Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Master in Robotics

Artificial Intelligence

Google Earth and K-Means: Exploring Urban and Natural Dynamics

Group 1

Jorge Guijarro Tolón (23075)

Josep María Barberá Civera (17048)

January 15, 2024

Table of Contents

CHAPTER 1

Introduction

In the following study, we intend to show different tools to observe, analyze and classify different images, both of aerial and urban landscapes. For this purpose, we will work with supervised and unsupervised classification methods that guarantee an optimal and efficient performance when classifying images. In particular, clustering techniques will be used, algorithms whose main function is to find groups of pixels with similar characteristics, grouping them according to these properties. All of this will be implemented using Google Colab and Jupyter Notebooks, where, in addition, a previous training of the algorithm will be required for its correct operation. Finally, the implemented code will be tested in different cases to analyze the results and its behavior

In particular, different urban environments (U.S. and European cities) will be analyzed and classified with supervised and unsupervised methods using the Python package *Geemap*. On the other hand, the effect of drought in our country will be analyzed, observing the variation of water present in a reservoir over the years. For this task, the package *Scikit-learn* will be used.

To achieve the aim of this project, it is necessary to break it down into objectives, each of which will further be subdivided into tasks or small work packages. The primary objectives can be summarized as follows:

- **Theoretical basis** of supervised classification and unsupervised classification.
- Concise review of **K-means clustering in environmental dynamics**.
- **Study scenarios**: Urban Environments and Sau and Susqueda Reservoirs.
- **Implemention** of Algorithms.
- Main **results**.
- Compile **results and draw conclusions**.
- **Roles Distribution**.
- **References** consulted.

All the information, as the codes implemented and their respective readmes, are available in a **Github**, whose link can be found here: https://github.com/jbarciv/ClusterDynamics.

CHAPTER 2

Theoretical basis of supervised and unsupervised classification

In the statistical field, classification consists of dividing a series of data into several sets that share a common property. Machine Learning studies the way to make predictions using a previous knowledge base. One of its objectives is the automatic classification of objects, and there are currently three types of techniques to make this possible [1]:

- **Supervised classification**: The data are associated with labels known in advance. The established training model indicates whether the data are correctly classified or not according to these labels.
- **Unsupervised classification**: The data are not associated with any label. Their internal structure (patterns and similarities between objects) is usually used for classification.
- **Semi-supervised classification**: Some data have labels, but some do not.

2.1 Supervised Learning

Supervised learning is the use of labeled data to train algorithms that accurately classify data or predict outcomes. As data is fed into the model, its weights are adjusted until the system is correctly matched. Supervised learning can be separated into **two types of data mining problems**:

- **Classification** uses an algorithm to accurately assign test data to specific categories. The most common classification algorithms are linear classifiers, decision trees, etc.
- **Regression** is used to understand the relationship between dependent and independent variables. Linear regression method, logistic regression or polynomial regression are popular regression algorithms.

Supervised classification relies on the availability of training areas to teach models to generate the desired output. The class to which these areas belong is known *a priori*, and they are useful for the generation of characteristic marks in each zone. Once this is done, obtaining the features of each of the classes will consist of a simple query of the images.

2.2 Unsupervised Learning

On the other hand, unsupervised classification methods are able to work with any image without the need for it to be labeled, and can also find apparently hidden relationships between different sets that would not be visible if they were labeled. Another great advantage is the computational optimization that this kind of algorithm presents, compared to those that use supervised classification.

2.2.1 Clustering

One of the main techniques used in unsupervised classification is clustering, whose main function is to find related groups of instances (clusters). There are a wide variety of algorithms based on this technique, for example, the one used for this work: the K-Mean algorithm.

This algorithm, designed by Hugo Steinhaus and Stuart Lloyd in 1957, attempts to iteratively assign a cluster to each point of the dataset, so that the sum of the distance between each point and the centroid (point representing the geometric center of the cluster) of its group is minimized. This operation will be repeated with all the points of the cluster until the centroids are not modified (see Figure 1).

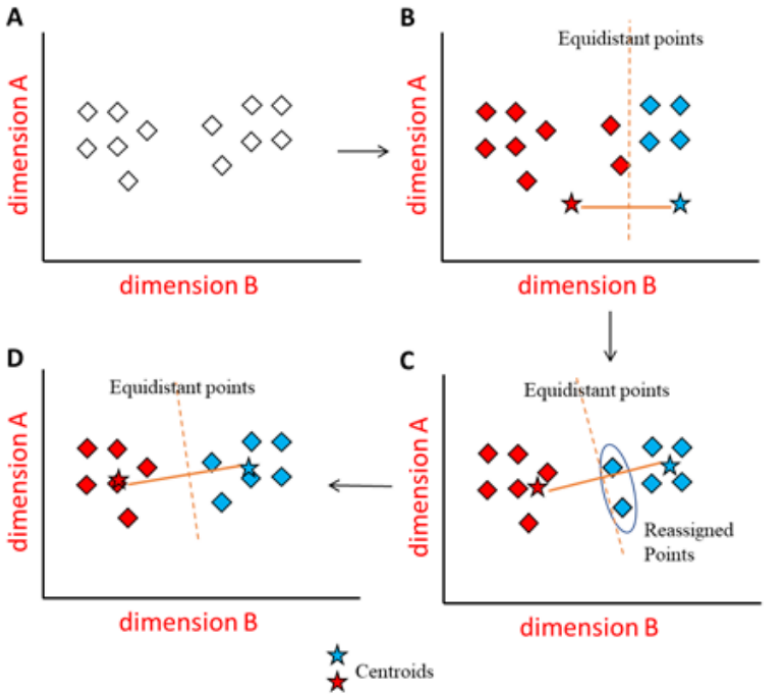

Figure 1: *Modification of the centroids until the total distance is minimized. Figure extracted from data [2].*

The steps followed to find the clusters based on datas [3] and [4] are shown below:

1. **Initialization**: once the number of clusters to be used, K, is decided, K centroids are established in the data space. This first position is usually done in a random way.

2. **Assignment of objects to centroids**: Each data is assigned to its nearest centroid. For this purpose, the quadratic Euclidean distance is usually used.

3. **Update of centroids**: The position of the centroids is updated taking as new geometric center the mean position of the data belonging to each group.

Steps 2 and 3 will be repeated until the position of the centroids is not modified, as seen above.

Mathematically, the problem of minimizing, within each group, the sum of the distances of the objects to their centroid can be expressed as follows:

$$\min E(\mu_i) = \min \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{1}$$

where μ_i represents the centroid of each group, k represents the number of groups and S represents the data set.

In addition, for each iteration, the necessary condition of *extrema* is imposed on the function $\mathrm{E}(\mu_i)$:

$$\frac{\partial E}{\partial \mu_i} = 0 \rightarrow \mu_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j \tag{2}$$

taking the mean of the elements of each group as the new geometric center.

The k-means method stands out for its simplicity, speed and efficiency in data clustering. However, it also has its drawbacks such as the requirement to predefine the number of clusters (k) and to be sensitive to centroid initialization, which may lead to local minima instead of the global one. In practice, to choose the optimal value of k it is necessary to compare results of runs with different k and select the best one according to specific criteria [5]. A pseudocode of this method is shown in Figure 2:

Algorithm 1 k-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

Figure 2: *K-means algorithm extracted from [6]*

CHAPTER 3

K-means Clustering in Environmental Dynamics

This section briefly summarizes some of the latest research on urban and environmental changes using artificial intelligence techniques, especially supervised and unsupervised classification with algorithms such as k-means clustering.

From rock classification in the area of lithology [7] to zoning environmental risk zones in industrial settings for risk management [8] passing clustering performance metrics in climate models [9], the use of the K-means algorithm is key to classifying entities given certain similarities or metrics. Even studies such as socio-economic-environmental studies can find applications in this algorithm as in the case of [10] where it allows the correlation between the estimation of ecological restoration, economic development and investment in maritime areas.

More specifically, for this work we are interested in geographic and environmental studies using satellite images as is the case of [11] in which a study is carried out on simulations of changes in the soil of large areas and over multiple time steps, in particular, in this study the clustering technique is used as a preliminary step to separate the input information to their simulations. In addition to rural areas, urban areas can also be studied as presented in [12] where they perform measurements and characterization of different urban morphologies. Although the K-means algorithm has been applied to urban flood risk estimation with great success, as seen in the publication by Xu et al. [13] and illustrated in Figure 3, determining the optimal number of clusters remains a major challenge. The authors employ the function *silhouette* [14] to address this challenge in their use of the k-means clustering method.

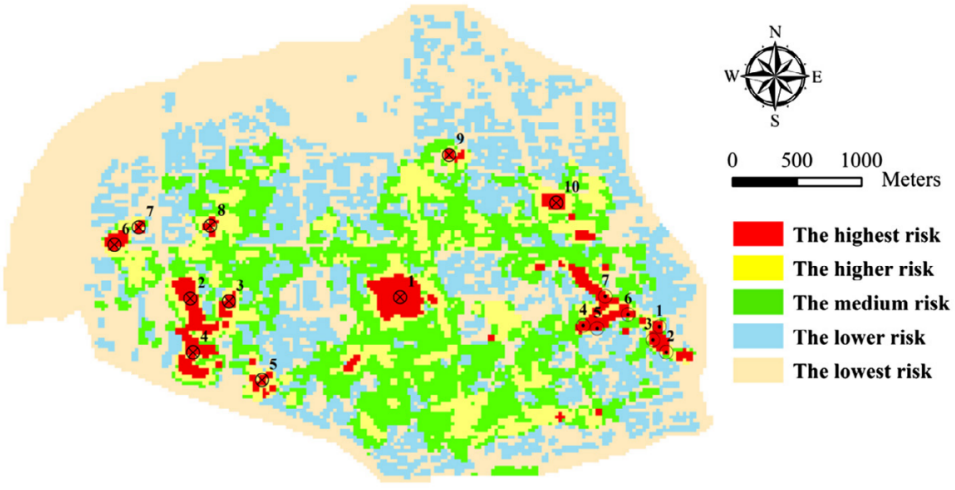

Figure 3: *Flood risk map using the enhanced entropy-cluster algorithm from [13]. The integration of k-means clustering significantly enhances the classification accuracy in this particular research.*

Finally, and as a clear example of methodologies based on these classification techniques together with the use of Google Earth Engine (GEE), it is worth mentioning [15] where the authors create a fully automated method for mapping the

extent of wetland flooding at the watershed scale using GEE, by employing the k-means algorithm, they use an unsupervised approach coupled with a random seeding technique, which significantly increases both speed and accuracy which is known as k-means++ [16].

The upcoming Table 1 serves as a summary of the consulted papers and the primary methodologies they employed in their research.

Authors	**Research topic**
Jing et al. (2021) [7]	Energy method of geophysical logging lithology based on K-means dynamic clustering analysis
Shi & Zeng (2014) [8]	Application of k-means clustering to environmental risk zoning of the chemical industrial area
Yokoi et al. (2011) [9]	Application of cluster analysis to climate model performance metrics
Dai et al. (2023) [10]	Research on ecological restoration assessment and eco-economic development of sea area by introducing the K-means clustering algorithm
Omrani et al. (2019) [11]	The land transformation model-cluster framework: Applying k-means and the Spark computing environment for large scale land change analytics
Goerlich Gisbert et al. (2017) [12]	Clustering cities through urban metrics analysis
Wu et al. (2019) [15]	Integrating LiDAR data and multi-temporal aerial imagery to map wetland inundation dynamics using Google Earth Engine

Table 1: *Summary of research on urban and environmental changes using k-means algorithm.*

CHAPTER 4

Study Scenarios

4.1 Urban Environments

Firstly, different cities have been chosen to compare their respective urban development, as well as the environment and relief that surround them. Two North American cities and three European cities were chosen:

4.1.1 US cities

New York City (Figure 4) has been selected as the "base" city, because this city has a high population density, in addition to having a great diversity of "cold" ecosystems in its periphery (sea, forest and prairie). On the other hand, the city of Albuquerque (Figure 5) has also been selected,because it is a city with a lower population density (about 15 times less), with characteristically dry ecosystems (mountain and desert).

Figure 4: *New York City*

Figure 5: *Albuquerque*

4.1.2 European Cities

In terms of European cities, the dense city of Paris (Figure 6) has been analyzed first, together with the course of its characteristic Seine River. This city also stands out for its high urban development. Traveling to our country, it has also been decided to analyze the city of Madrid, (Figure 7) specifically, the downtown area and its periphery. Finally, it has been proposed to analyze the city of Toledo, (Figure 8) a small medieval city whose characteristics compared to the previous ones is its low urban density and its considerably dry terrain.

Figure 6: *Paris*

Figure 7: *Madrid*

Figure 8: *Toledo*

4.2 Sau and Susqueda Reservoirs

In addition to the study of urban areas, geographical features of different nature have been chosen. Initially, the idea was to capture the progressive reduction of the Aneto glacier and the change in the beaches of Benidorm. Due to the low resolution of the images freely and publicly available, a larger study area had to be chosen. Based on the following web publication [17] (refer to Figure 9), which explains the great drought affecting Spain for some years and exemplified by the emptying of the Sau and Susqueda reservoirs, it has been decided to study this phenomenon in depth instead of the other two small ideas above (see Figure 10).

Figure 9: *Sau and Susqueda reservoirs in 2020 and 2023 (top and down respectively. Images sourced from [17].*

The drought is particularly severe in Catalonia, with Sau and Susqueda (reservoirs on the Ter river) being among the most affected in the entire autonomous community. This is aggravated because they are the reservoirs that supply Barcelona and the surrounding area.

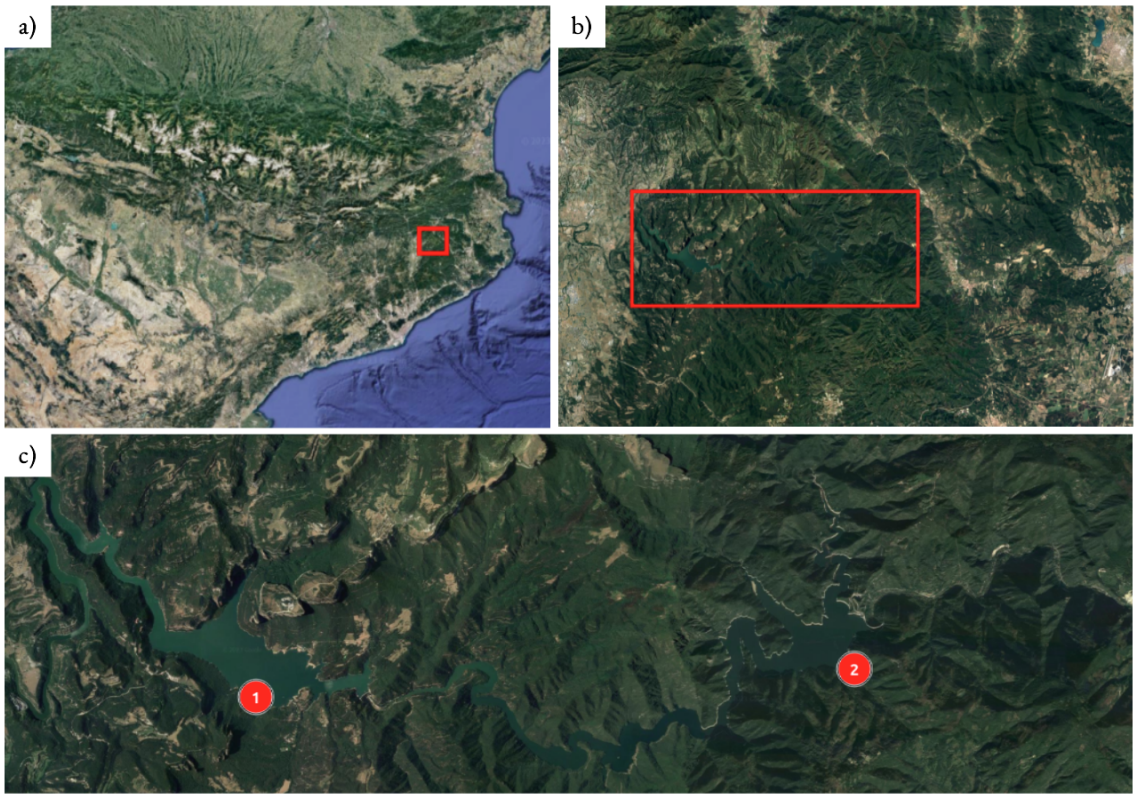

Figure 10: *Sau and Susqueda reservoirs from different points of view. In a) its location in Catalonia (northern Spain) can be seen. In b) a closer view that locates the reservoirs in the* Natural Area of Guilleries-Savassona*. Finally in c) both reservoirs can be clearly distinguished, being (1) Sau reservoir and (2) Susqueda reservoir. All the images have been taken thanks to Google Earth [18].*

CHAPTER 5

Implementation

In this project, two different libraries have been used to apply the supervised and unsupervised classification methods. These libraries and their corresponding implementations are shown and explained below.

5.1 Geemap

Geemap is a Python package that enables geospatial visualization and analysis using Google Earth Engine (GEE). This tool allows computational requests to GEE servers, allowing users to analyze and visualize GEE datasets interactively within a Jupyter-based environment [19]. This package will be used to work with the urban environments analyzed in the previous chapter, being the working images shown below:

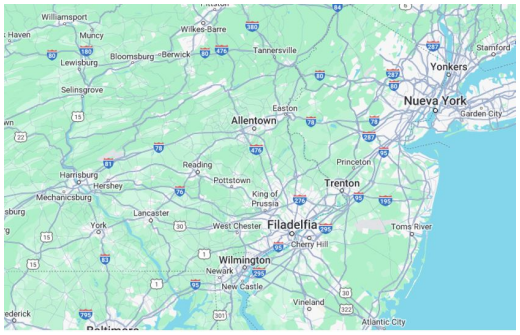

Figure 11: *New York and surroundings*

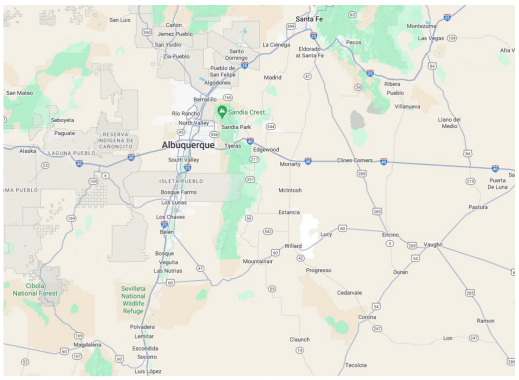

Figure 12: *Albuquerque and surroundings*

Figure 13: *Paris*

Figure 14: *Madrid*

Figure 15: *Toledo*

In the Figures 11-12, corresponding to the American cities, the focus has been on the city itself and its periphery, while in the European cities, Figures 13-15, the focus has been only on the city itself. These choices have been taken, using the *Geemap* library, which allows to create a map and select various points on it. In this case, the geographical coordinates of these cities were entered and a square was drawn in the areas of interest. For the correct selection of the images to be analyzed, the Dataset *LANDSAT* was used, which gathers information from several US satellites, to obtain the desired images [20]. Once the images were chosen from among all the resulting images, they were filtered by date and avoiding finding images with clouds that would hinder the work of the classifier. Finally, the corresponding bands, (B3, B4 and B5, corresponging to RGB), are selected to drawn the image viewing.

5.1.1 Supervised Classification

Working with supervised methods, the classification was carried out thanks to the NLCD (National Land Cover Database) Dataset, provided by the US Geological Survey [21]. In it, various regions of that country (ocean, mountainous areas, housing areas, etc.) have been classified with different labels.

For the realization of this method, we will use an approach similar to the one seen in [22]. Getting started, the training data are created through the *sample* function applied to the data coming from the NLCD dataset. For this, the application scale, the number of pixels used and the consideration (or not) of the geometry of the environment are considered. These data have been reorganized (starting the numbering of the classes at 0), in order to work in a more comfortable way. Once this is done, the information of interest will be obtained by superimposing the points of the selected image with the training data, using the previously defined bands. Then, the two final sets are defined: the training and validation sets, so that, after the execution of the algorithm, the accuracy with which the algorithm classifies the different pixels can be specified. Then, a CART (Classification And Regression Tree) classifier is used to train the model with these samples. The reason for its choice was purely experimental, as this classification model gave more realistic results than using an RF (Random Forest) Classifier. Next, the trained classifier is applied to the input image, using the same bands used for training. Finally, the values and class palette of the original NLCD layer are obtained and this information is used to visualize the classification on the map and establish a legend. In addition, to classify the percentage of each category, the total pixels of the image of interest have been obtained, as well as the pixels of each class separately. With a simple division, it is possible to determine the percentage mentioned above.

5.1.2 Unsupervised Classification

Firstly, the training data is created through the *sample* function, with the difference that in this case, it will be applied to our image. Once the training is completed, we proceed to select the number of clusters to be applied to our image. This point is fundamental for obtaining good results, since an excessively small number of clusters will not allow the complete information to be viewed, and an excessive number of clusters would imply a higher computational cost, as well as a resulting image that

is too overloaded. For the selection of the ideal number of clusters, the "elbow method" will be used, an heuristic technique based on the realization of an inertia-cluster graph, so that the number of clusters for which the variation of the inertia begins to stabilize will be selected, forming in the graph a kind of elbow (hence the name of the method) [2]. After doing so, using the function *cluster*, the clustering algorithm is applied as such, segmenting by groups the different pixels of the trained image according to their characteristics. Once this is done, the legend is configured according to the number of clusters and the program is completed. In this case, we have followed an approach based on [23].

5.2 Scikit-learn

Within unsupervised learning there are a multitude of possible algorithms. As already mentioned, the k-means clustering algorithm is used in this study. This is due to its great versatility in image segmentation. In addition to libraries such as *Geemap*, there are others such as *Scikit-learn* [24] that include a large number of implementations of artificial intelligence algorithms, specifically k-means.

First, in order to analyze environmental dynamics, it is necessary to have sufficient and quality data. This has been one of the main problems, since there is no single source and most of them are very limited and, in practice, inadequate for the study to be carried out. Finally, Sentinel-Hub [25], a terrestrial data explorer that allows the consultation and download of historical data on the same study area, has been used. Specifically, the following data files (with .gif extension) have been obtained (see Figure 16). The red color that can be seen in them is due to the use of the "false color" consisting of using not the usual RGB bands (which correspond to bands B4, B3 and B2) but bands B8, B4 and B3, being B8 the near infrared (NIR).

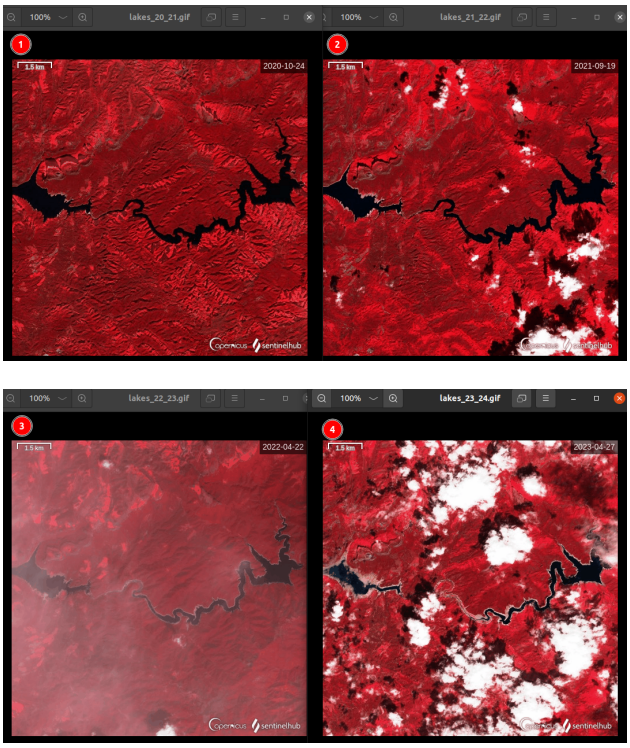

Figure 16: .GIF *files obtained thanks to Sentinel-Hub [25]. They correspond to about 200 images of the Sau and Susqueda reservoirs for the period between January 2020 and January 2024 with weekly images.*

Once the data sets have been obtained, a preprocessing has been carried out. First of all, each of the *.gif* files is taken and each of its frames is separated in order to obtain the images to be analyzed. In addition, the images have been cropped from their original size to the strictly necessary one (as can be seen in Figure 17. The reason is because the area of interest is narrower than the initial image and allows us to avoid false positives in the detection of water, such as cloud shadows or hillsides. In addition, to automate the image labeling process and given that the time stamp appears in the upper right corner of the images, a free access text recognizer (Tesseract OCR [26]) has been used to automatically obtain the name of each frame consisting of the date.

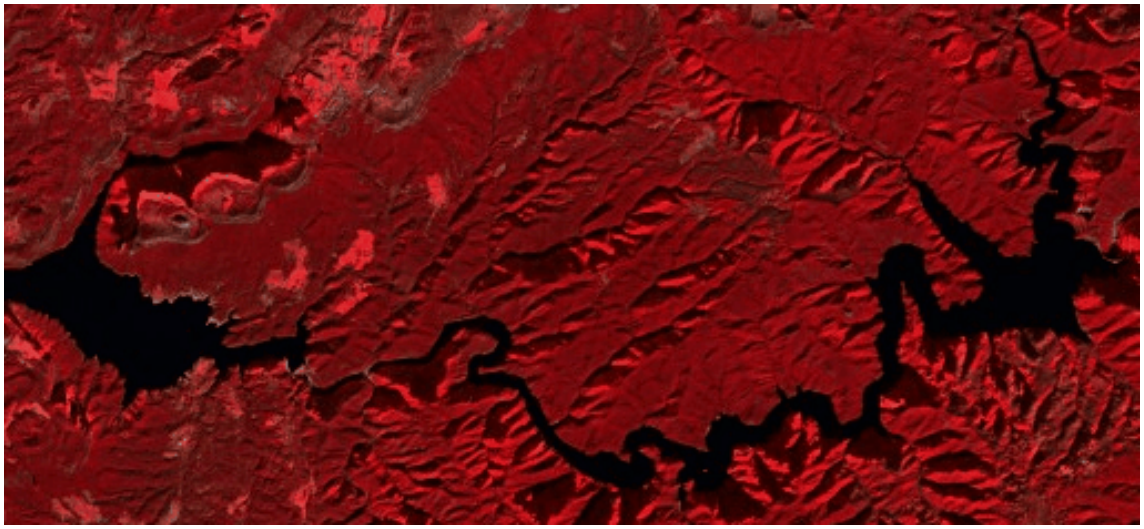

Figure 17: *Cropped figure for the k-means analysis. Apart from cropping, figures should be also converted to gray scale.*

After processing the *gifs* and obtaining the labeled photos, a first automatic filtering has been performed to discard those images that have an excessive cloud cover that hides the desired information. This filtering was performed using the clustering algorithm and analyzing the "white" content in the image (which corresponds to the cloud level). Subsequent manual filtering was necessary to ensure that the classification, but also the filtering, was correct. The 200 images for the reservoir dataset were finally reduced to 95.

Classification is performed using a clustering algorithm. To mitigate outliers in black pixel counts during clustering, four reference images have been selected for each year. These reference images are known to provide very accurate segmentation results for the entire aquatic area (see Figure 18). Next, for the remaining images, segmentations are calculated in the interval from $k = 3$ to $k = 10$, where k represents the number of clusters in the k-means algorithm. The segmentation area closest to the reference area for the specific year is considered the correct one. Despite possible variations within the same year, attributable to the low definition of the available images, it has been found that such variations do not significantly affect the segmentation.

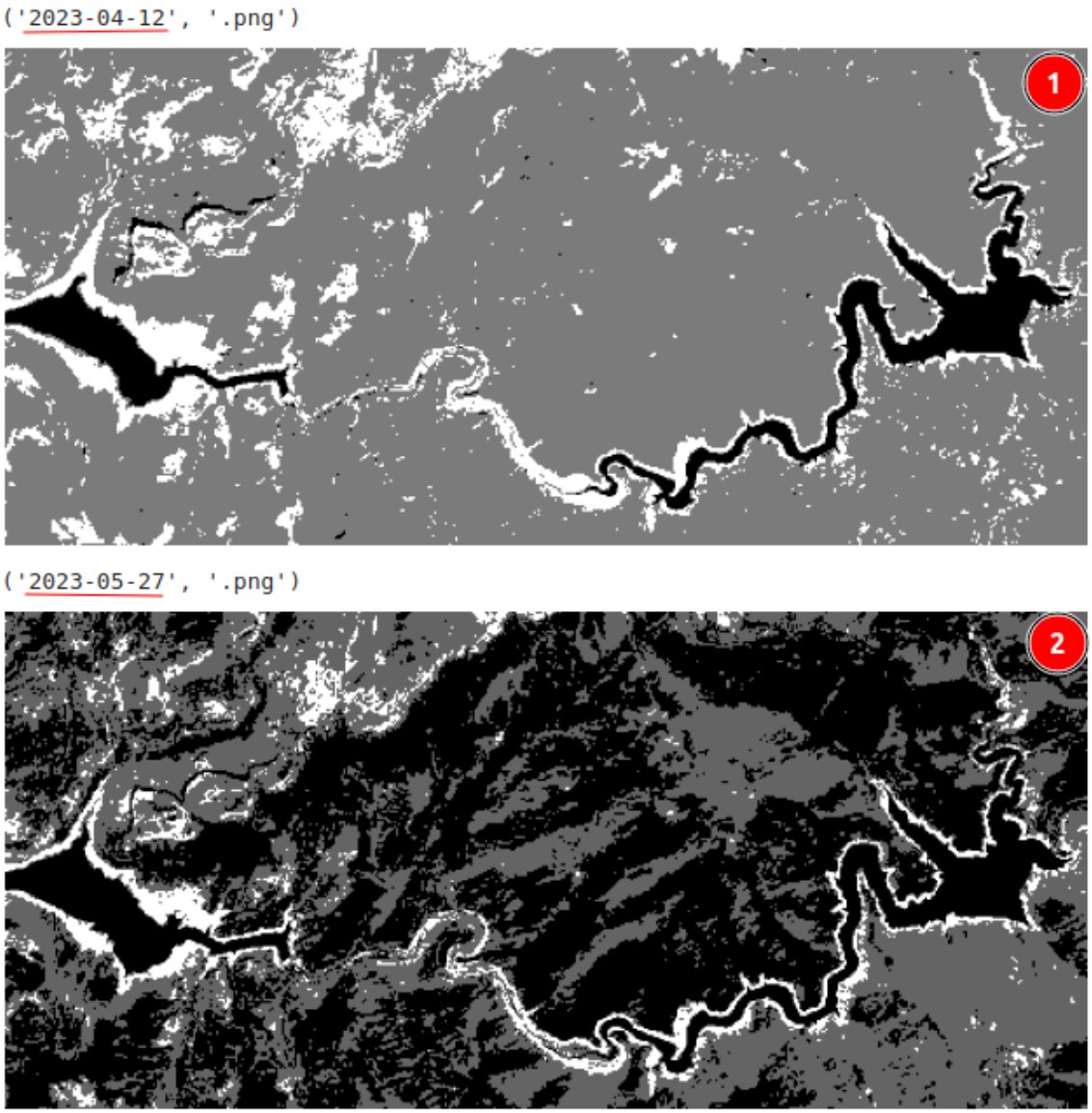

Figure 18: *k-mean clustering applied for two different images corresponding to the same year. In these cases the cluster number is 3 (black, gray and white). Figure (1) is properly segmented while (2) is not. Similar examples as figure (1) have been chosen to compute the reference area for the further analysis.*

On the other hand, given that weekly values are available, it has been decided to perform the monthly mean and the calculation of the sample variance. The annual mean is also calculated. The results are explained in more detail in the following section.

CHAPTER 6

Results

6.1 Urban Area Analysis

The main results for both supervised and unsupervised methods are presented below.

6.1.1 Supervised Classification

- **New York City**: Two large urban masses (in red) corresponding to the city of New York and the city of Philadelphia can be easily seen in the Figure 19. They are bordered by the sea and the course of several rivers (in blue). On the coast, it can also see a certain urban area corresponding to the maritime towns and cities of the American east coast. In addition, advancing towards the west, it can be seen how the urban masses are disappearing, observing more natural and wooded landscapes (in green color). The accuracy obtained for this image was 100% for the previously trained data, and 42% for the validation data. This indicates a poor generalization of the model to be extrapolated to other possible regions. Finally, it has been determined, for this image, the areas of Table 2.

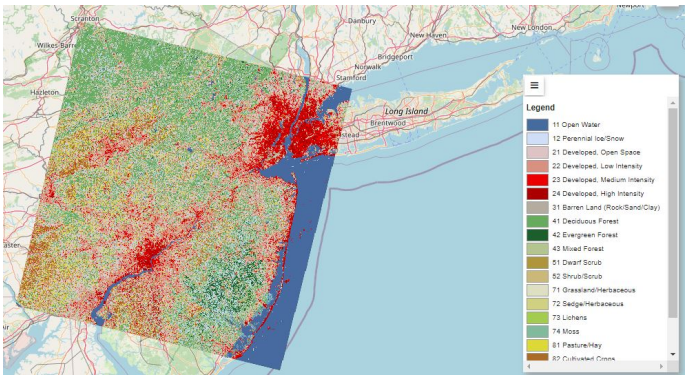

Figure 19: *Results of Supervised Classification*

Area	Percentage
Urban Area	30.12%
Aquatic Area	12.46%
Dry Area	15.96%
Grassland Area	0.57%
Forest Area	29.68%
Mountain Area	0.3%
Another Area	10.88%

Table 2: *Areas NY*

Entering New York City, it can be appreciated in Figure 20,the outline of the main avenues, as well as some of the most famous parks in the city (Central Park or Highland Park) and the Hudson River.

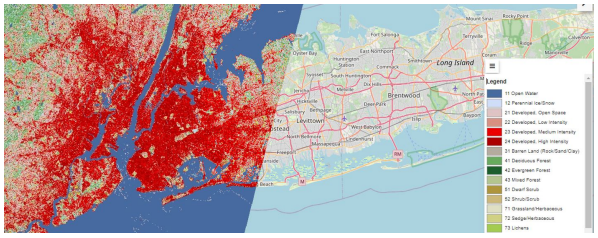

Figure 20: *Results of Supervised Classification in NY city*

- **Albuquerque**: This time, in the resulting image, Figure 21, the brown tones typical of the American plains stand out. One of the exceptions is the city of Albuquerque, surrounded by several forested mountains to the east of the city. As in the case of New York, an accuracy for the training data is obtained of 100%, significantly improved the validation data, reaching an accuracy of 71%. Each area section is determined in Table 3.

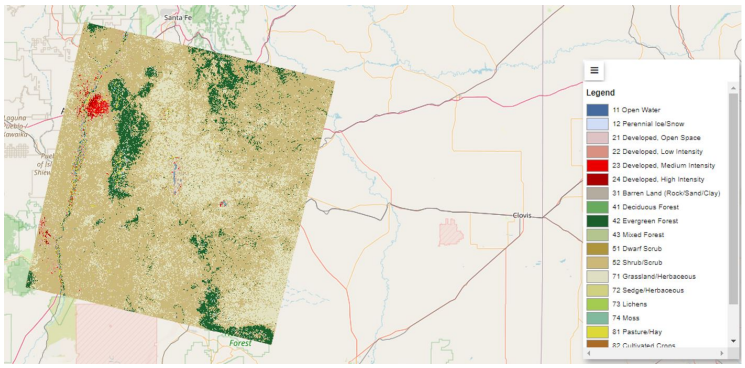

Figure 21: *Results of Supervised Classification in ALB*

Area	Percentage
Urban Area	2.4%
Aquatic Area	0.04%
Dry Area	62.51%
Grassland Area	24.92%
Forest Area	9.46%
Mountain Area	0.04%
Another Area	0.61%

Table 3: *Areas ALB*

For European cities, it has not been possible to implement this method due to the lack of a good DataSet that correctly identifies the labels for the cities and environments of the old continent. An approximation could be made using the classifier implemented for New York or Albuquerque, although in this way accurate results are not being obtained. For the city of Paris, for example, these two classifiers were implemented manually[1], obtaining the following results:

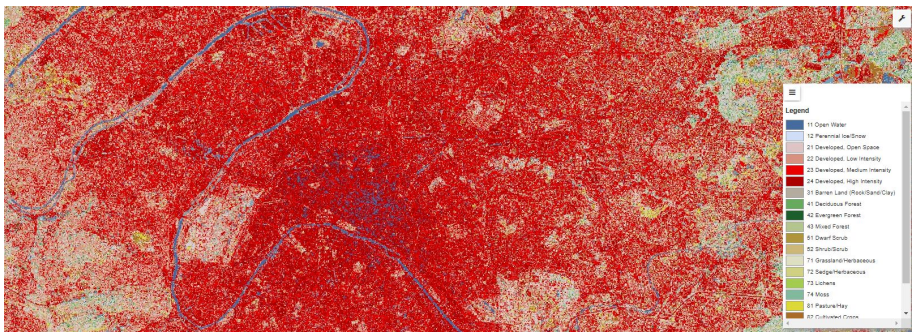

Figure 22: *Paris with NY training*

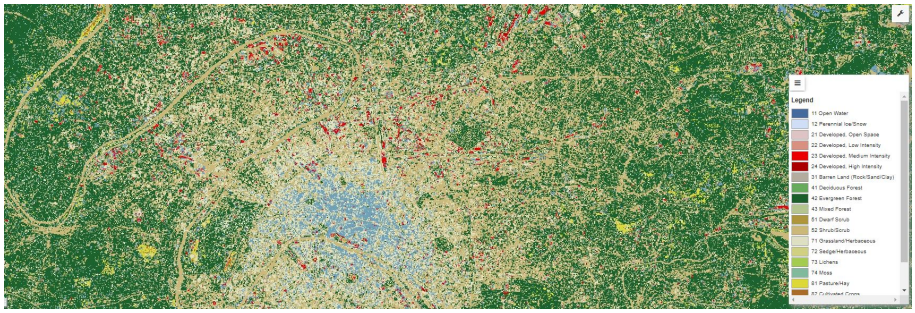

Figure 23: *Paris with ALB training*

[1] i.e., significant changes had to be made to the code structure and it has not been presented in the final documentation.

Area	Percentage
Urban Area	38.30%
Aquatic Area	1.15%
Dry Area	41.92%
Grassland Area	0.02%
Forest Area	16.43%
Mountain Area	0.08%
Another Area	2.07%

Table 4: *Areas Figure 22*

Area	Percentage
Urban Area	0.51%
Aquatic Area	1.11%
Dry Area	25%
Grassland Area	0.56%
Forest Area	68.15%
Mountain Area	0%
Another Area	5.64%

Table 5: *Areas Figure 23*

Significant discrepancies are observed in terms of classifying the groups. Figure 22 , shows how the urban area and the river are correctly detected, since this model was trained in a densely populated area with several similar areas. On the other hand, for Figure 23 , it can be seen that the urban masses are detected as wooded or dry areas. This may be due to the fact that in this model, the most detected zones were those mentioned beforehand. The results for each area analyzed are shown in the tables 4-5.

This leads to conclude on the importance of using consistent training models that are consistent with the image to be analyzed. Therefore, the use of supervised methods for regions that are not registered in a Dataset could be ruled out.

6.1.2 Unsupervised Classification

In the next subsection, it will be analyzed the results obtained for the unsupervised methods. It should be noted that Geemap does not incorporate any tool for the quantitative validation of the data and its verification must be done visually.

First, the optimal number of clusters for the visualization of results in each of the cases will be established. As can be seen in Figure 24; for New York City, the variation of inertia starts to stabilize from 8 clusters onwards. Similarly, as seen in Figure 25, for Albuquerque this happens at 8 clusters. Unlike the supervised methods, with this technique the analysis of European cities is possible, with the ideal number of clusters for the city of Paris being 7 (slightly before a small peak), as seen in Figure 26, and 8 for the cities of Toledo and Madrid, as seen in Figure 27. The results for images with a higher and lower number than ideal will also be visualized to appreciate the significant differences between them and the importance of establishing a correct number of clusters. Once the ideal clusters for each image have been analyzed, the results will be displayed.

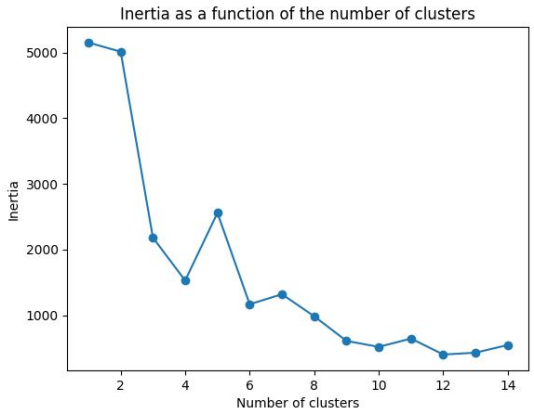

Figure 24: *Results of NY*

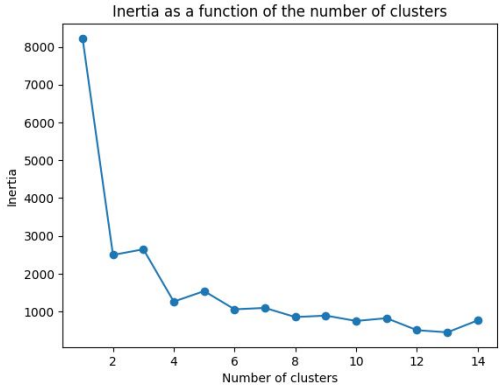

Figure 25: *Results of ALB*

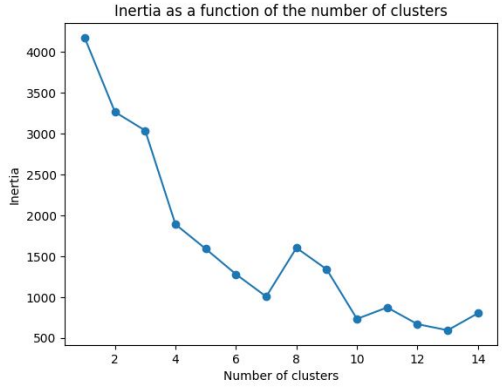

Figure 26: *Results of Paris*

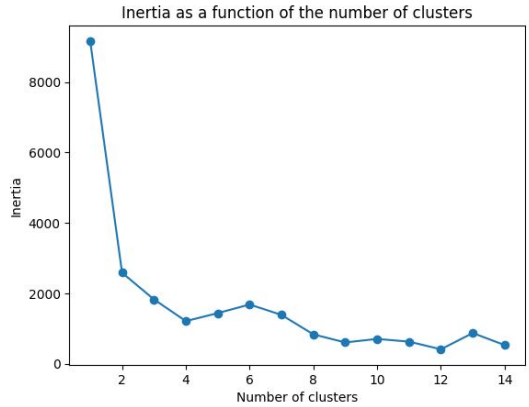

Figure 27: *Results of Madrid and Toledo*

- **New York City**: A panoramic view of the environment is shown in Figure 28. A practically similar result to that obtained with the supervised method analyzed in Figure 19 is observed. Going into more detail, we will proceed to compare the view of New York City by varying the number of clusters: it can be seen that in Figure 29 a partially correct distribution of the city is observed, unlike the results obtained in Figure 30, where it is not possible to differentiate the green areas of the city, also grouping the urban areas with the dry ones. On the other hand, in the representation of Figure 31, the algorithm differentiates river water from sea water. In addition, the layout of some of the streets and buildings is diffused. It is worth noting the use of different color palettes for each cluster value, this is because using the unsupervised method, none of the regions have color labels associated to their class and the distribution of colors is done randomly according to the pre-established legend.

- **Albuquerque**: Similarly, a panoramic view of the region, reflected in Figure 32, is presented. As in the previous case, similar patterns are observed to its analogous image for the supervised method (Figure 21). Analyzing the city itself, a correct distribution is observed in Figure 32, with a correct street layout, differentiation between buildings and different natural elements. On the other hand, in Figure 33, the algorithm fails to differentiate the mountainous terrain with the river and some elements of the city itself. In Figure 35, the layout of the streets is confused with the city buildings themselves.

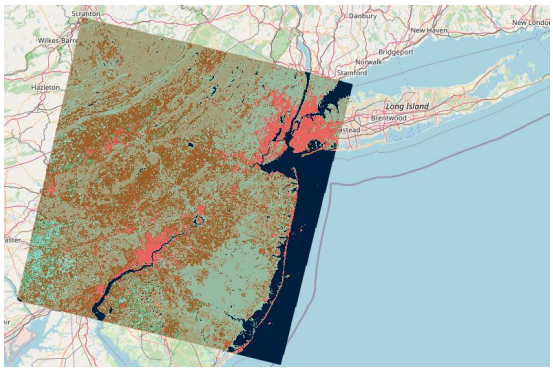

Figure 28: *Results of NY*

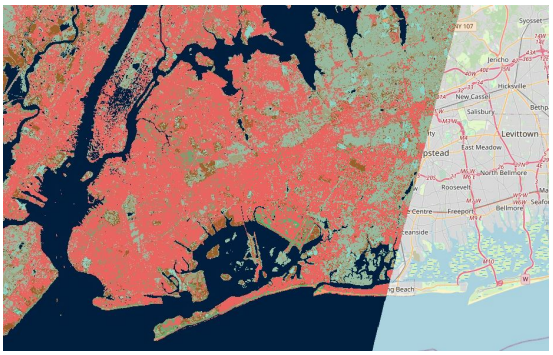

Figure 29: *NY with 8 clusters*

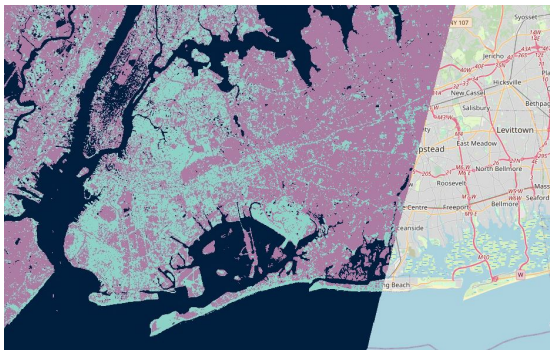

Figure 30: *NY with 3 clusters*

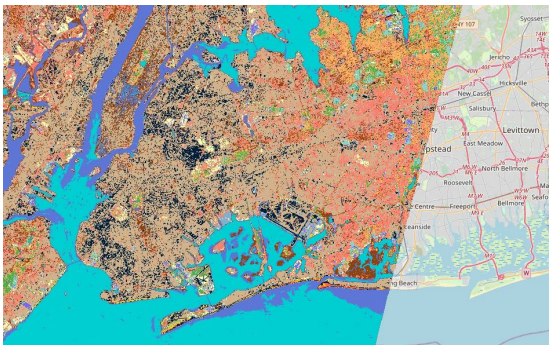

Figure 31: *NY with 14 clusters*

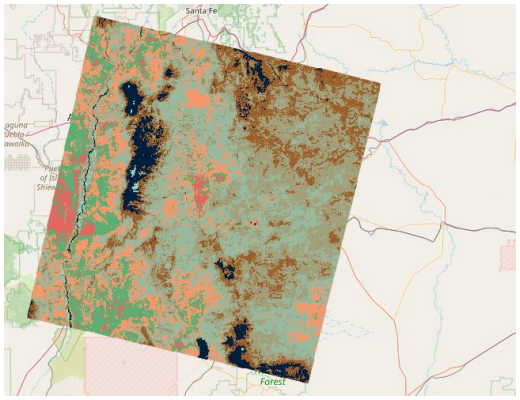

Figure 32: *Results of ALB*

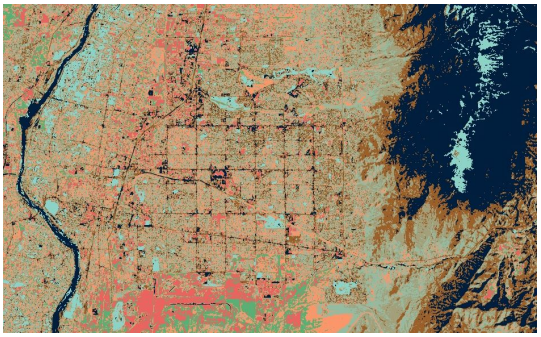

Figure 33: *ALB with 8 clusters*

Figure 34: *ALB with 3 clusters*

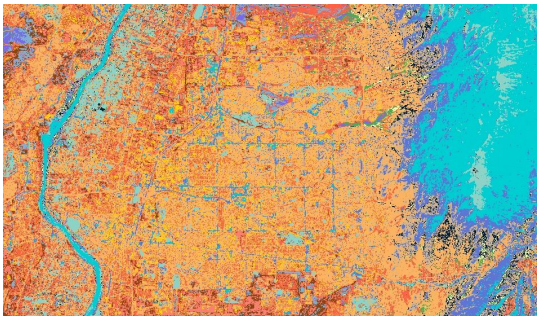

Figure 35: *ALB with 14 clusters*

- **Paris**: Analyzing Figure 36, we can see the course of the Seine River marked in blue in great detail, as well as the layout of the main parks of the city in an orange color. However, the layout of the streets could be confused with the course of the river. Moving on to using fewer clusters, Figure 37, it is observed that the algorithm fails to differentiate between the buildings, the river and the green areas, generating an image that is certainly confusing. On the other hand, using a larger number of clusters, Figure 38, results are somewhat more confusing than those presented beforehand.

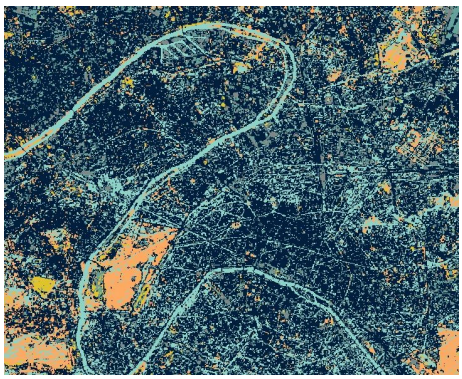

Figure 36: *Paris with 6 clusters*

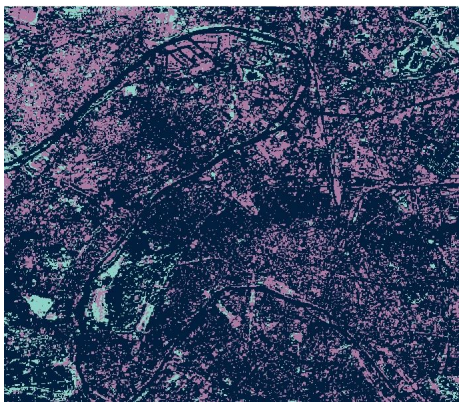

Figure 37: *Paris with 3 clusters*

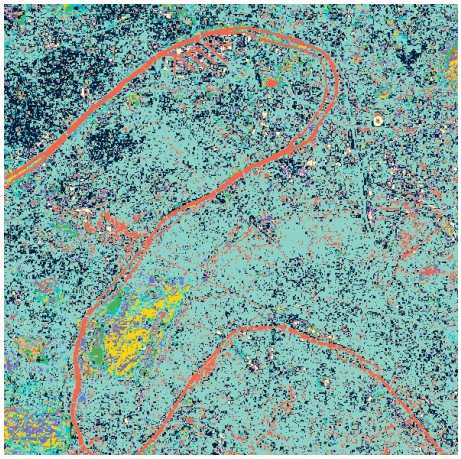

Figure 38: *Paris with 14 clusters*

- **Madrid and Toledo**: Finally, these two Spanish cities will be analyzed. Analyzing the city of Madrid, for the optimal number of clusters, Figure 39, the layout of streets, (in black) of urban areas (in green) and green areas (in red) is correctly appreciated. On the other hand, by using a smaller number of clusters, as shown in Figure 40, the differentiation between streets and buildings is not possible. Finally, if more clusters than necessary are used, Figure 41, a too overloaded result is observed when analyzing the streets. Moving on to analyze the city of Toledo, using the ideal number of clusters, Figure 42, the passage of the Tajo river bordering the city, is appreciated; however, certain urban areas are confused with some open fields. By decreasing the number of clusters, Figure 43, it is not possible to distinguish the river from the different rural plots that surround it. Finally, using a higher number of clusters than necessary, Figure 44, a confusing result is seen for some rural areas (combining colors such as purple, yellow, orange and red), although it is possible to discern the urban areas (reflected in green).

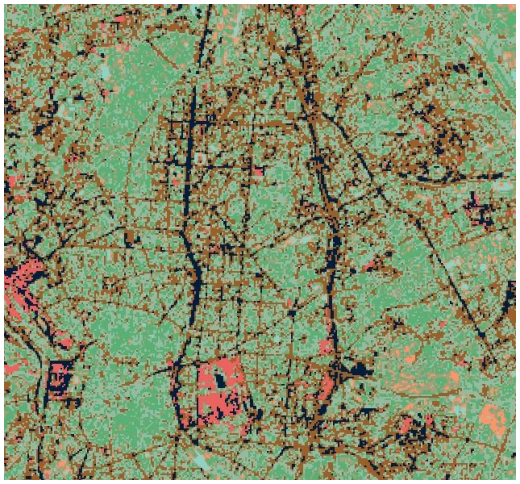

Figure 39: *Madrid with 8 clusters*

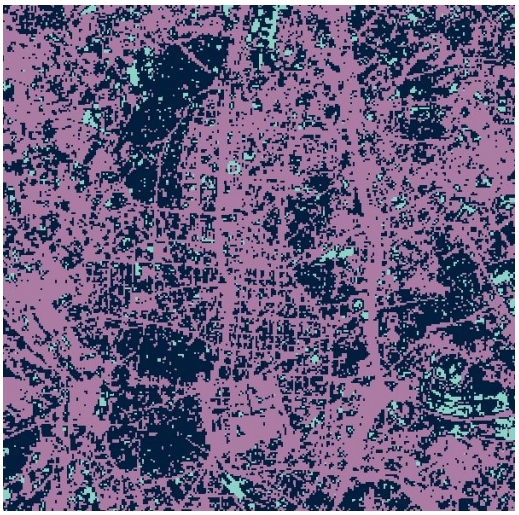

Figure 40: *Madrid with 3 clusters*

Figure 41: *Madrid with 14 clusters*

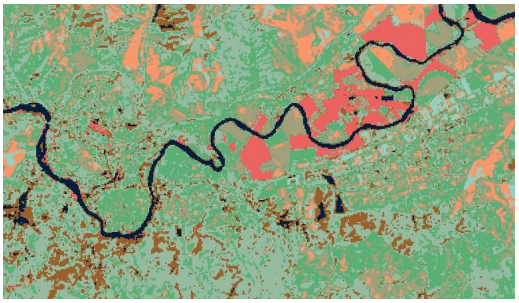

Figure 42: *Toledo with 8 clusters*

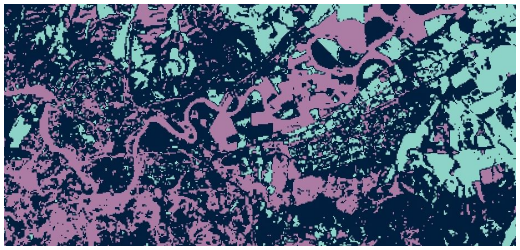

Figure 43: *Toledo with 3 clusters*

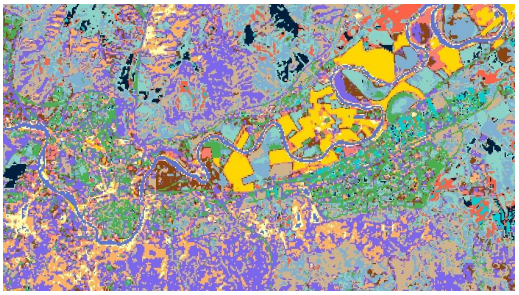

Figure 44: *Toledo with 14 clusters*

6.2 Natural Dynamics Analysis

The results obtained from the segmentation analysis of the satellite images obtained for the reservoirs *Sau* and *Susqueda* between February 2020 and November 2023 are shown below (see Figure 45) and confirm numerically what was expected based on the reduction of the reservoirs in Figure 9.

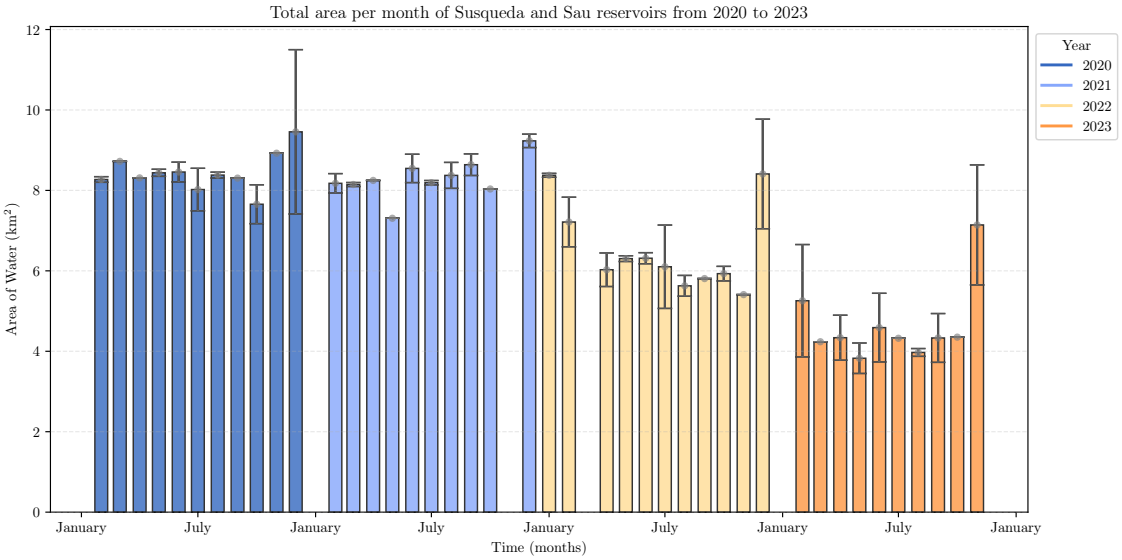

Figure 45: *Total area per month of Susqueda and Sau reservoirs from 2020 to 2023.*

Each image was processed, previously rejecting those that did not allow a good visibility of the reservoirs. Segmentation was performed using the k-means algorithm and the data were post-processed. Since weekly images were available, the filtering did not affect the results too much and in most months more than one water area value was available. In this way it has been possible to calculate the mean and variance for the measurements obtained monthly. The graph shows the mean monthly water values and their variance in gray. It can be clearly seen that there is a strong tendency for the area to decrease.

On the other hand, it is worth noting how the area has been calculated. Knowing the total number of bits that make up the figure and knowing the real area of the image, simply calculate this quotient by the number of black bits (water) detected.

$$\text{black\_pixels} \times \frac{\text{real\_area}(\text{km}^2)}{\text{pixels\_in\_figure}}$$

Specifically the *real_area* has been computed thanks to Google Earth as shown in Figure 46 with a value of approximately 84 km^2.

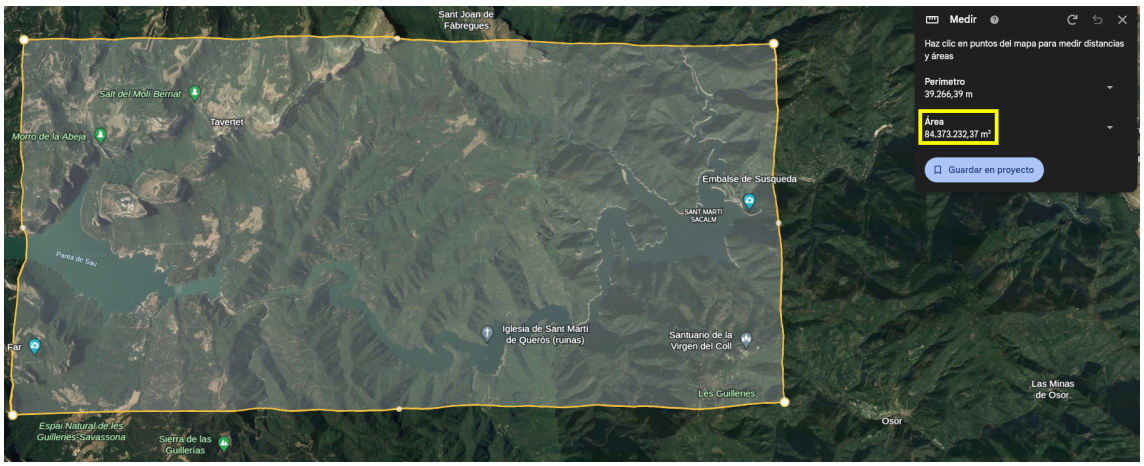

Figure 46: *Real area for the images segmented. With a value of approximately* 84 km^2. *Obtained thanks to Google Earth [18].*

To better appreciate the change in reservoir area, the annual mean has been calculated and plotted (see Figure 47) including again the associated variance. It can be concluded that the drought is water being the decrease of a 1.89% in 2021, a 23.05% in 2022 and a decrease of a 45.13%.

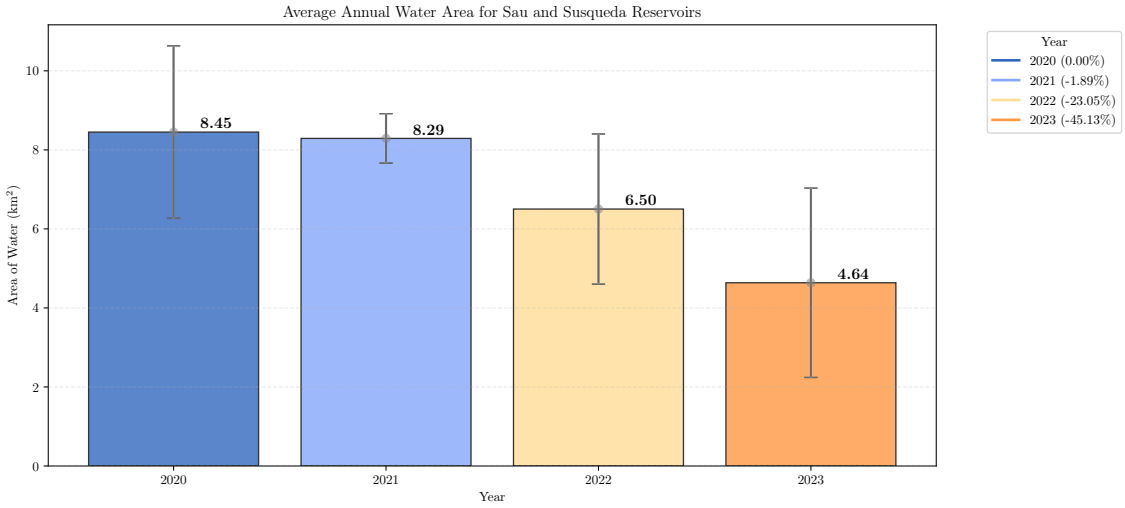

Figure 47: *Total area per year of Susqueda and Sau reservoirs.*

CHAPTER 7

Conclusions

The main conclusions of the project are summarized below:

- *Geemap* requires a **complex installation process** compared to other Python packages (This is shown in the readme uploaded to Github).
- Also, **the existing work area is very limited**, since it only allows working with rectangles emerging from the selected point or, failing that, with polygons given their vertices (being necessary to obtain the geographical coordinates).
- Poor image resolution was obtained working with *Geemap* when capturing images at close range. The mains reason of this is the low download capacity of Google Earth Engine, allowing only 40 mb of data to be downloaded.
- For **supervised methods**, it is necessary to use a well detailed and consistent DataSet of the area to be analyzed for a correct analysis and good results. For the cases analyzed, good results have been obtained, in terms of training data, although not very generalizable.
- For **unsupervised methods**, in most cases, the results are very similar to those obtained with the supervised methods using, in this case, less information; however, their validation must be done visually (the package does not incorporate any quantitative tools). In addition, it is necessary to determine the correct cluster number beforehand in order to visualize consistent results.
- The data obtained with the **Sentinel Hub** scanner have been adequate. To emphasize the great search carried out, being this source the only one that has been really useful for the study to be carried out.
- Another big problem has been the need for **filtering**. From the initial 200 images, it has been reduced to only 95 images. A possible improvement to avoid such high cloud levels is to use daily data instead of weekly data, so filtering would affect less and there would be more information to calculate monthly mean values.
- **Scikit** has turned out to be a great library that has made it possible to implement the k-means algorithm with great ease. In addition to this library, it is worth mentioning the use of **Jupyter Notebooks** for this type of studies as well as **Matplotlib** for its great versatility in making graphs and figures.
- It can be concluded that the Sau and Susqueda reservoirs are suffering from a serious drought. The use of satellite images and clustering algorithms such as k-means have proven sufficient to quantify such reduction. We believe that this type of research can be very beneficial at present and requires further study and investment.
- It is clear that access to historical and high quality satellite images is reserved for private access, which in general is a difficulty for the study of these phenomena of interest to all.

CHAPTER 8

Role Distribution

For the distribution of each of the tasks contemplated in the project proposal, the following table 6 has been created, where each member of the group is a letter from A to B, according to the table of team members (see Table 7).

PROJECT REPORT WRITING	A	B
Introduction	X	X
Theoretical basis	X	
State of the Art		X
Study Scenarios - Urban	X	
Study Scenarios - Natural		X
Implementation - Geemap	X	
Implementation - Scikit-learn		X
Results - Urban	X	
Results - Natural		X
Conclusion	X	X
Role Distribution		X

Table 6: *Project report documentation distribution.*

PARTICIPANTS	NAME	REGISTRATION NUMBER
A	Jorge Guijarro Tolón	23075
B	Josep Mª Barberá Civera	17048

Table 7: *Members and registration numbers*

References

[1] "Técnicas de clasificación para datos funcionales.Aplicación a series temporales de número de positivos en Covid19 por departamento de salud de la Comunitat Valenciana". In: (2022).

[2] "K-Means clustering made simple". In: (2020). URL: https://www.blopig.com/blog/2020/07/k-means-clustering-made-simple/.

[3] "El algoritmo k-means aplicado a clasificación y procesamiento de imágenes". In: (2018). URL: https://www.unioviedo.es/compnum/laboratorios_py/new/kmeans.html.

[4] "Introducción al clustering (I): algoritmo k-means". In: *elmundodelodatos* (2021). URL: https://elmundodelosdatos.com/introduccion-al-clustering-algoritmo-k-means/.

[5] "Aprendizaje No Supervisado: K-Means Clustering". In: *Aprendeia* (2018). URL: https://aprendeia.com/aprendizaje-no-supervisado-k-means-clustering/.

[6] "K-Means Clustering in Python: A Practical Guide". In: (). URL: https://realpython.com/k-means-clustering-python/.

[7] J. Jing, S. Ke, T. Li, and T. Wang. "Energy method of geophysical logging lithology based on K-means dynamic clustering analysis". In: *Environmental Technology and Innovation* 23 (2021), p. 101534. ISSN: 23521864. DOI: 10.1016/j.eti.2021.101534. URL: https://doi.org/10.1016/j.eti.2021.101534.

[8] W. Shi and W. Zeng. "Application of k-means clustering to environmental risk zoning of the chemical industrial area". In: *Frontiers of Environmental Science and Engineering* 8.1 (2014), pp. 117–127. ISSN: 2095221X. DOI: 10.1007/s11783-013-0581-5.

[9] S. Yokoi et al. "Application of cluster analysis to climate model performance metrics". In: *Journal of Applied Meteorology and Climatology* 50.8 (2011), pp. 1666–1675. ISSN: 15588424. DOI: 10.1175/2011JAMC2643.1.

[10] J. Dai, J. Fang, Z. Guo, and J. Hou. "Research on ecological restoration assessment and eco-economic development of sea area by introducing the K-means clustering algorithm". In: *Environmental Science and Pollution Research* (2023). ISSN: 1614-7499. DOI: 10.1007/s11356-023-30127-1. URL: https://doi.org/10.1007/s11356-023-30127-1.

[11] H. Omrani, B. Parmentier, M. Helbich, and B. Pijanowski. "The land transformation model-cluster framework: Applying k-means and the Spark computing environment for large scale land change analytics". In: *Environmental Modelling and Software* 111.October 2018 (2019), pp. 182–191. ISSN: 13648152. DOI: 10.1016/j.envsoft.2018.10.004. URL: https://doi.org/10.1016/j.envsoft.2018.10.004.

[12] F. J. Goerlich Gisbert, I. Cantarino Martí, and E. Gielen. "Clustering cities through urban metrics analysis". In: *Journal of Urban Design* 22.5 (2017), pp. 689–708. ISSN: 14699664. DOI: 10.1080/13574809.2017.1305882.

[13] H. Xu, C. Ma, J. Lian, K. Xu, and E. Chaima. "Urban flooding risk assessment based on an integrated k-means cluster algorithm and improved entropy weight method in the region of Haikou, China". In: *Journal of Hydrology* 563.June (2018), pp. 975–986. ISSN: 00221694. DOI: 10.1016/j.jhydrol.2018.06.060. URL: https://doi.org/10.1016/j.jhydrol.2018.06.060.

[14] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20.C (1987), pp. 53–65. ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7.

[15] Q. Wu et al. "Integrating LiDAR data and multi-temporal aerial imagery to map wetland inundation dynamics using Google Earth Engine". In: *Remote Sensing of Environment* 228.March (2019), pp. 1–13. ISSN: 00344257. DOI: 10.1016/j.rse.2019.04.015. URL: https://doi.org/10.1016/j.rse.2019.04.015.

[16] D. Arthur and S. Vassilvitskii. "K-means++: The advantages of careful seeding". In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* 07-09-January-2007 (2007), pp. 1027–1035.

[17] E. Cassidy. *Shrinking reservoirs in Catalonia*. Apr. 2023. URL: https://landsat.visibleearth.nasa.gov/view.php?id=151257.

[18] G. Earth. *Susqueda Reservoir, Spain*. Accessed: January 15, 2024. 2024. URL: http://earth.google.com.

[19] Q. Wu. "geemap: A Python package for interactive mapping with Google Earth Engine". In: *Journal of Open Source Software* 5.51 (2020), p. 2305. DOI: 10.21105/joss.02305. URL: https://doi.org/10.21105/joss.02305.

[20] *Landsat 8 Collection 1 Tier 1 8-Day TOA Reflectance Composite*. URL: https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C01_T1_8DAY_TOA.

[21] *NLCD: USGS National Land Cover Database [Deprecated]*. URL: https://developers.google.com/earth-engine/datasets/catalog/USGS_NLCD.

[22] Q. Wu. *32 Supervised Classification - GEEMAP*. URL: https://geemap.org/notebooks/32_supervised_classification/.

[23] Q. Wu. *31 Unsupervised Classification - GEEMAP*. URL: https://geemap.org/notebooks/31_unsupervised_classification/.

[24] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[25] sentinel-hub. *Earth Observation Browser*. 2024. URL: https://apps.sentinel-hub.com/eo-browser.

[26] Tesseract-Ocr. *Tesseract Open Source OCR Engine (Github main repository)*. Oct. 2023. URL: https://github.com/tesseract-ocr/tesseract.