3 Branches 0 Tags Go to file Add file Code

rbnyng Merge pull request #31 from u…

874f13f · 3 days ago

img	update readme…	last month
interactive	Adapt to use g…	last month
tesserac…	Adapt to use g…	last month
.gitignore	ignoring __pyc…	2 months ago
LICENSE	Create LICENSE	2 months ago
READM…	Update READ…	3 days ago
app.ipynb	Adapt to use g…	last month
config.y…	change to no …	last month
labels.json	add an exampl…	2 months ago
pyprojec…	Adapt to use g…	last month
require…	Adapt to use g…	last month
viz.ipynb	Updated the in…	2 months ago

About

Interactive land cover classification of TESSERA embeddings

Readme
MIT license
Activity
Custom properties
19 stars
1 watching
2 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 5





README MIT license

- **Python** 93.6%
- **Jupyter Notebook** 6.4%

Interactive Tessera Embedding Classifier

This repository contains a Jupyter notebook based tool for interactive, human-in-the-loop classification of geospatial data using the Tessera foundation model embeddings.

The tool allows a user to define an area of interest, visualize the high-dimensional embedding data with PCA, and iteratively train a machine learning model by simply clicking on the map to label.

Features

- **Interactive Map Interface**: Pan and zoom on a satellite or terrain basemap.
- **Data-Driven Visualization**: Uses PCA to create a false-color visualization of Tessera embeddings.
- **Point-and-Click Training**: Simply click on the map to add labeled training points.
- **Custom Classes & Colors**: Dynamically add new classes and customize their colors with a color picker.
- **Live Classification**: Train a k-Nearest Neighbors model and classify the map with a click.
- **Iterative Refinement**: Add more pins to correct mistakes and re-run the classification for immediate (relatively) feedback.

Prerequisites

Before installing and running the tool, make sure you have:

- Python – install from python.org (v13 tested)
- Git – Install from git-scm.com

Installation

1. **Clone the Repository**:

```
git clone https://github.com/ucam-eo/tessera-interactive-map.git
cd tessera-interactive-map
```

2. **Create and Activate a Virtual Environment**:

```
python -m venv venv
source venv/bin/activate
```

3. **Install Dependencies**: The required packages are listed in `requirements.txt`.

```
pip install -r requirements.txt
```

For jupyter lab, also run `python -m ipykernel install --user --name=venv` to register the kernel

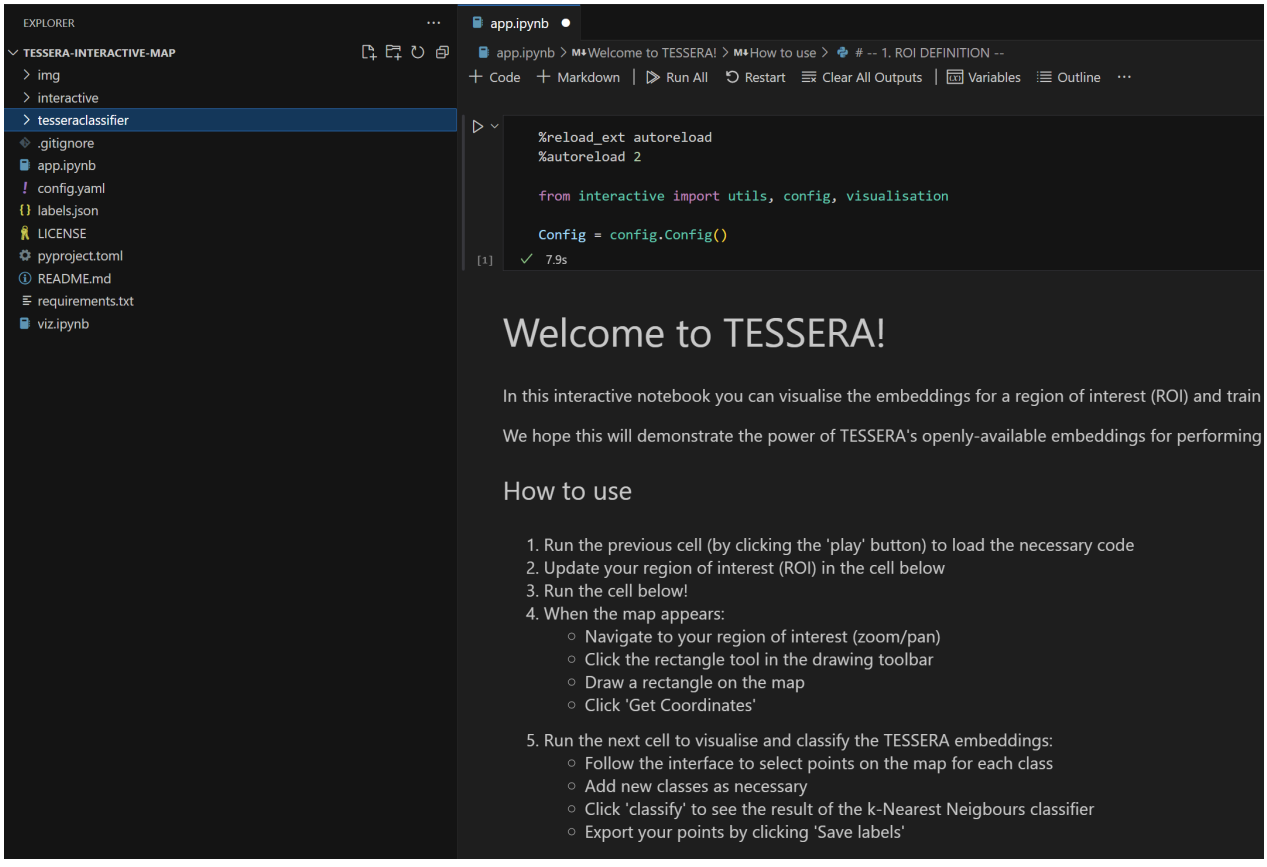
How to Use the Tool

1. Open the Project

Open the cloned repository folder in VS Code or Jupyter Notebook, or whatever else that runs ipynb files.

2. Launch the Notebook

Using the file browser on the left, find and open the `app.ipynb` file.



3. Initialize the Environment

Once the notebook opens, you're ready to begin. Run the top cell to load the required packages and set up the environment.

4. Set the Region of Interest

Run the ROI selector cell. Define your bounding box with the tool by dragging a square.

```
# -- 1. ROI DEFINITION --

# Create and display the bounding box selector
bbox_selector = visualisation.BoundingBoxSelector()
bbox_selector.display()
```

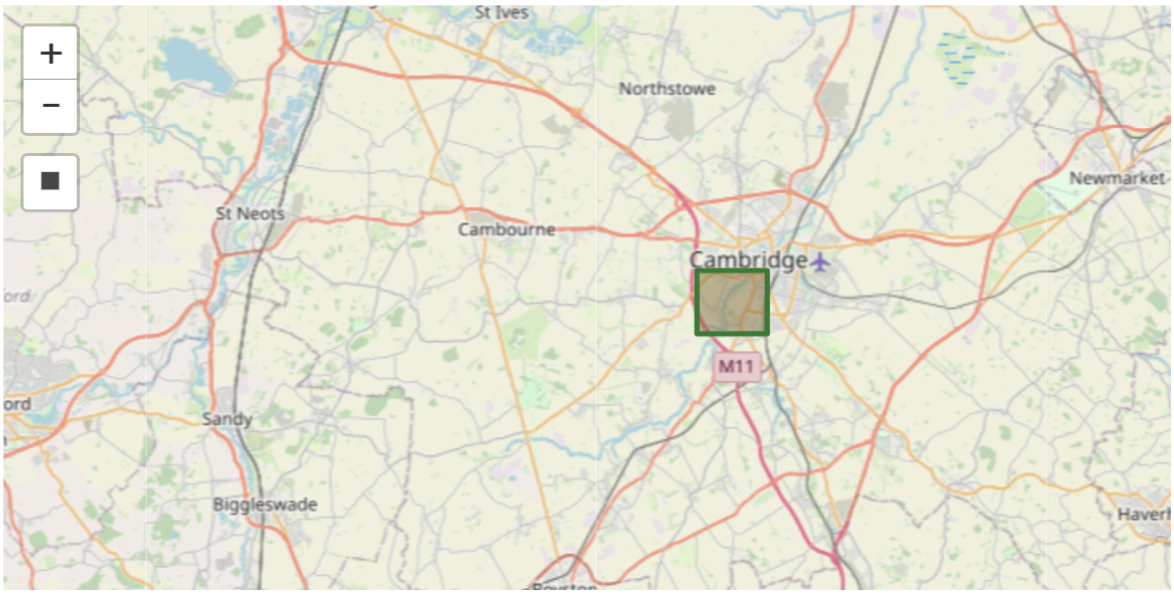
[2] ✓ 0.0s

✓ Bounding Box Selected

Longitude: 0.0782 to 0.1331

Latitude: 52.1701 to 52.2004

Draw a new rectangle to replace this selection



5. Generate the Interactive Map

Click Run to execute the cell below the ROI selector in the notebook. This may take a couple of minutes.

```
bbox_tuple = bbox_selector.get_bbox()
if bbox_tuple:
    (MIN_LAT, MAX_LAT), (MIN_LON, MAX_LON) = bbox_tuple
else:
    raise ValueError("No bounding box selected")

# -- 2. FETCH AND MOSAIC RELEVANT TESSERA TILES --

embedding_mosaic, mosaic_transform = utils.TesseraUtils().process_roi_to_mosaic(
    lat_coords=(MIN_LAT, MAX_LAT),
    lon_coords=(MIN_LON, MAX_LON),
)

# -- 3. VISUALISE MAP, PLACE TRAINING POINTS, & RUN CLASSIFICATION --

mapping_tool = visualisation.InteractiveMappingTool(
    MIN_LAT,
    MAX_LAT,
    MIN_LON,
    MAX_LON,
    embedding_mosaic,
    mosaic_transform,
)
mapping_tool.display()
```

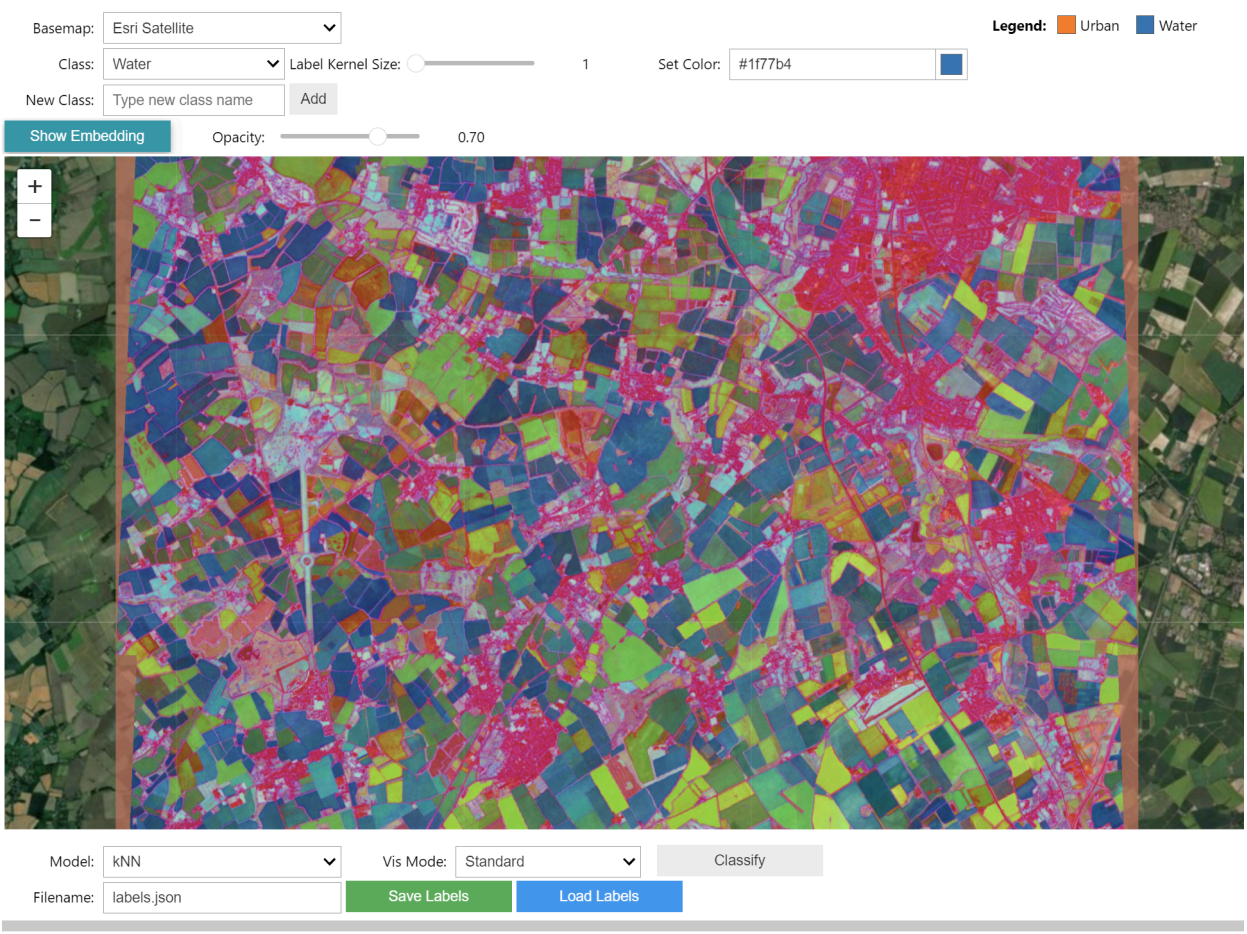
[3] ✓ 6.8s

```
Bounding box defined:
└ (52.17°, 0.08°) ┐ (52.20°, 0.13°)

Searching for tiles in ROI: (0.078223, 52.170067, 0.133145, 52.200378) for year 2024
ROI intersects with 1 registry block(s). Loading them...
Required registry blocks loaded.

Found 1 tiles to merge.
```

Processing tiles: 100% 1/1 [00:05<00:00, 5.37s/it]

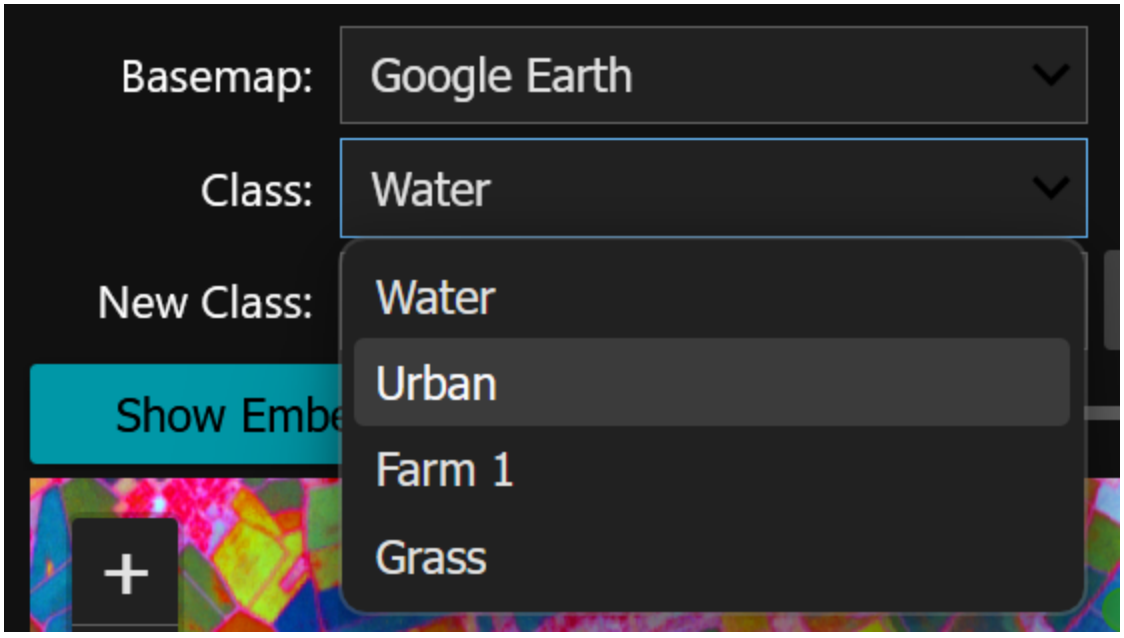
6. Wait for the Map to Load

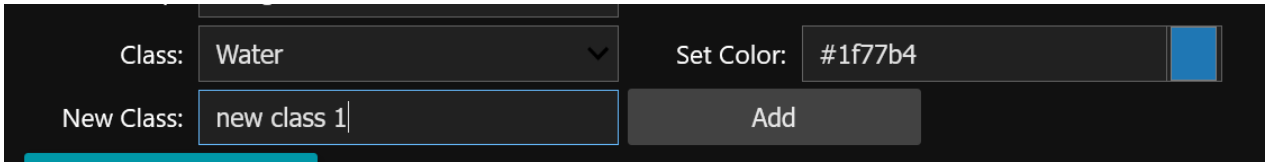
Once processing completes, a map will appear in the notebook.



7. Select or Add a Class

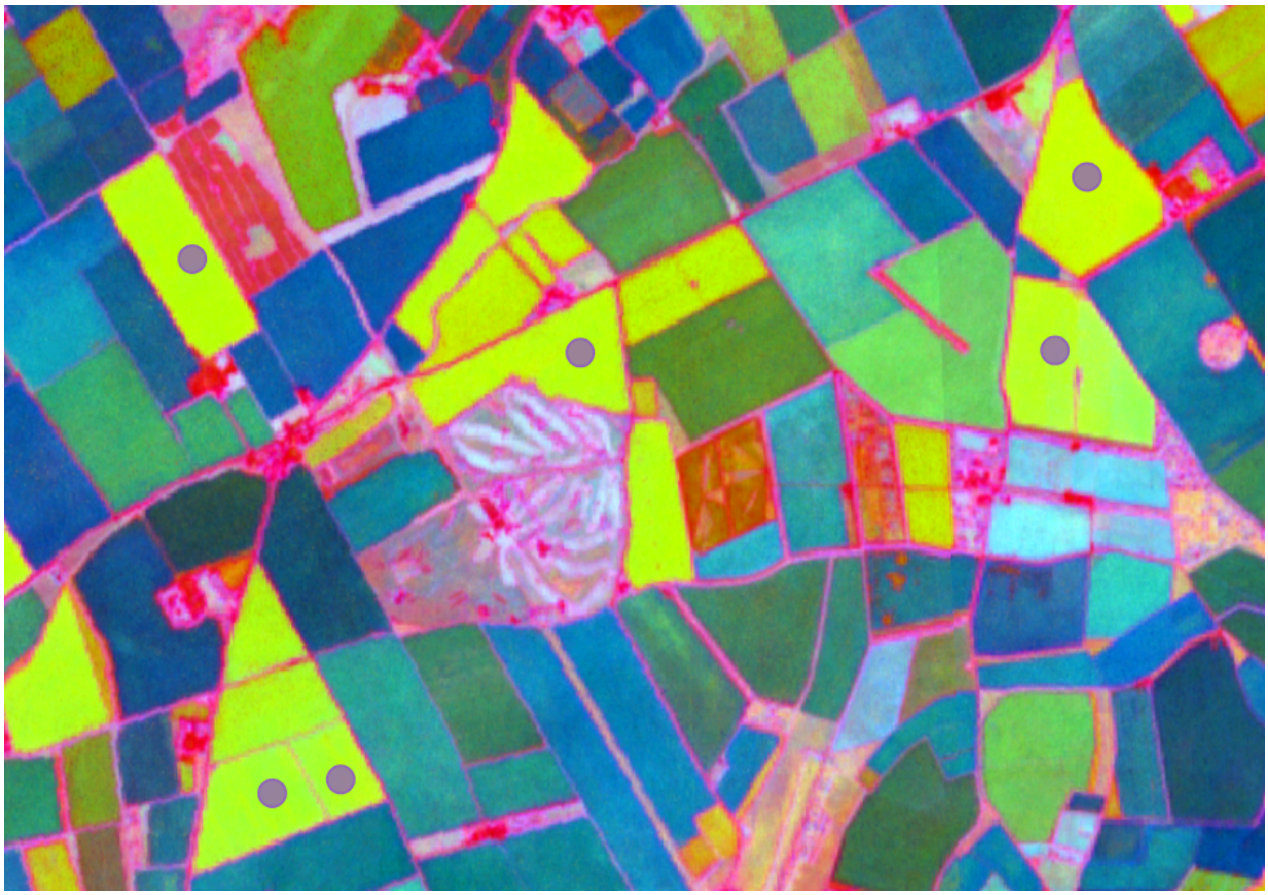
Use the class selector to pick an existing class or create a new one. You can customize the color using the color palette.





8. Label the Map

Click directly on the map to place pins for each class. Clicking a pin again will remove it.



9. Start Classification

Once you've added pins from **at least two classes**, click the **Classify** button to begin the classification.

Model: Random Forest Vis Mode: Confidence (Opacity) Classify