Lake area extraction from satellite images

Primary Topic: GIS
Secondary Topic: TS
Course: Data Science, Edition: 2024-2A – Group: 56 – Submission Date: 2024-04-26

MARTIN POZDECH (2496712) and MIHAI-BOGDAN BÎNDILĂ (3264424)

1 MOTIVATION

Visualizing a water body's area evolution in time is important for many reasons; for example, to see if a lake is growing or shrinking, to check what areas were flooded at different times, or to compare the growth of a lake with relevant climate statistics. These all give insight to a variety of actors, such as farmers or lakeside area developers, who both want to know if the area they will be working in has been flooded or dry previously. Correlating the growth with climate statistics lets researchers see what features have a significant impact on the area's evolution.

While many researchers have addressed the problem of lake identification from satellite images, most of the related work focuses on the analysis of one individual lake at a time using machine learning models or thresholding methods [16], [1]. Other authors have developed deep learning methods based on convolutional neural networks for analyzing multiple lakes at a time in the Tibetan Plateau [8].

We want to bridge the gap between analyzing only one lake and the use of complex deep-learning models. Hence, we developed a general and threshold-based method for detecting a lake in a hassle-free manner for any area that had sufficient satellite imagery captured of it between 1984 and 2022. We focus on seasonal lakes because they exhibit a large area variability through time, which lets us analyze the water behavior. A key feature of the model is the use of the Automatic Water Extraction Index (AWEI) that, through the values in different image bands, marks the 'strength' of water presence in any pixel. Moreover, our method enables rich visualizations of the lake's evolution over time together with the time series of the lake area and correlations with the climate statistics that enable the stakeholders to gain more insights.

2 (BUSINESS/RESEARCH) QUESTIONS

In the pursuit of extracting a timeline for a water body's evolution two research questions have been established:

- How can we create a pipeline for automatic water extraction and analysis from satellite images?
- To what extent are certain natural climate phenomena correlated to the evolution of the lake?

3 SOURCE DATA

To prove the capabilities of our method, we chose four medium-sized seasonal lakes (Figure 1) from various climates. Table 1 enumerates them and the corresponding number of images after data cleaning.

Five datasets retrieved from Google Earth Engine (GEE) were used in this project. Three contained satellite data from which the images were extracted, the fourth was the ground truth, and the fifth contained the climate statistics. The satellite data used are the Landsat 5 [13], 7 [14], and 8 [15] datasets with the atmospherically

Authors' address: Martin Pozdech (2496712); Mihai-Bogdan Bîndilă (3264424).

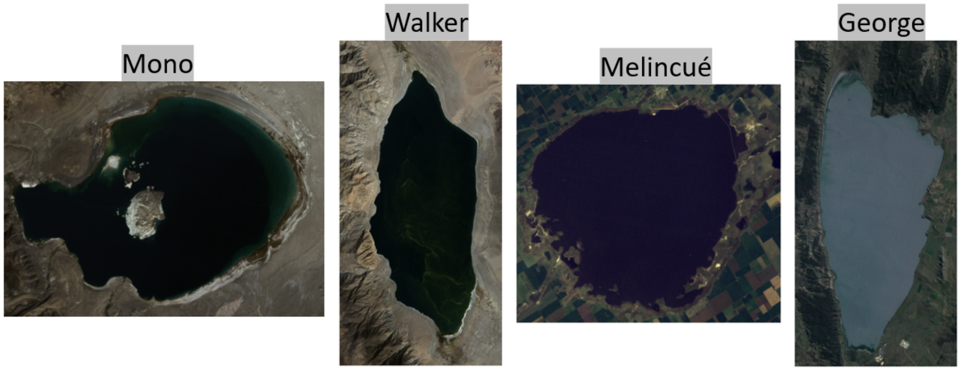

Fig. 1. Analyzed lakes

corrected surface reflectance. This correction was chosen as it normalizes the values in each band against their minima, removing atmospheric effects [6]. The Ground Truth (GT) dataset was the JRC monthly water history [4] created using an expert system [10]. It contained information if any pixel in the region was water, unknown, or not water. A benefit of this dataset was that it was generated using the Landsat data as well. The climate statistics were taken from the ERA5-Land dataset [3] that aggregated them per month. This dataset was chosen as it contained consistent observations for our analysis period of the relevant statistics, namely total runoff and surface evaporation.

Lake name	Location	Number of images
Mono	California, USA	222
Walker	Nevada, USA	247
Melincué	Santa Fe, Argentina	234
George	New South Wales, Australia	106

Table 1. Details of the analyzed lakes

While exploring the data visually in Python with Matplotlib and NumPy, we notice a few shortcomings.

Incomplete images

Sometimes the image extracted for a region had missing data in it, making its use detrimental. This was managed by checking each of the images for the number of pixels with data, and if this amount was insufficient, the image was ignored. A similar issue was observed in the GT data, but this had to be managed later in the image analysis due to limitations in the method used to assess image coverage. Figure 2 illustrates an example of missing data as a stripe pattern. Additionally, every white pixel of the GT image on the right represents missing information due to inaccuracies of the expert system used to label it. We dealt with this issue by not considering these pixels when training and evaluating our methods.

Raincloud cover

Even after filtering out the dataset for images with more than 20% cloud coverage, there remained many cloudy images. If these were rain clouds, then the water contained within them would show up in the AWEI calculation

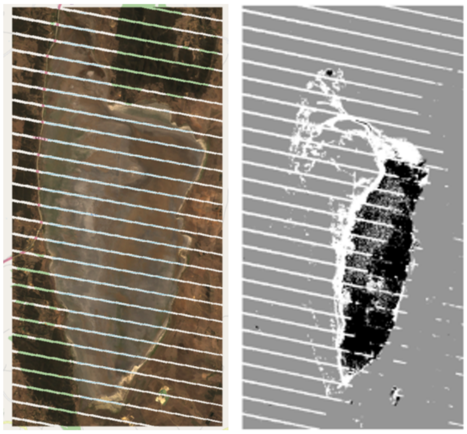

Fig. 2. Example of images with missing data

as an abnormally high value. This would have skewed the model results, so each image had its found water pixels (surface + rain clouds) compared to the pixels in the GT. If the match between these was sufficient the image was kept. In addition, images that had abnormally high AWEI values were dropped. This value is set by looking at the usual distribution of 'real' water values and those found in rain clouds.

Different projections
Even though the GT dataset used Landsat imagery the projections of the two differed. This caused different amounts of pixels being calculated in each as well as issues with visible stretching in any visualizations. As such, the extraction included a step where the GT was reprojected using the satellite image's projection.

4 METHOD

4.1 Data pre-processing

After performing the pre-processing steps presented in section 3 and shown in Figure 3, we computed the AWEI at pixel level because it performs very well compared to other water indices such as NDWI or MNDWI [2]. For this project, we chose the non-shadow version, as the selected lakes were not in a mountainous region.

4.2 Classification

We employed a naive thresholding method and a Random Forest Classifier (RFC) to label each pixel as water or non-water. We trained both methods using each group of three lakes and predicted the fourth one. Because of data imbalance, we measured the F1 score and additionally the Kappa index to check whether correct predictions occurred by chance.

In the case of thresholding, we tested 30 values taken linearly between the 5th and 95th percentile of all training AWEI values and applied to the AWEI feature as it was proven to be an effective baseline method [5].

The RFC was trained with 7 pixel-level features: the RGB color channels, the surface reflectance of type near-infrared, shortwave infrared 1, 2, and the AWEI feature. We chose them to give the model a holistic view of the

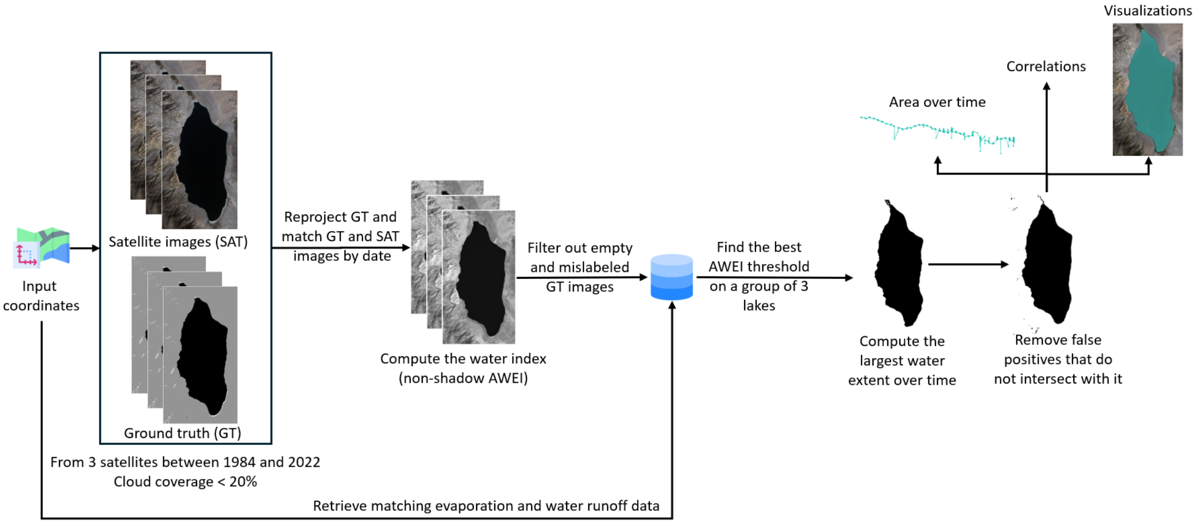

Fig. 3. Diagram of the proposed pipeline

data from the visible and infrared spectrum, in addition to the water index as experimented in [11]. We employed the 3-fold cross-validation method (a fold was represented by one lake at a time) to have better performance estimates while optimizing the model hyperparameters. In the appendix is presented the operated optimization space. We optimized the model because the initial performance was lower than expected and the model was overfitted.

4.3 Area estimation

After training, we detected the fourth lake from the test set, obtaining for each satellite image a binary picture. The number of false positives from areas around the lake was minimized by constraining the region of interest to the largest extent of the lake over time. This was achieved by dilating the image with a 3x3 kernel to bridge the eventual gaps between the bodies of water and finding the largest connected component. This was used as a mask and there were considered only the detected water bodies that intersected with it. This approach works effectively when the lake shrinks or expands in time. Finally, the holes in the detected lake that appeared due to missing data were filled with a binary closing applied with a kernel of 2x2. Knowing that the spatial resolution of Landsat images is 30 meters, we computed the lake area for an image by counting the water pixels and multiplying it with the area covered by one pixel.

4.4 Analysis

To validate the results, in addition to calculating performance values against the ground truth, we plotted the area as a time series along with an interactive detection visualization. We double-checked the estimated surface by looking up actual measurements online. Finally, we calculated the Spearman correlation of lake area with water runoff and surface evaporation. This measure of correlation was chosen because the scatterplots showed a non-linear relationship.

5 RESULTS

As can be seen in Figure 4, the thresholding method works very well, having an average F1 score on the train and test set of 0.9775 and 0.9839. Even the Kappa score shows a high level of agreement between predictions and ground truth, with values around 0.9, revealing the method does not have a good performance by chance. When it comes to the RFC from Scikit-learn [9], the model overfits the data a lot, with an F1 difference between the train and the test of 0.4251. Even though we tried to optimize the hyperparameters by using 90 combinations (Table 2) or pruning the Decision Trees, there have not been significant improvements. We identified two main solutions: to add more images or more features like vegetation index [11]. However, we did not have enough computing resources to add more data because the optimization process with cross-validation took a long time to run.

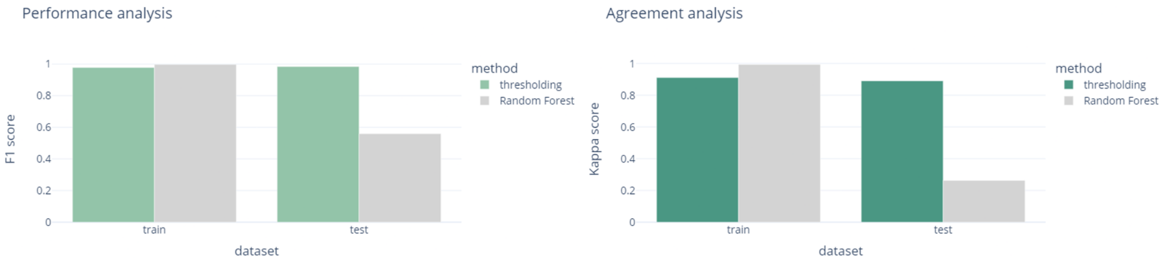

Fig. 4. Averaged metrics across all experiments

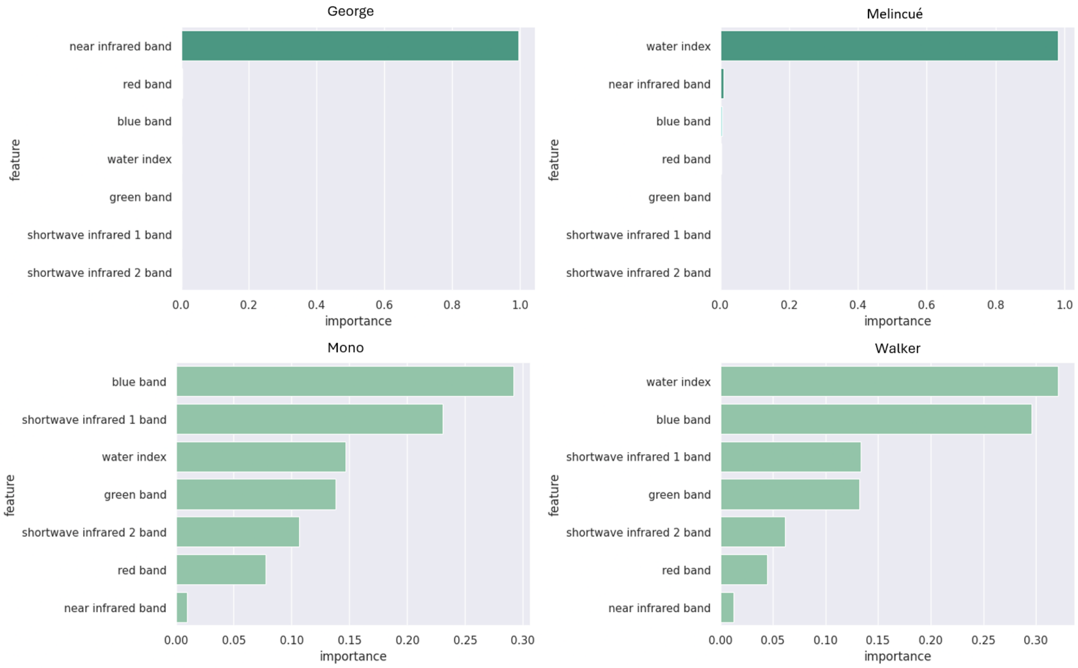

Fig. 5. Optimized RF feature importance after training to predict each fourth lake

To better understand the performance of the RF, we retrieved the feature importance scores (Figure 5). The model is not robust at all because while training to predict the test lake George or Melincué, the F1 score is larger, but the model is making use only of the near-infrared band or AWEI. On the contrary, when the performance is worse, all features are considered. We conclude that the data has too much variability and too few examples, and for the model to generalize we would need many more lakes in various environments.

Lake name	Train	Train optimized	Validation	Validation optimized
George	0.9990	0.9963	0.8568	0.8700
Melincué	0.9987	0.9965	0.8771	0.8814
Mono	0.9987	0.9954	0.7169	0.7172
Walker	0.9998	0.9998	0.7180	0.7183

Table 2. F1 score obtained by the RFC before and after hyperparameter optimization

The correlations between lake area and climate statistics are shown in Table 3. We can observe there is no relationship between surface evaporation and lake area, with an absolute correlation lower than 0.2. However, the water runoff is strongly correlated in the case of Lake George because it has forests around it, In the other cases we observe weak positive correlations. Lake Mono is in a desert, so the soil permeability is low and Lake Walker has a reduced watershed that leads to a small runoff. Lake Melincué has a low correlation because people are extracting water to irrigate the crops in the area. Therefore, this climate static can be used to some extent to explain the historic behavior of the lake area.

Lake name	Area - runoff	Area - evaporation
George	0.6122	-0.1017
Melincué	0.2025	0.0440
Mono	0.3314	0.1500
Walker	0.2025	0.0440

Table 3. Correlation of monthly lake area with water runoff and evaporation

Figure 6 shows the area evolution for each lake together with relevant detection examples. Interestingly, each lake has a different pattern: George had periods when it was empty, Melincué fluctuated a lot, Waker exhibits a decreasing trend and Mono is special because it has small islands inside. In every situation, our simple methods worked better than expected. We cross-check the estimated area against the ground truth and online-available information. When looking at the detection, we observe a small coverage of false positives. They mostly appeared because of errors in the ground truth. Visually, our method surpassed the ground-truth quality for these four lakes, but we could not quantify by how much.

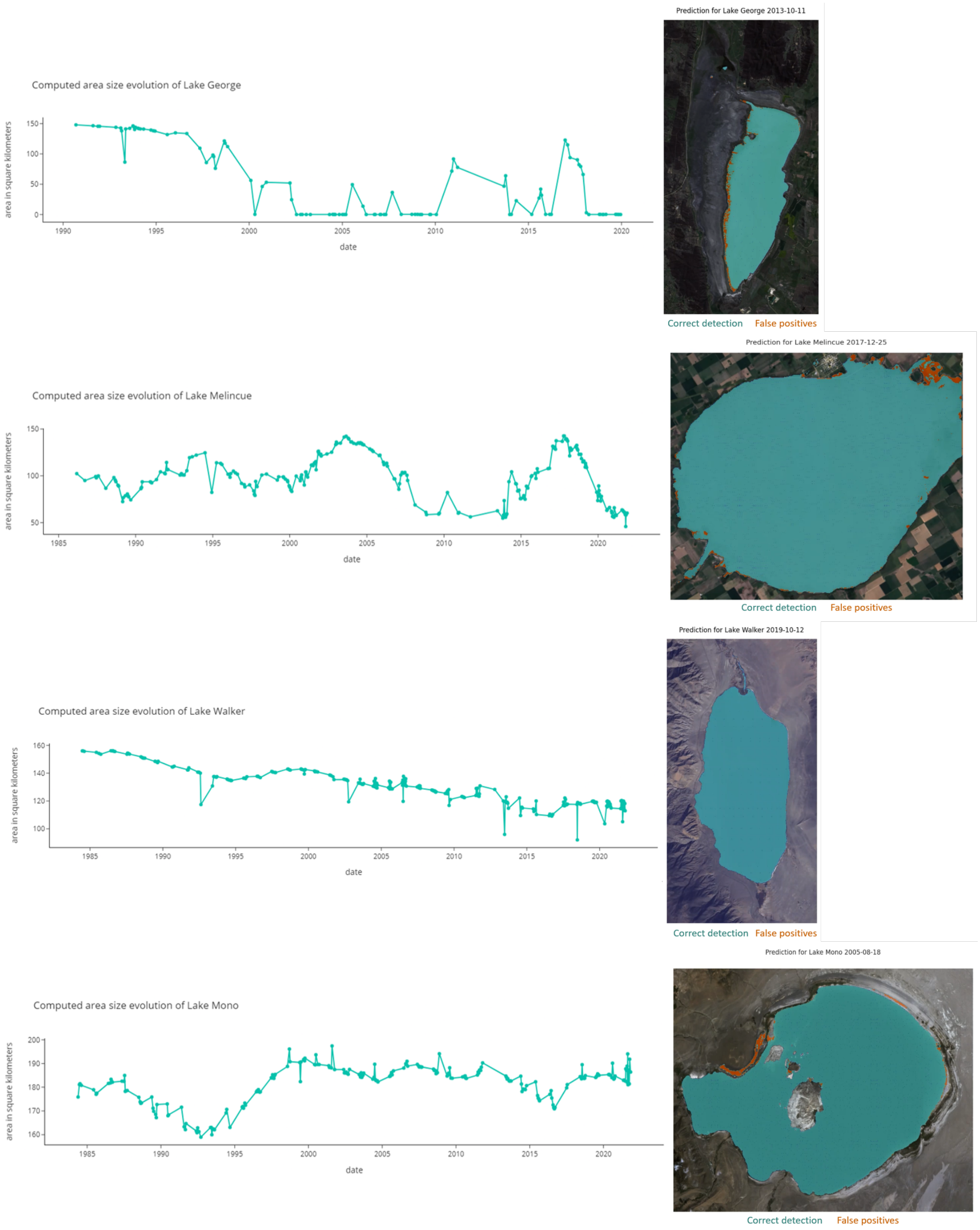

Fig. 6. Estimated time series area with the thresholding method along with a detection example for each analyzed lake

6 RELIABILITY OF RESULTS

While our method is simplistic, we proved its reliability for estimating the area of four lakes from different environments and consider it a good baseline that can be utilized for other lakes. However, we highlight below its limitations together with suggestions for future improvements (We did not implement them because of time constraints.):

- We can only analyze lakes that entirely fall within a single satellite image → Develop a script that can merge partial-pass images into one.
- The non-shadow AWEI struggles for lakes in mountainous regions → Use a GIS lookup to check if the region of interest (ROI) is in a mountainous area and if yes, switch the index being used to shadow AWEI.
- Arbitrary selection of image filtration parameters → Develop a pre-extraction tool that can create a histogram of AWEI values for the ROI and let the user fine-tune the specific cutoff.
- The dataset can be biased because retrieved images have least a 40% match between AWEI and GT → Use a combination of a match between AWEI and GT with the number of pixels that are above the high-value threshold to remove highly cloudy images.
- Cannot forecast the lake area due to irregular observations → We tried to solve this issue by using data from multiple Landsat satellites, but even so, too many images are cloudy, especially during winters. A solution can be the retrieval of images from satellites like Sentinel.
- The RFC model is overfited → Train it with more images, craft new features like the vegetation index and potentially ensemble it with the thresholding method and a third model to improve the reliability.
- Missing data restricts us from checking the models' performance for each pixel → We have not found a better ground truth dataset so manual labeling remains the last solution.

7 TECHNICAL DEPTH

We used a combination of image processing, data visualization, and machine learning techniques along with knowledge of water balance. For example, we applied image processing operations such as dilation, closing, and connected component detection, all learned during the Image Processing and Computer Vision course. We involved them using SciPy and OpenCV. Furthermore, we designed the optimization experiments, defined the hyperparameter space, and optimized the RFC, while using the k-fold cross-validation method to obtain better performance estimates. These concepts were learned while working on personal projects and doing the undergraduate thesis. Interactive visualizations with libraries such as IPyWidgets, Seaborn, Matplotlib, and Plotly allowed us to easily visualize how the extent of the lakes evolved. We even looked for metrics suitable for the context of water detection from satellite imagery and noticed that many researchers use the Kappa index [1], [16], so we incorporated it into our project. Finally, prior knowledge gained during the water management course helped us make inferences about climate correlations.

8 CONCLUSIONS & RECOMMENDATIONS

We have been successful in creating a hassle-free pipeline that accepts a set of coordinates and returns a historical timeline for water evolution in the given area, along with correlations with climate statistics such as runoff and water evaporation. Even though the thresholding method seems simple, the key ingredients are the search for optimal values to maximize the F1 score, the AWEI feature it applies to, and because we post-process the detections to reduce false positives. Thus. our method produces results for the four seasonal lakes that exceed our expectations.

There are a variety of stakeholders that could benefit from our model; for example, developers who want to build on the shore of a lake, farmers who want to irrigate their crops, or conservation agencies who want to track the size of a lake. All of these could benefit our model, but it should be noted that it only provides a historical evolution of the area and not a projection into the future. Therefore, these actors should always make their own inferences using the results of our model for their specific case.

Unfortunately, the method is unsuitable for forecasting, so for future research, we recommend looking for methods that are more robust to high cloud cover so that more frequent observations can be extracted. In addition, we advise training the RFC model with more features and more images to increase its generalization capability.

9 REFLECTION

Three challenges stand out for this project: filtering poor-quality images, extracting the best (most representative) extents of a lake, and using the RFC model. The way these were dealt with is described in the Source Data and Method sections respectively. The GIS course did well at introducing the basic concepts of GIS and satellite image analysis. Moreover, it gave us a useful starting point for using machine learning in the context of geospatial data. It included a practical presentation of different data sources and methods possible to do the analysis, but the actual procedure applied here was learned by working with the model and gradually improving on it. Given that we were comparing different images with different projections, it would have been good to discuss the implications of different coordinate reference systems in the course.

The results our model showed were overall really good, with it performing at times far better than the GT. Of course, that is somewhat countered by the model only being fed images that already somewhat coincide with the GT to address the cloud issue.

ChatGPT was a big help for getting the correct Python functions to perform many of the steps in the image extraction; examples include calculating the number of pixels with data, comparing different image bands, and creating the AWEI expression.

REFERENCES

[1] Carolina Doña, Ni-Bin Chang, Vicente Caselles, Juan Manuel Sánchez, Lluís Pérez-Planells, Maria Del Mar Bisquert, Vicente García-Santos, Sanaz Imen, and Antonio Camacho. 2016. Monitoring Hydrological Patterns of Temporary Lakes Using Remote Sensing and Machine Learning Models: Case Study of La Mancha Húmeda Biosphere Reserve in Central Spain. *Remote Sensing* 8, 8 (2016). https://doi.org/10.3390/rs8080618

[2] Gudina L. Feyisa, Henrik Meilby, Rasmus Fensholt, and Simon R. Proud. 2014. Automated Water Extraction Index: A new technique for surface water mapping using Landsat imagery. *Remote Sensing of Environment* 140 (2014), 23–35. https://doi.org/10.1016/j.rse.2013.08.029

[3] Google. [n. d.]. ERA5-Land Monthly Aggregated - ECMWF Climate Reanalysis. https://developers.google.com/earth-engine/datasets/catalog/ECMWF_ERA5_LAND_MONTHLY_AGGR

[4] EC JRC / Google. [n. d.]. JRC Monthly Water History, v1.4. https://developers.google.com/earth-engine/datasets/catalog/JRC_GSW1_4_MonthlyHistory?hl=en

[5] Hao Jiang, Min Feng, Yunqiang Zhu, Ning Lu, Jianxi Huang, and Tong Xiao. 2014. An Automated Method for Extracting Rivers and Lakes from Landsat Imagery. *Remote Sensing* 6, 6 (2014), 5067–5089. https://doi.org/10.3390/rs6065067

[6] Government of Canada. 2015. Pre-processing. https://natural-resources.canada.ca/maps-tools-and-publications/satellite-imagery-and-air-photos/tutorial-fundamentals-remote-sensing/image-interpretation-analysis/pre-processing/9403

[7] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. 2012. How many trees in a random forest?. In *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*. Springer, 154–168.

[8] Yunxuan Pang, Junchuan Yu, Laidian Xi, Daqing Ge, Ping Zhou, Changhong Hou, Peng He, and Liu Zhao. 2024. Remote Sensing Extraction of Lakes on the Tibetan Plateau Based on the Google Earth Engine and Deep Learning. *Remote Sensing* 16, 3 (2024). https://doi.org/10.3390/rs16030583

[9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[10] Jean-François Pekel, Andrew Cottam, Noel Gorelick, and Alan S Belward. 2016. High-resolution mapping of global surface water and its long-term changes. *Nature* 540, 7633 (2016), 418–422.

[11] Thanh Noi Phan, Verena Kuch, and Lukas W. Lehnert. 2020. Land Cover Classification using Google Earth Engine and Random Forest Classifier—The Role of Image Composition. *Remote Sensing* 12, 15 (2020). https://doi.org/10.3390/rs12152411

[12] Yunlei Sun, Huiquan Gong, Yucong Li, and Dalin Zhang. 2019. Hyperparameter importance analysis based on n-rrelieff algorithm. *International Journal of Computers Communications & Control* 14, 4 (2019), 557–573.

[13] USGS. [n. d.]. USGS Landsat 5 Level 2, Collection 2, Tier 1. https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LT05_C02_T1_L2

[14] USGS. [n. d.]. USGS Landsat 7 Level 2, Collection 2, Tier 1. https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LE07_C02_T1_L2

[15] USGS. [n. d.]. USGS Landsat 8 Level 2, Collection 2, Tier 1. https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C02_T1_L2

[16] Jiajun Zuo, Wenliang Jiang, Qiang Li, and Yankai Du. 2024. Remote sensing dynamic monitoring of the flood season area of Poyang Lake over the past two decades. *Natural Hazards Research* 4, 1 (2024), 8–19. https://doi.org/10.1016/j.nhres.2023.12.017

APPENDIX

Details of the Random Forest Optimization

In total, we tested 90 combinations of hyperparameters. The search space of the hyperparameters has the following four dimensions:

- number of Decision Trees: 64, 96, 128
- the percentage of samples used to train a Decision Tree: 0.5, 0.7, 1.0
- minimum number of samples in a leaf: 1, 5, 10, 15, 20
- the number of features to consider when looking for the best-split: sqrt, all

We chose these hyperparameters and values based on a literature review. As recommended in [12], we picked the minimum number of samples in a leaf as a primary hyperparameter that stops the process of growing a Decision Tree, leading to less overfitted estimators. The number of trained Decision Trees in the Random Forest is not that large, considering that after around 100 trees, a marginal improvement is obtained by adding more [7].