model performed best, with an accuracy score of 87.04%. Table 3 presents the classification report of the natural color image using the KNN model. From this it can be observed that the precision scores for

each individual class are: 77% for ultramafic, 90% for mafic, and 79% for felsic rock types. The recall scores for the KNN model using the natural color image are 89% for ultramafic, 73% for mafic, and 82% for felsic rock types. The F1 Score values for the KNN model using the natural color image are 83% for ultramafic, 81% for mafic, and 81% for felsic rock types. Focusing on the F1 Score for each of the classes for the KNN model using the natural color image, it can be determined that KNN was in fact the best performing model for natural color image classification.

According to Table 2, for infrared images, the RF model was the best performing one, with an accuracy score of 82%. Table 3 presents the classification report of the infrared image using the RF model. From this, it can be observed that the precision scores for each individual class is: 83% for ultramafic, 79% for mafic, and 79% for felsic rock types. The recall scores for the RF model using the infrared image are 71% for ultramafic, 82% for mafic, and 79% for felsic rock types. The F1 Score values for the RF model using the infrared image are 77% for ultramafic, 81% for mafic, and 86% for felsic rock types. Upon further observation of the F1 Score of each class, it can be determined that RF was in fact the best performing model for infrared image classification.

7 Conclusions and Future Work

We proposed the use of five machine learning models for rock type classification of felsic, mafic, and ultramafic rocks, using satellite images of the Palmira and La Victoria area in Colombia. The best overall results, for both image types, used the RF model, while the best results, for the images were the ones obtained with the KNN and RF models.

In future work, we expect to utilize more images and shapefiles obtained from future visits to the area of study. Also, we expect to test our classifier in the field to corroborate the validity of the results obtained in the lab. In order to test our results, fieldwork, remote sensing technique and the study of the spectral signature of the rocks should be considered. The combination of macro results (i.e., obtained with the analysis of satellite images) with micro results (i.e., obtained with the analysis of samples in the field), can prove useful for future work. Also, we expect to analyze hyperspectral images obtained from the area of study using a drone in order to have more detailed data.

References

1. Earth Observing Data Analytics Inc.: EOS LandViewer: browse real-time Earth observation (2021). https://eos.com/products/landviewer/
2. Asokan, A., Anitha, J.: Machine learning based image processing techniques for satellite image analysis—a survey. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (2019)
3. Baghdadi, N., Zribi, M., Mallet, C.: QGIS and Generic Tools. ISTE LTD. ISBN: 978-1-78630-187-1 (2018)
4. Bérubé, C.L., et al.: Predicting rock type and detecting hydrothermal alteration using machine learning and petrophysical properties of the Canadian malartic ore and host rocks, Pontiac Subprovince, Québec, Canada. Ore Geol. Rev. **96**, 130–145 (2018)

5. Chakouri, M.: Geological and mineralogical mapping in Moroccan central Jebilet using multispectral and hyperspectral satellite data and machine learning. Int. J. Adv. Trends Comput. Sci. Eng. **9**(4), 5772–5783 (2020)
6. Lu, D., Weng, Q.: A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. **28**(5), 823–870 (2007). https://www.tandfonline.com/doi/pdf/10.1080/01431160600746456?needAccess=true
7. Dhingra, S., Kumar, D.: A review of remotely sensed satellite image classification. Int. J. Electr. Comput. Eng. **9**, 1720 (2019)
8. IBM Cloud Education: What is Random Forest? (2021). https://www.ibm.com/cloud/learn/random-forest
9. Encyclopedia Britannica: Felsic and mafic rocks - igneous rock (2021). https://www.britannica.com/science/felsic-rock
10. Garosi, Y., Sheklabadi, M., Conoscenti, C., Pourghasemi, H.R., Oost, K.V.: Assessing the performance of GIS-based machine learning models with different accuracy measures for determining susceptibility to gully erosion. Sci. Total Environ. **664**, 1117–1132 (2019)
11. Gavade, A.B., Rajpurohit, V.S.: Systematic analysis of satellite image-based land cover classification techniques: literature review and challenges. Int. J. Comput. Appl. **43**(6), 514–523 (2019)
12. GDAL/OGR: GDAL/OGR Geospatial Data Abstraction Software Library, Open Source Geospatial Foundation. https://gdal.org (2021)
13. Geologyin: How to classify igneous rocks into (ultramafic, mafic, intermediate and felsic)? (2021). https://www.geologyin.com/2014/12/how-to-classify-igneous-rocks-into.html
14. Geron, A.: Hands-on Machine Learning With SciKit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media Inc, Sebastopol, CA (2017)
15. Grosse, R.: Lecture 5: Multilayer perceptrons (2018). http://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/readings/L05%20Multilayer%20Perceptrons.pdf
16. Liao, X., Huang, X., Huang, W.: ML-LUM: a system for land use mapping by machine learning algorithms. J. Comput. Lang. **54**, 100908 (2019)
17. Nelson, D.: What is a KNN (k-nearest neighbors)? (2020). https://www.unite.ai/what-is-k-nearest-neighbors/
18. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 3426 (2011)
19. Korstanje, J.: The k-nearest neighbors (kNN) Algorithm in Python (2021). https://realpython.com/knn-python/
20. Rodríguez Ramos, B.P.: Estudio metalgenético de las mineralizaciones auríferas del área de Ginebra y zonas aledañas, Valle del Cauca Universidad Nacional de Colombia (2012). https://repositorio.unal.edu.co/handle/unal/21336
21. Tangthaikwan, K., Keeratipranon, N., Agsornintara, A.: Multiclass support vector machine for classification spatial data from satellite image. Multiclass support vector machine for classification spatial data from satellite image (2017)
22. Pennsylvania State University: 12.1—logistic regression | stat 462 2021 (2021). Available: https://online.stat.psu.edu/stat462/node/207/
23. Vera Torres, J.A.: RACEFN Glosario de Geología (2009). http://www.ugr.es/~agcasco/personal/rac-geologia/0-rac.htm