Reproducibility Review of: Maximizing Data Coverage: Fusing Street View and Oblique Imagery to Quantify Vertical Greenery Potential in Urban Areas

Item	Value
Title	Maximizing Data Coverage: Fusing Street View and Oblique Imagery to Quantify Vertical Greenery Potential in Urban Areas
Authors	Aruscha Kramm (0009-0002-7801-9388), André Ludwig, Bogdan Franczyk
Ref. paper	Kramm, A., Ludwig, A., and Franczyk, B.: Maximizing Data Coverage: Fusing Street View and Oblique Imagery to Quantify Vertical Greenery Potential in Urban Areas, AGILE GIScience Ser., 7, 8, https://doi.org/10.5194/agile-giss-7-8-2026, 2026.
Codechecker(s)	Jeonghwan Choi (0009-0002-5027-1151)
Date of Check	2026.03.13
Summary	Partially reproducible
Repository	https://github.com/reproducible-agile/Quantify-VGS
Ref. certificate	https://doi.org/10.17605/OSF.IO/HETV8

Table 1: Reproduction metadata

Output	Comment
~result/orientation_orig.png	Distribution of façade orientations derived from LoD2 preprocessing. This plot confirms successful preprocessing; it does not directly correspond to a manuscript figure.
~result/wwr_orig.png	Distribution of window-to-wall ratio values derived from the fusion workflow. This plot demonstrates computed façade-level features.
~result/swa_orig.png	Distribution of façade surface area values derived from the fusion workflow. This plot illustrates computed façade features.
~result/index_weighted.png	Weighted vertical greenery potential index values calculated from the fused façade features. This plot shows the successful execution of the index calculation workflow.

Table 2: Summary of output files generated

Summary

The paper includes a comprehensive Data and Software Availability section, and the repository's README provides a clear outline of the workflow steps. The authors provided a public GitHub repository containing code for preprocessing LoD2 data, computing building façade features, performing image feature fusion, and calculating a vertical greenery potential index.

Due to licensing restrictions, some parts of the workflow cannot be made publicly available, including the street-view and oblique image extraction pipeline, proprietary data sources, and trained object detection models.

During the review process, I identified an issue in the preprocessing script, which the authors subsequently fixed. The repository was also updated to include example data for four walls (stored in the dataframes directory), allowing execution of the downstream workflow without requiring access to proprietary image sources or trained models.

With the updated repository and publicly available example data, I successfully executed the downstream workflow (feature fusion and index calculation).

Reproducibility reviewer notes

Installation prerequisites and computational environment

The reproducibility check was conducted using Python 3.14.3 in a standard local environment. During the review process, the author indicated that the original workflow was developed using Python 3.12.11.

The test version of the workflow does not require the machine learning libraries used in the original pipeline, as the trained object detection models cannot be shared. According to the author, only commonly used Python packages (e.g., pandas, NumPy, Shapely, and OpenCV) were required to run the provided example workflows.

A dedicated environment specification (e.g., requirements.txt or environment.yml) was not available in the repository. However, the scripts were executed successfully in the reviewer’s environment without any compatibility issues.

Data preparation

The GitHub repository provides sample LoD2 building data required for the preprocessing step. In addition, the repository includes example intermediate data for four walls (stored as .pkl files in the dataframes directory), enabling execution of the downstream fusion and index calculation steps.

However, the complete workflow described in the paper relies on additional inputs that cannot be shared publicly, including:

- proprietary street-view and oblique imagery
- the image extraction pipeline
- trained object detection models

As described in the repository README, users must construct intermediate data structures (e.g., detection outputs and image metadata dataframes) when applying the workflow to new data.

Running the code

The reproducibility check followed the workflow described in the repository README.

LoD2 preprocessing

The following scripts were executed using the sample LoD2 data provided in the repository:

- 01_read_lod_data.py
- 02_compute_features_refactored.py

During the initial execution, I encountered an error in the wall-merging stage. After reporting the issue to the author, the repository was updated, and the problem was corrected. The scripts were then executed successfully.

Feature fusion workflow

The following script was executed using the example detection outputs provided in the repository (dataframes directory):

- compute_ai_image_features_fused.py

This workflow successfully generated façade-level features required for the potential index calculation.

Potential index calculation

The following notebook was executed using the created feature data:

- calculate_potential_index.ipynb

The notebook successfully calculated potential index values and generated plots illustrating the distributions of façade features and index values.

Since the example dataset contains only four walls, I was able to verify that the downstream computational workflow executes correctly, rather than reproducing the full numerical results reported in the manuscript based on the original dataset.

Outputs and results

Running the workflow produced several plots illustrating distributions of façade characteristics and the calculated potential index. These include:

- façade orientation distributions
- window-to-wall ratio values
- façade surface area values
- calculated vertical greenery potential index values

The following figures were produced from the four-wall example dataset and demonstrate the execution of the downstream workflow. The regenerated plots are consistent with those produced by the provided notebook in the repository (calculate_potential_index.ipynb).

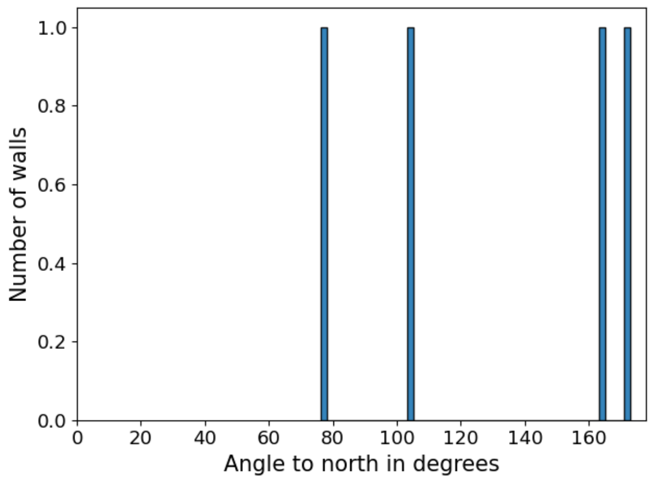

Figure 1. Orientation distribution (Regenerated plot).

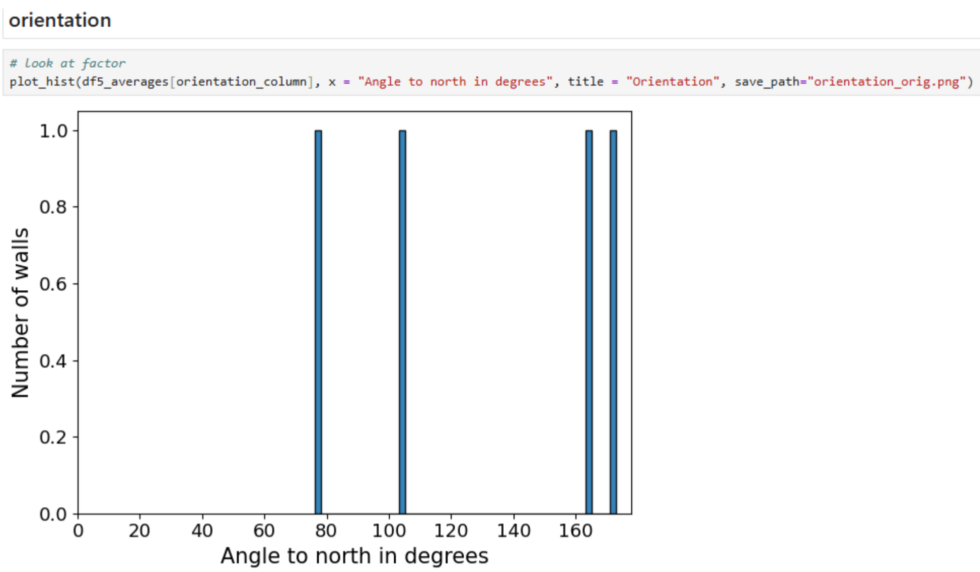

Figure 2. Orientation distribution (calculate_potential_index.ipynb).

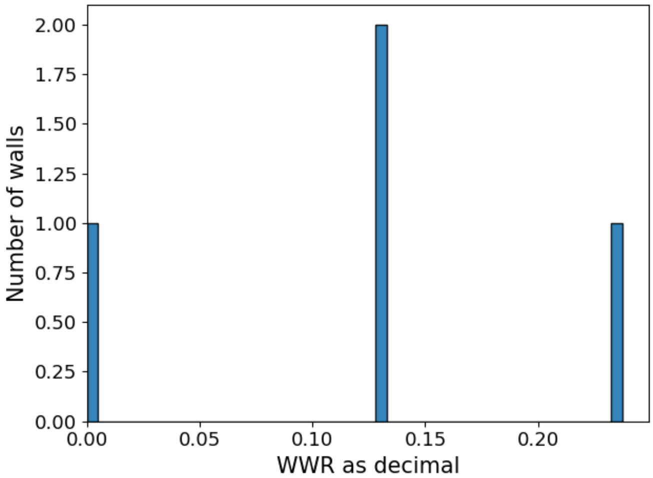

Figure 3. Window-to-wall ratio values (Regenerated plot).

wwr_corrected

```
# look at factor
plot_hist(df5_averages[wwr_column], x = "WWR as decimal", title = "Window-to-wall ratio", save_path="wwr_orig.png")
```

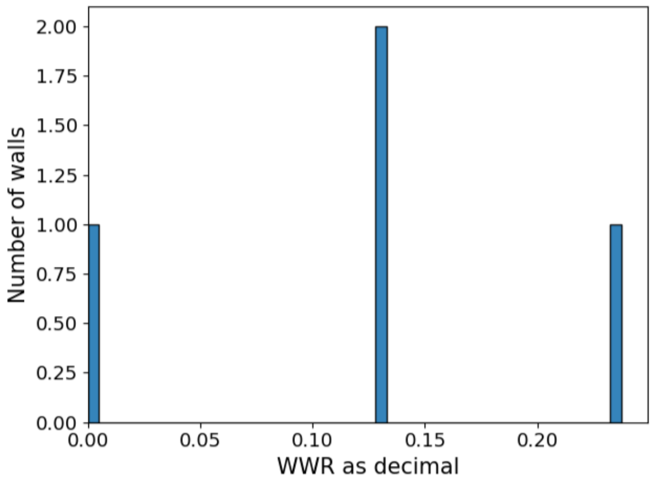

Figure 4. Window-to-wall ratio values (calculate_potential_index.ipynb).

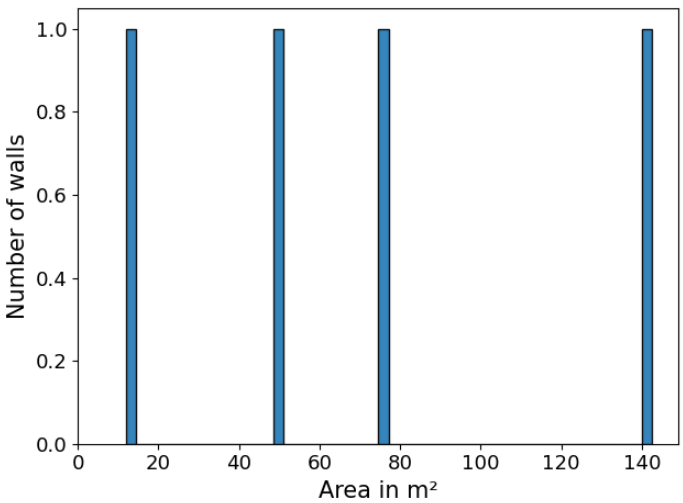

Figure 5. Surface area values (Regenerated plot).

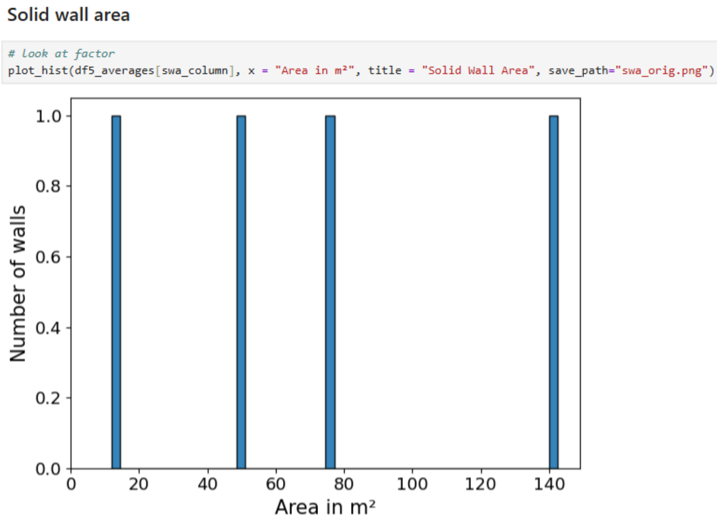

Figure 6. Surface area values (calculate_potential_index.ipynb).

By combining solid wall area, window-to-wall ratio, and orientation values, index values for walls were constructed (Figure 7).

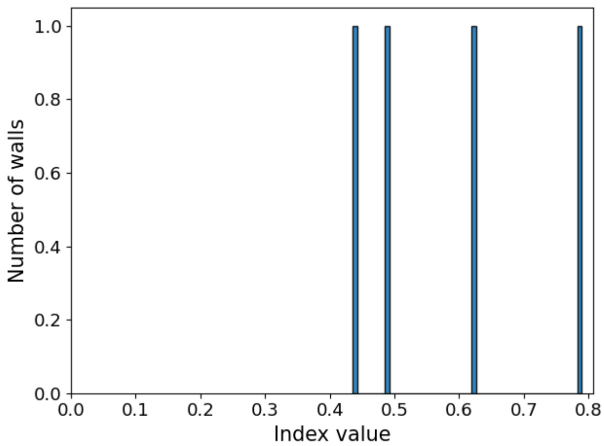

Figure 7. The calculated index for four walls in the test area (Regenerated plot).

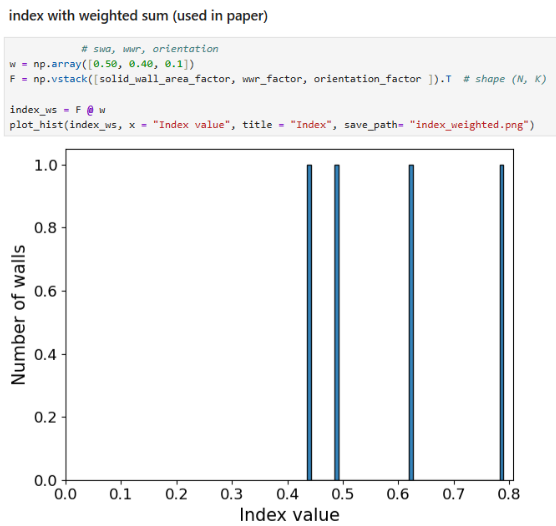

Figure 8. The calculated index for four walls in the test area (calculate_potential_index.ipynb).

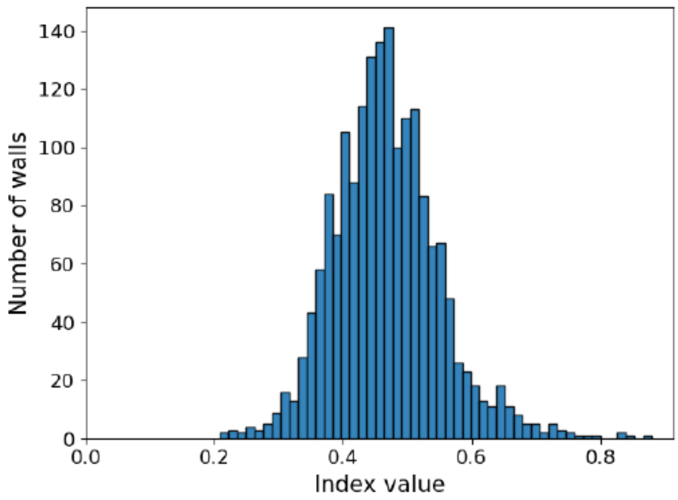

Figure 9. The calculated index for all walls in the test area (Original plot in paper).

Figure 7 represents the output of the reproduced pipeline based on the provided sample dataset (four walls). As the corresponding figure in the paper (Figure 9) is based on the full dataset, a direct comparison is not possible. Therefore, the reproduced results demonstrate correct execution of the workflow rather than replication of the published results.

Communication with the author

The author was helpful and responsive during the reproducibility assessment. After identifying an issue in the preprocessing script, the author promptly corrected the problem and updated the repository. Also, the author provided additional intermediate example data and detection outputs, which enabled verification of the downstream computational steps.

Remaining limitations

Full end-to-end reproduction of the workflow from raw inputs was not possible because several components depended on non-public resources. In particular, the street-view and oblique image acquisition workflow, proprietary imagery datasets, and trained object-detection models used for façade element detection are not publicly available. The workflow also relies on API-based data acquisition, which requires restricted credentials. Scripts for executing with user-provided credentials are not available in the repository, thereby limiting the reproducibility of the data acquisition step. Therefore, I was unable to reproduce the entire pipeline independently of the raw inputs.

Recommendations

Reproducibility could be further enhanced in the following ways:

- Provide software specifications such as requirements.txt or environment.yml file.
- Include a minimal runnable example dataset covering all publicly executable stages of the workflow.

- Archive a fixed version of the repository using a persistent identifier (e.g., Zenodo DOI) to ensure long-term access to the version used for the publication.
- Add an explicit software license to clarify reuse permissions.

Reproducibility assessment

Result: Partially reproducible.

I was able to run the main computational parts of the workflow using the provided repository and additional example data. However, complete independent reproduction from raw inputs is limited due to proprietary data sources and unavailable trained models.

Citing this document

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF/ResearchEquals at https://doi.org/10.17605/OSF.IO/HETV8

To cite the report use

Choi, J. (2026, June 11). Reproducibility review of: Maximizing Data Coverage: Fusing Street View and Oblique Imagery to Quantify Vertical Greenery Potential in Urban Areas. https://doi.org/10.17605/OSF.IO/HETV8