# EFFICIENT AGGREGATE LAND COVER QUERIES WITH CLOUD-OPTIMIZED RASTER FORMATS

Luke McQuade
*Department of Geoinformatics – Z_GIS*
*Paris Lodron Universität Salzburg*
Salzburg, Austria
luke.mcquade@plus.ac.at

Martin Sudmanns
*Department of Geoinformatics – Z_GIS*
*Paris Lodron Universität Salzburg*
Salzburg, Austria
martin.sudmanns@plus.ac.at

Dirk Tiede
*Department of Geoinformatics – Z_GIS*
*Paris Lodron Universität Salzburg*
Salzburg, Austria
dirk.tiede@plus.ac.at

***Abstract*—Semantic queries of Earth observation (EO) imagery, such as "Find the images with less than 5% cloud cover in this area between two dates", rely on aggregating some kind of scene classification data. Given the millions of pixels in a typical image, this can be resource-intensive. If knowledge is required of only the relative distribution of classes, though, do we need to process every pixel? The size and shape (morphology) of natural features, when observed by high-resolution optical sensors such as Sentinel-2, mean that simple downsampling can be used to drastically reduce the processing required for such queries, with only small losses in accuracy. Given an error tolerance of 1%, memory usage can be reduced by 625× and query runtime by 12×, for areas of interest of 60 km × 60 km. Using cloud native technologies such as Cloud Optimized GeoTIFF (COG), this can also lead to significant reductions in network and disk usage.**

***Index Terms*—scene classification, land cover, Sentinel-2, cloud-native geospatial**

## I. Introduction

Some typical use cases of satellite Earth Observation (EO) datasets are queries such as "Find the images with less than 5% cloud cover between two dates", or "Find the first image of the year where the vegetation content of this area rises above 20%". These require some kind of scene classification information, which is then aggregated to the user's area of interest. Given the millions of pixels in a typical image, and dozens to hundreds of images to query against, this can be resource-intensive. The natural features that underlie classification maps often exhibit some degree of spatial autocorrelation [3], [17], so if we are querying against only the relative distribution of classes rather than their precise location, do we need to process every pixel? In other words, to what extent does downsampling (upscaling) affect query accuracy?

The multispectral instrument (MSI) of the Copernicus Sentinel-2 mission captures data from 13 optical bands at spatial resolutions of 10 m to 60 m (ground sampling distance). From these, Level-2A (L2A) analysis-ready products are created. One component of this is the Scene Classification Layer (SCL)—an auxiliary information layer generated with a rules-based classifier, comprising twelve classes covering various land cover types, cloud, and anomalous readings (see Table I) [5].

Cloud Optimized GeoTIFF (COG) is a geospatial raster file format where data are stored as tiles, such that spatial subsets can be read without needing to load the entire file. This subsetting works over a network using, e.g., HTTP range requests. The underlying GeoTIFF offers pyramid/overview features that allow access to data at different scales. This can drastically lower the data transfer and processing costs of geospatial analysis [2]. But whether this can be leveraged depends on the use case, two factors of which are the size of the user's area of interest (AoI) and the scale (areal unit size).

TABLE I
SCL Classes. Source: [5].

| Label | Class |
|---|---|
| 0 | No data |
| 1 | Saturated or defective |
| 2 | Dark area pixels |
| 3 | Cloud shadows |
| 4 | Vegetation |
| 5 | Bare soils |
| 6 | Water |
| 7 | Unclassified |
| 8 | Cloud - medium probability[a] |
| 9 | Cloud - high probability[a] |
| 10 | Thin cirrus[a] |
| 11 | Snow/ice |

[a] Combined in our analysis.

The Microsoft Planetary Computer (MSPC) platform [11] hosts a Sentinel-2 L2A data archive, including the SCLs. The asset data is stored as COGs, which can be located via a Spatiotemporal Asset Catalog (STAC) metadata search and retrieved via HTTP(S). The SCL data is stored as a raster with 20 m spatial resolution [10]. A typical S2 image granule spans approximately 110 km × 110 km [6]. An entire SCL raster, therefore, has around 30 megapixels. The class distribution for the entire scene is actually

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation program under the Grant Agreement No. 101082493 (Project: LEONSEGS)

pre-computed and stored in the metadata, accessible by parameters such as `s2:vegetation_percentage` and `s2:snow_ice_percentage` [12]. This is useful if a user's AoI matches the granule's footprint, but this is often not the case, e.g., for analyses based on administrative boundaries.

The role of scale in remote sensing has been covered extensively in [19], but this was before the time of the Sentinel missions, COGs, and Big Earth Data. Later works have revisited the topic, exploring different resampling approaches and demonstrating the effects of downsampling in a range of other settings, such as classification and land cover accuracy on a pixel-by-pixel basis [13], [14], [20], [21].

## II. Methods

### A. Study areas and period

Two study areas were selected: an area of the Douro Valley, Spain (between 41.47°N to 42.57°N latitude and 3.57°W to 5.40°W longitude), and Stockerau, Lower Austria (between 47.78°N to 48.87°N latitude and 15.01°E to 16.65°E longitude), representing two different agricultural areas in Europe with different land use/land cover. The study period was 1st June to 1st July 2023, when the expected conditions for these areas of Europe would be a mix of cloudy and clear days, with significant vegetation growth.

### B. Downsampling analysis

Based on the study areas, a series of synthetic AoIs were generated: square tiles of side lengths (sizes) varying from 1 km to 60 km, using the coordinate reference system (CRS) of the corresponding Universal Transverse Mercator (UTM) grid zone. For reference purposes, these were grouped by size. An overview is provided in Table II.

TABLE II
Summary of AoIs and Valid Image Samples

| AoI tile size (m) | Size group | # AoIs | Valid samples[b] |
|---|---|---|---|
| 1000 | Small | 123 | 2315 |
| 2000 | Small | 37 | 680 |
| 3000 | Small | 19 | 340 |
| 4000 | Small | 12 | 210 |
| 6000 | Medium | 21 | 345 |
| 8000 | Medium, Large | 43 | 694 |
| 10 000 | Large | 15 | 214 |
| 12 000 | Medium, Large | 22 | 348 |
| 20 000 | Large, Extra-large | 76 | 1021 |
| 40 000 | Extra-large | 8 | 82 |
| 60 000 | Extra-large | 2 | 20 |

[b] Matching, complete image chips.

For each AoI, a STAC search was performed. For each matching image, the SCL was windowed to the AoI, and the class distribution (as percentages) at native resolution (20 m) was calculated. The SCL was then queried at a series of lower resolutions using regular nearest-neighbor downsampling. The resolutions were selected using a loosely exponential pattern, representing a variety of analytical scales: 50 m, 100 m, 200 m, 500 m, and 1000 m.

The class distributions were calculated for each of the downsampled variants and compared against that of the native resolution. The maximum absolute difference in percentages, which we refer to as the maximum classification error, is our key metric; it directly informs the accuracy of our use case queries. An illustrative example is given in Table III.

TABLE III
Hypothetical Class Distribution Change

| | Vegetation | Bare soil | Cloud |
|---|---|---|---|
| Native (20m) | 10% | 10% | 80% |
| Downsampled (100m) | 8% | 11% | 81% |
| Difference | −2% | +1% | +1% |
| Absolute difference | 2%[c] | 1% | 1% |

[c] Maximum classification error

Some preliminary results showed low spatial autocorrelation among the cloud classes (8, 9, 10) producing high eventual maximum classification errors. However, our use cases suggested that an aggressive approach to cloud filtering would be appropriate, and these classes were combined via summation of their percentages.

Sometimes an AoI would only partially overlap with the footprint of an image granule, yielding areas of 'no data', or the image data itself would contain 'no data' values. These were excluded from the class distribution comparisons. Valid samples, as in Table II, therefore refer to image chips that contain zero 'no data' values.

The maximum classification errors were compared across all AoIs for each tile size and downsampling resolution. From this, we could determine the distributions of classification errors due to downsampling, and where the maximum exceeded a user-defined threshold/tolerance (1% as an initial suggestion).

The analysis was performed with Python 3.13 in a `marimo` notebook [1], using packages `pystac_client` [7] and `stackstac` [9] for data loading, packages `xarray` [8], `pandas` [16], `geopandas` [4] and `scipy` [18] for data processing and analysis, and `holoviews` [15] for visualization. This code was running on an HP EliteDesk workstation with an Intel Core i5-8500 processor (3 GHz, 6 cores) and a 256 GB solid-state system drive, in a Windows Subsystem for Linux (WSL) Docker container with 24 GB of memory available.

### C. Performance

For each AoI, we timed the duration of querying the matching SCL assets from MSPC and calculating the class distributions at a given resolution. This consisted of remote cloud computation, storage access and network transfer, as well as local computation. The measurements are, therefore, highly subject to external conditions such as network, storage and computation contention, and various types of caching. Samples containing 'no data' values were included in these measurements. There were around 10 to 20 images matching each AoI in the date range; the duration to load and process all of these was recorded and divided by the number of

images, giving a mean per-scene runtime. The number of pixels in each image was also recorded, as this, rather than physical size, determined the amount of data involved. These ranged from single-pixel images (as in a heavily downsampled small AoI) to 9 megapixels (as in a 60 km size tile at native resolution). The Python standard library's `time` module was used. Note that the duration of the initial STAC search to find the matching images for the AoI was not included.

## III. Results

Fig. 1 shows the distributions of the maximum classification errors at a downsampling resolution of 50 m for AoIs in the Small subset. Here it can be seen that maximum classification errors approach 6% at tile sizes of 1 km but fall under 1% at tile sizes of 4 km. For readability, similarly detailed presentations of the results for the remaining resolutions and AoIs have been omitted. However, the key results are summarized in Table V, which shows the maximum downsampling resolution for each tile size before the maximum classification error exceeds a tolerance threshold of 1%. Note how the maximum downsampling resolution rises to 500 m at AoI tile sizes of 40 km.

A summary of the performance measurements by pixel count (grouped exponentially due to the wide range of magnitudes) is given in Table IV. Individual data are shown in Fig. 2. Together, these show that the average runtime is approximately constant for images up to around 1000 pixels, before steadily increasing with the number of pixels.

Two queries were excessively long-running compared with those of similar pixel counts and were deemed outliers. It was observed that memory usage was very high during this time, and paging to disk could be the cause.

A simple least-squares linear regression model fits the data (with outliers removed), with a slope of $2.07 \times 10^{-8}$ s/pixel, $y$-intercept of 0.016 s, and an $R^2$ of 0.77. From this, the query speed-up from downsampling can be estimated. These are shown for each tile size in Table V. Downsampling offers a mild benefit to query runtimes for AoIs of 10 km or less, then reductions quickly increase with larger AoI sizes.

TABLE IV
Summary of Mean-per-scene Query Runtimes

| Image size (pixels) | Query count | Min. (s) | Max. (s) | Median (s) |
|---|---|---|---|---|
| 1 to 10 | 302 | 0.009 | 0.054 | 0.012 |
| 10 to 100 | 267 | 0.009 | 0.061 | 0.012 |
| $10^2$ to $10^3$ | 607 | 0.009 | 0.042 | 0.013 |
| $10^3$ to $10^4$ | 469 | 0.010 | 0.054 | 0.015 |
| $10^4$ to $10^5$ | 417 | 0.010 | 0.050 | 0.017 |
| $10^5$ to $10^6$ | 202 | 0.012 | 0.077 | 0.023 |
| $10^6$ to $10^7$ | 112 | 0.026 | 9.781 | 0.041 |

## IV. Discussion and Conclusion

Our results show that downsampling can significantly reduce the data usage and runtime of aggregate, distribution-based land cover queries, if afforded a small tolerance for error: up to a 625× data reduction and estimated 12× decrease in query runtime for the largest AoIs in our experiment. For smaller AoIs, the benefits were smaller. In terms of class distribution error, smaller AoIs are more sensitive. In terms of performance, the smaller number of pixels meant they occupied the 'sill' where runtime was predominantly made up of network latencies rather than data processing.

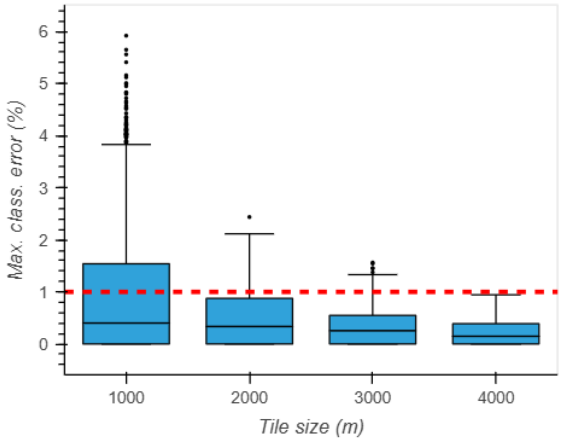


Fig. 1. Distributions of the maximum classification errors at a downsampling resolution of 50m, for the Small subset of AoIs. Quartiles are represented by the boxes, data within 1.5 × interquartile range by whiskers, and outliers as dots. The dashed red line marks a 1% error tolerance threshold.

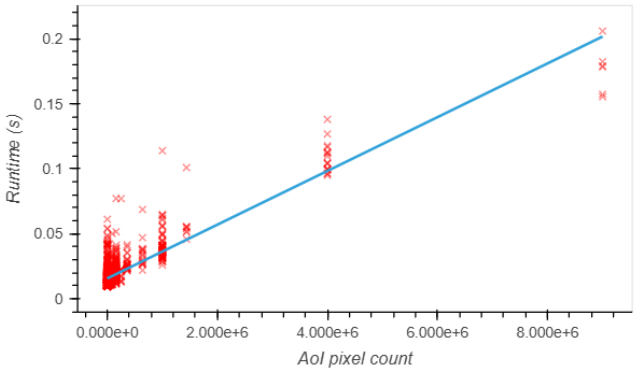


Fig. 2. Query runtime, mean per-scene. The blue line is a linear trendline, with $R^2 = 0.77$ and $y$-intercept= 0.016s. 2 outliers (per-scene runtime $> 1.0$s) have been omitted.

The average number of valid samples per AoI may seem high, given the revisit times of Sentinel-2 and the study period of only one month. This is partly explained by AoIs matching more than one granule footprint, but is mainly due to the STAC search returning several reprocessed versions of the original data. This behavior was uniform across all AoIs and therefore should not affect the conclusions drawn from the results.

There are likely more optimal resolutions to which AoIs can be downsampled, especially for the smallest AoIs and those larger than we tested. The experiment was kept constrained to avoid excessive use of cloud resources (a full run completes in approximately 1 h).

The linear regression model used to estimate the potential

TABLE V
MAXIMUM DOWNSAMPLING RESOLUTION BEFORE MAXIMUM CLASSIFICATION ERROR EXCEEDS 1% ("N/A", OTHERWISE), AND ASSOCIATED DATA AND ESTIMATED PER-SCENE QUERY RUNTIME REDUCTIONS

| Tile size (m) | Max. downsampling resolution (m) | Pixels (native) | Pixels (downsampled) | Data reduction (multiple) | Est. runtime (native) | Est. runtime (downsampled) | Est. runtime reduction (multiple) |
|---|---|---|---|---|---|---|---|
| 1000 | N/A | 2500 | N/A | N/A | 0.0157 | N/A | N/A |
| 2000 | N/A | 10 000 | N/A | N/A | 0.0159 | N/A | N/A |
| 3000 | N/A | 22 500 | N/A | N/A | 0.0161 | N/A | N/A |
| 4000 | 50 | 40 000 | 6400 | 6.25 | 0.0165 | 0.015 78 | 1.04 |
| 6000 | 50 | 90 000 | 14 400 | 6.25 | 0.0175 | 0.015 95 | 1.10 |
| 8000 | 50 | 160 000 | 25 600 | 6.25 | 0.0190 | 0.016 18 | 1.17 |
| 10 000 | 100 | 250 000 | 10 000 | 25 | 0.0208 | 0.015 86 | 1.31 |
| 12 000 | 100 | 360 000 | 14 400 | 25 | 0.0231 | 0.015 95 | 1.45 |
| 20 000 | 200 | 1 000 000 | 10 000 | 100 | 0.0363 | 0.015 86 | 2.29 |
| 40 000 | 500 | 4 000 000 | 6400 | 625 | 0.0983 | 0.015 78 | 6.23 |
| 60 000 | 500 | 9 000 000 | 14 400 | 625 | 0.2016 | 0.015 95 | 12.64 |

speed-ups from downsampling is also likely less than optimal, and this should be considered before extrapolating or expanding on the resulting numbers.

Only uniform, regular grid downsampling was used here. Other methods, such as the adaptive approach of Mîrț et al. [13], could perform better in terms of maintaining class distribution. But, this could lose the data handling performance benefits tied to the rectangular-tiled COGs.

Since the COG already stores pyramids with layers having different resolutions, there is no additional storage required on the server side. If the COG internal tile size and overview structure are known ahead of time, adjusting the downsampling resolution to match these could lead to additional performance gains. As a corollary, our results could inform the parameters for generating future COGs.

Repeating the approach on additional, geographically diverse study areas would be a useful extension of this work, as would applying it to land cover maps beyond the Sentinel-2 SCL, or even other binned/categorical data types.

## ACKNOWLEDGMENT

Many thanks to our colleague, Felix Kröber, whose earlier explorations and Jupyter notebooks on Sentinel-2 STACs and the SCL were a great help; and to the Microsoft Planetary Computer team for making their service available to us.

## REFERENCES

[1] A. Agrawal and M. Scolnick, "marimo," Apr. 2025, version: v0.12.9. [Online]. Available: https://github.com/marimo-team/marimo
[2] C. O. G. Community, "Cloud Optimized GeoTIFF," 2025. [Online]. Available: https://cogeo.org/
[3] P. J. Curran, "The semivariogram in remote sensing: An introduction," *Remote Sensing of Environment*, vol. 24, no. 3, pp. 493–507, Apr. 1988.
[4] J. V. den Bossche et al., "geopandas/geopandas: v1.0.1," Jul. 2024. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.12625316
[5] European Space Agency, "Sentinel-2 Level-2A Algorithm Theoretical Basis Document," ESA – European Space Agency, Frascati, Italy, Technical Report S2-PDGS-MPC-ATBD-L2A, Nov. 2021, version: 2.10.
[6] ——, "Sentinel-2 Products Specification Document," ESA – European Space Agency, Frascati, Italy, Tech. Rep. S2-PDGS-TAS-DI-PSD, Mar. 2021.
[7] M. Hanson et al., "stac-utils/pystac-client," Feb. 2025, version: v0.8.6. [Online]. Available: https://github.com/stac-utils/pystac-client
[8] S. Hoyer et al., "xarray," Mar. 2025, version: v2025.03.1. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.15110615
[9] G. Joseph, "gjoseph92/stackstac," Aug. 2024, version: v0.5.1. [Online]. Available: https://github.com/gjoseph92/stackstac
[10] Microsoft, "Sentinel-2 Level-2A | Microsoft Planetary Computer," accessed: May 13 2025. [Online]. Available: https://planetarycomputer.microsoft.com/dataset/sentinel-2-l2a
[11] Microsoft Open Source, M. McFarland, R. Emanuele, D. Morris, and T. Augspurger, "microsoft/PlanetaryComputer: October 2022," Oct. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7261897
[12] M. Mohr and P. Varner, "stac-extensions/sentinel-2," Nov. 2023, version: v1.0.0. [Online]. Available: https://github.com/stac-extensions/sentinel-2
[13] A. Mîrț, J. Reiche, J. Verbesselt, and M. Herold, "A Downsampling Method Addressing the Modifiable Areal Unit Problem in Remote Sensing," *Remote Sensing*, vol. 14, no. 21, p. 5538, Jan. 2022.
[14] R. Raj, N. A. S. Hamm, and Y. Kant, "Analysing the effect of different aggregation approaches on remotely sensed data," *International Journal of Remote Sensing*, Jul. 2013.
[15] P. Rudiger et al., "holoviz/holoviews: Version 1.20.2," Mar. 2025. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.15019128
[16] The pandas development team, "pandas-dev/pandas: Pandas," Sep. 2024. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.13819579
[17] W. R. Tobler, "A Computer Movie Simulating Urban Growth in the Detroit Region," *Economic Geography*, Jun. 1970, publisher: Routledge.
[18] P. Virtanen et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020.
[19] S. Weigel, "Scale, Resolution and Resampling: Representation and Analysis of Remotely Sensed Landscapes Across Scale in Geographic Information Systems." *LSU Historical Dissertations and Theses*, Jan. 1996.
[20] H. Wu and Z.-L. Li, "Scale Issues in Remote Sensing: A Review on Analysis, Processing and Modeling," *Sensors*, vol. 9, no. 3, pp. 1768–1793, Mar. 2009. [Online]. Available: https://www.mdpi.com/1424-8220/9/3/1768
[21] K. Xu, Q. Tian, Y. Yang, J. Yue, and S. Tang, "How up-scaling of remote-sensing images affects land-cover classification by comparison with multiscale satellite images," *International Journal of Remote Sensing*, Apr. 2019.