ANKARA YILDIRIM BEYAZIT UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

COMPUTER ENGINEERING

Generating Land Cover Maps for Unseen Data

AYBU CENG463 Machine Learning Project, Fall 2022

by

Buğra Alptekin Sarı (20050811028)

Rukiye Kurt (19050161001)

Supervised by

Dr. Mustafa Teke

25.01.2023

ABSTRACT

Within the scope of the project, land cover maps were generated for unseen data using a limited number of training samples in the Gibraltar area.

Table of Contents

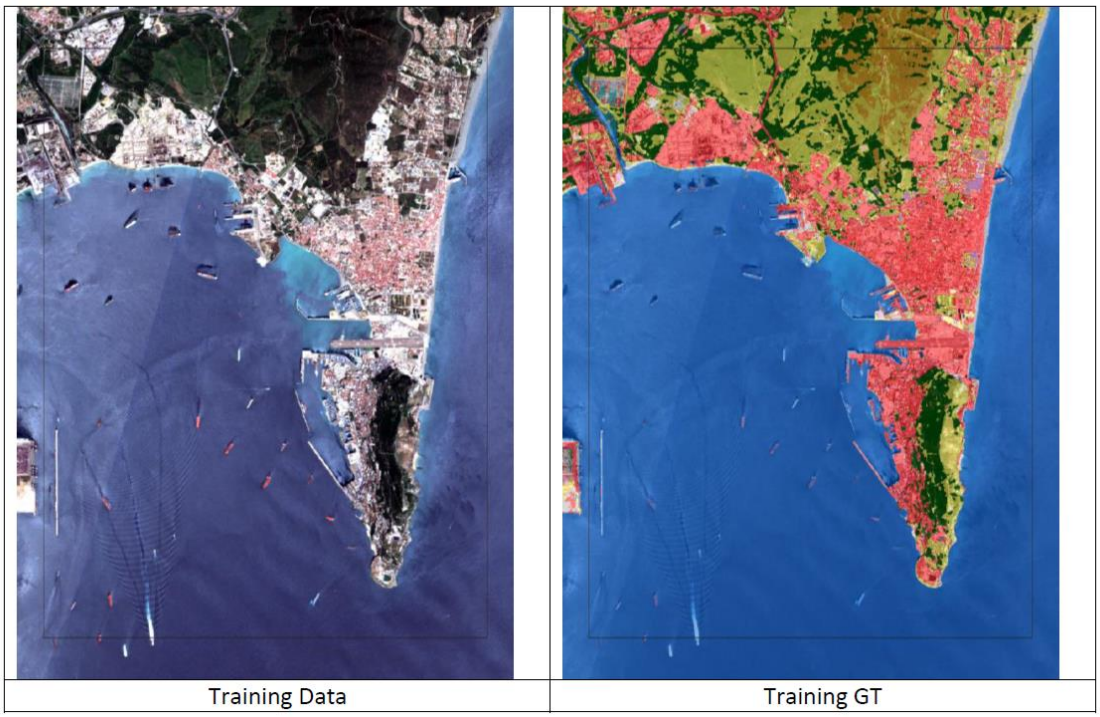
1. INTRODUCTION

Land cover refers to the surface cover on the ground like tree cover, shrubland, grassland, cropland, built-up, bare/sparse vegetation, snow and ice, permanent water bodies, herbaceous, mangroves, moss and lichen. The table regarding the data obtained from European Space Agency (ESA) is below:

Map code	Land Cover Class	LCCS code	Definition	Color code (RGB)
10	Tree cover	A12A3 // A11A1 A24A3C1(C2)-R1(R2)	This class includes any geographic area dominated by trees with a cover of 10% or more. Other land cover classes (shrubs and/or herbs in the understorey, built-up, permanent water bodies, ...) can be present below the canopy, even with a density higher than trees. Areas planted with trees for afforestation purposes and plantations (e.g. oil palm, olive trees) are included in this class. This class also includes tree covered areas seasonally or permanently flooded with fresh water except for mangroves.	0,100,0
20	Shrubland	A12A4 // A11A2	This class includes any geographic area dominated by natural shrubs having a cover of 10% or more. Shrubs are defined as woody perennial plants with persistent and woody stems and without any defined main stem being less than 5 m tall. Trees can be present in scattered form if their cover is less than 10%. Herbaceous plants can also be present at any density. The shrub foliage can be either evergreen or deciduous.	255, 187, 34
30	Grassland	A12A2	This class includes any geographic area dominated by natural herbaceous plants (Plants without persistent stem or shoots above ground and lacking definite firm structure): (grasslands, prairies, steppes, savannahs, pastures) with a cover of 10% or more, irrespective of different human and/or animal activities, such as: grazing, selective fire management etc. Woody plants (trees and/or shrubs) can be present assuming their cover is less than 10%. It may also contain uncultivated cropland areas (without harvest/ bare soil period) in the reference year	255, 255, 76
40	Cropland	A11A3(A4)(A5) // A23	Land covered with annual cropland that is sowed/planted and harvestable at least once within the 12 months after the sowing/planting date. The annual cropland produces an herbaceous cover and is sometimes combined with some tree or woody vegetation. Note that perennial woody crops will be classified as the appropriate tree cover or shrub land cover type. Greenhouses are considered as built-up.	240, 150, 255
50	Built-up	B15A1	Land covered by buildings, roads and other man-made structures such as railroads. Buildings include both residential and industrial building. Urban green (parks, sport facilities) is not included in this class. Waste dump deposits and extraction sites are considered as bare.	250, 0, 0
60	Bare / sparse vegetation	B16A1(A2) // B15A2	Lands with exposed soil, sand, or rocks and never has more than 10 % vegetated cover during any time of the year	180, 180, 180
70	Snow and Ice	B28A2(A3)	This class includes any geographic area covered by snow or glaciers persistently	240, 240, 240
80	Permanent water bodies	B28A1(B1) // B27A1(B1)	This class includes any geographic area covered for most of the year (more than 9 months) by water bodies: lakes, reservoirs, and rivers. Can be either fresh or salt-water bodies. In some cases the water can be frozen for part of the year (less than 9 months).	0, 100, 200
90	Herbaceous wetland	A24A2	Land dominated by natural herbaceous vegetation (cover of 10% or more) that is permanently or regularly flooded by fresh, brackish or salt water. It excludes unvegetated sediment (see 60), swamp forests (classified as tree cover) and mangroves see 95)	0, 150, 160
95	Mangroves	A24A3C5-R3	Taxonomically diverse, salt-tolerant tree and other plant species which thrive in intertidal zones of sheltered tropical shores, "overwash" islands, and estuaries.	0, 207, 117
100	Moss and lichen	A12A7	Land covered with lichens and/or mosses. Lichens are composite organisms formed from the symbiotic association of fungi and algae. Mosses contain photo-autotrophic land plants without true leaves, stems, roots but with leaf-and stemlike organs.	250, 230, 160

Data obtained through satellites is very important in land cover applications. These data need to be analyzed carefully so that we can get useful results. The main purpose of the project is to classify the land cover of the data obtained from the satellite by using the methods and algorithms covered in the Machine Learning Course (CENG 463). For this purpose, satellite images of the Gibraltar area were used as data. Firstly, studies were carried out to understand the data. Later, different models were developed based on the training data and label given below.

Data:



Training Data	Training GT

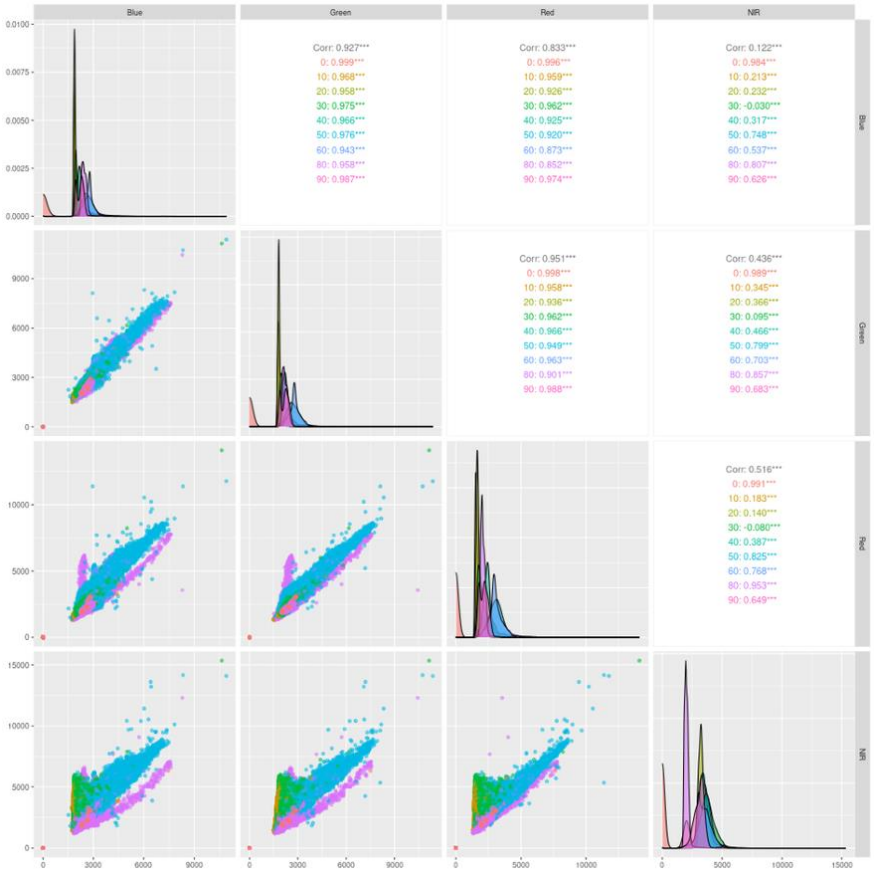
Also, the github link of our project : *github.com/BugraAlptekinSari/landuse-classifier*
Contributor : *github.com/RLinaK*

2. DATA

The first task is to understand the data we have. It was tried to visualize the data to understand what it is and to distinguish it from each other. Since it was not possible to draw graphics in the Colab environment, a Jupyter server was installed on the Linux machine and tried there. The computer locked up on each attempt. It was necessary to restart the

computer. Then, The RStudio Server was installed on the Linux machine. It was tried to draw the graph and again the computer locked up. After about six iterations it was discovered that results can be obtained without using the graphical interface. The results were obtained by writing the graph into a file.

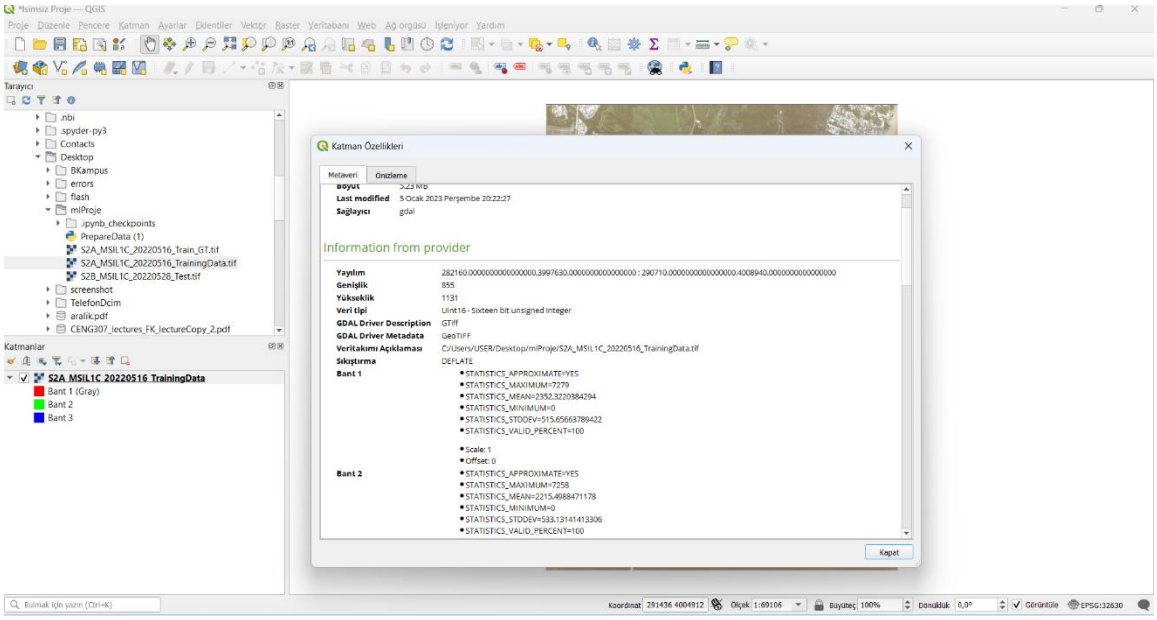
Below is the graph drawn to analyse the data in a meaningful way. The graph shows the colour distribution of the labels.



This graphic was drawn in R studio environment using phyton language. Below is the image of the code where this graph is drawn.

```r
# Download and install required packages
required_packages <- c("ggplot2", "GGally", "readr", "grDevices") # c() is the function used to make new vectors in R
new_packages <- required_packages[!(required_packages %in% installed.packages()[,"Package"])] # Filter out the packages installed
if(length(new_packages)) install.packages(new_packages) # Install missing packages if any.

# Load the installed packages into the namespace.
library("ggplot2")  # The plotting library
library("GGally")   # An add on for the plotting library which includes a function to draw the pairplot.
library("readr")    # Reads the csv file.
library("grDevices")# Includes the png() and dev.off()

print("Reading data.")
# Read the csv file and specify the correct data types for it.
# factor data type is a special data type used for repetitive classes with a look-up table. It is used here to let R know how many classes we have.
train <- read_csv("train.csv", col_types = cols(...1 = col_skip(), Code = col_factor(levels = c("0", "10", "20", "30", "40", "50", "60", "70", "80", "90"))))

print("Drawing the Plot.")

# This function makes it so any graphical input is written into a png file.
png("pairs.png",width=1024,height=1024)

# Overloaded print function to draw the plot into the png file.
# aes() function is used specifically for aesthetic arguments like colour and translucency(alpha) in here.
print(ggpairs(train,columns = 2:5, aes(colour = Code, alpha = 0.4)))
# Close Graphical device (PNG file) to write the finished plot into the file.
dev.off()

# Debug function to make sure nothing went wrong.
print("Plotting Done.")
```

Similarly, the data was analyzed in the QGIS environment and tried to be understood.



Besides, information about the data was also obtained from the Kaggle environment used for the competition.

3. METHODOLOGY

All actions taken are listed below in order.

***** Due to the problems with Python, the R Studio environment was used first. SVM classifier was used to fit the training data. However, the computer locked up. After about five iterations, it was decided that there was no proper machine learning library in R and started using Python with Google Colab. As a first attempt, it was tried to fit the data with an SVM classifier without any pre-processing of the data. Colab crashed after two hours due to RAM utilization.

***** An SVM implementation using Stochastic Gradient Descent (SGD) was observed in the Scikit-Learn documentation and then tried. It did the fit in two minutes. But the accuracy was very low.

* To improve Accuracy, Pipeline was used with Polynomial features and the data was fit again. Accuracy was even worse than before.

* After observing that the use of Standard Scaler is crucial for SVM, SVM fit using Standard Scaler and Polynomial feature was submitted. This was our first successful submission. Kaggle score: 0.37492

* After filtering the outlier in the data, it was fit with a KNN classifier. The Kaggle Score of this was the same as the sample submission.csv.

* After filtering the outlier with the method used for the first successful submission, the data was fit. Accuracy dropped even more. Moreover, another identical one was made with the same classifier (same training data and parameters). Accuracy dropped even more.

* The same classifier was fit several more times with minor modifications (outlier filtering, higher iteration, no test split, zero filter, lower tolerance). Accuracy dropped further with each attempt.

* Then, NDVI and NDWI indices were calculated. With the existing bands in the data and the newly created NDVI and NDWI columns, a Random Forest classifier was used to fit the data at a depth of ten iterations. Pipeline was not used here and manual normalization was performed. Kaggle score 0.31709

* The data was fit several more times with the feature set used in the previous step. Accuracy was low in all cases.

* The Random Forest classifier was applied once more by increasing the "depth" parameter (1000). Kaggle score 0.37079

* Entropy Filter was applied using the Scikit-Learn image library. This took about thirty minutes. However, Random Forest train times became extremely long (> 30 minutes). Accuracy did not increase. Colab container started crashing due to lack of RAM.

In the meantime, the Pickle Module was used to save the data generated by all these problems. Once the data was saved with NDVI and NDWI columns and Entropy Filter, it could be fit.

* Using all the data, the data was fit again with SVM. Accuracy was worse than all the previous ones.

* A separate classifier was applied for the water data, rest SVM classifier. When trying to classify water, RAM maxed out and crashed. After setting polynomial=2, no RAM problem occurred. Kaggle Score=0.40495

* All plants were classified as a single group with SVM classifier. Kaggle score 0.35359

* SVM water classifier, SVM building classifier and Random Forest Plant Classifier were used to classify the data respectively. Kaggle score 0.35

When the classifier with the highest score was tried again, it did not give the same result and the score dropped.

4. FEATURE EXTRACTION

The given data set was represented by 4 bands. In all the studies, it was observed that these 4 basebands were not sufficient to make an accurate classification. Therefore, feature extraction techniques were used to better represent the data. In order to make accurate predictions and classifications from the data set, we needed other features that represented the data more accurately. At this point, **Normalized Difference Vegetation Index (NDVI),** which quantifies vegetation by measuring the difference between near-

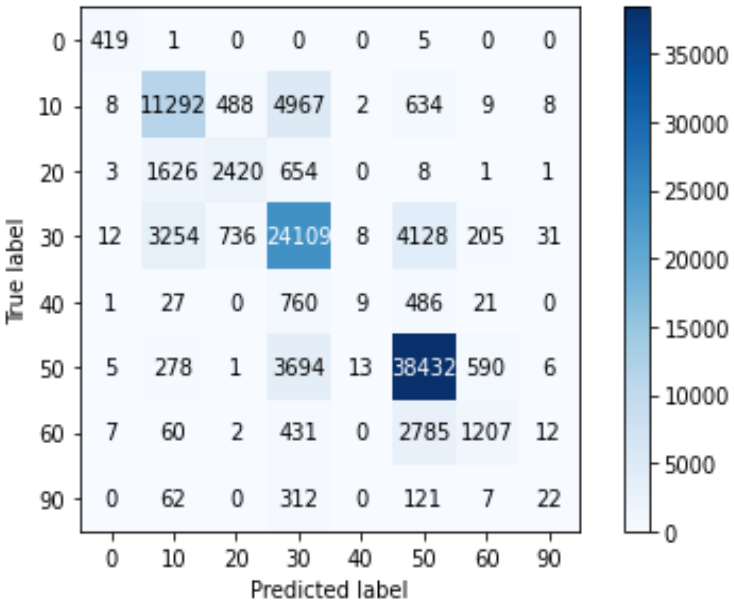
infrared and red light, and **Normalized Difference Water Index (NDWI),** which quantifies moisture content by measuring the difference between near-infrared and green light, were used.

In addition, convolution method is used for feature extraction. Interestingly and successfully, the feature selection and extraction were done by itself. For each of the four bands, a convolution was calculated separately, which was used as an attribute to classify the data. **'Conv_Blue', 'Conv_Green', 'Conv_Red', 'Conv_NIR'**

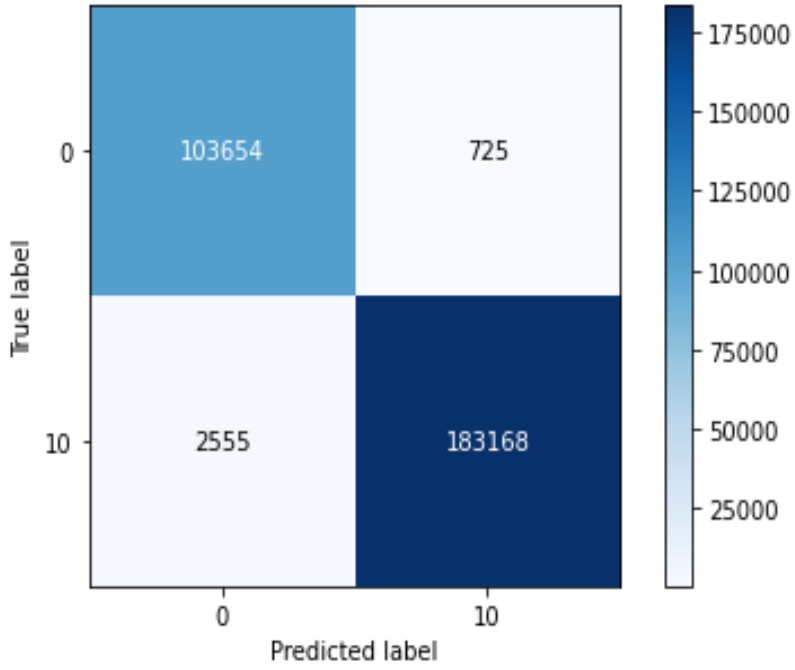
With all these feature extraction studies, a total of ten features were used for classification.

5. RESULTS

Below are the confusion matrices resulting from some studies. The first matrix is the result of the SVM predicting except for the water data.



The second matrix is the result of the SVM predicting only the water data.



Out of all the classifiers we used, Support vector machines yielded the best results with additional features like entropy, NDVI and NDWI combined with Standard Scaler with Polynomial Features of rank 3.

Due the internal one vs one structure of SVM in Scikit-Learn library, Water caused a cascading effect through all the classifiers made during fitting because of its high accuracy.

To prevent this, we implemented another one vs all classifier specifically for water and isolated it from every other class to attain a higher accuracy.

6. DISCUSSIONS AND CONCLUSIONS

In the end, the biggest problem we've faced was neither insufficient data or algorithm but rather the resources. It was frequent for algorithms to take 30+ minutes to fit

and pre-processing took at least 1 hour to finish. Crashes were frequent as the systems run out of memory trying to store the intermediate data. The resource consumption was big enough to compell us into using save/load systems to save memory and time.