



Type '/' to search projects

[Help](#)[Docs](#)[Sponsors](#)[Log in](#)[Register](#)

spectraforce 0.1.7



Latest version

`pip install spectraforce==0.1.7` 

Released: Feb 18, 2026

Premium desktop workstation for Earth observation segmentation.

Navigation

 [Project description](#) [Release history](#) [Download files](#)

Verified details

These details have been
[verified by PyPI](#)

Maintainers

[ArnabPython](#)

Unverified details

Project description

SpectraForge

A premium desktop workstation for **Earth observation processing and visualization** across sensors. Unsupervised segmentation and uncertainty are optional modules inside a much broader EO workflow. Built to run locally on Windows/macOS/Linux.

Why it exists

- **Multi-sensor preprocessing** (Sentinel-1/2/3, Landsat, ERA5, ENMAP, hyperspectral, PlanetScope, UAV)
- **Band management and index generation** for fast EO analysis
- **Interactive ROI labeling** for ground-truthing and validation
- **Optional unsupervised segmentation** with probability maps and uncertainty
- **Designed for patent-ready workflows** (keep private until filing)

Features (v1)

These details have **not** been
verified by PyPI

Meta

- **License**

Expression:

LicenseRef-
Proprietary

[SPDX](#) [↗](#) [License](#)

[Expression](#) [↗](#)

- **Author:**

SpectraForge

- **Requires:** Python
≥3.10



Bloomberg is a
Visionary sponsor
of the Python
Software
Foundation.

[PSF Sponsor](#) · [Served
ethically](#)

Report project as
malware

- Clean, modern studio UI (not a QGIS clone)
- GeoTIFF, NetCDF, ENVI (hyperspectral), Sentinel-2 folder support
- Sensor presets (Sentinel-1/2/3, Landsat, PlanetScope, UAV, ERA5, ENMAP, hyperspectral)
- Auto feature selection across multi-band data
- Unsupervised segmentation with probability maps and entropy-based uncertainty (optional)
- Index Builder with curated indices per sensor + custom JSON/YAML recipes
- ROI selection + cluster labeling workflow
- Run history saved to `runs/`
- Copernicus + Planet API panels (offline stub in this build)

Install

From PyPI:

```
python -m pip install spectraforce
```

From source:

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

Run

```
spectraforce
```

Alternative:

```
python -m spectraforce
```

Data formats

- **GeoTIFF:** `.tif`, `.tiff`
- **NetCDF:** `.nc` (ERA5 and other gridded products)

- **ENVI:** `.hdr` + `.img` (hyperspectral)
- **Sentinel-2:** SAFE folder with `.jp2` band files
- **NumPy:** `.npy` arrays (2D or 3D)
- **Tabular:** `.csv`, `.xlsx`

Indices

SpectraForge ships with curated indices (NDVI, NDWI, etc.) per sensor. You can add your own:

- JSON/YAML recipe files (see `samples/` for examples)
- Index layers appear in the layer stack like any other band

Example custom indices file: `samples/indices_example.json`

NPY export

Export data to `.npy` directly from the UI:

- All bands in one file
- Selected bands in one file
- Individual band files
- Optional index layer exports

Offline samples

Synthetic samples live in `samples/` so the repo runs without downloads. Real cropped samples live in `samples/real/`. Manipur example outputs live in `samples/manipur/`.

API keys (stored locally)

API keys are stored in `~/.spectraforce/config.json` on your machine.

Connectors (step by step)

The **Connectors** tab provides a guided workflow for sensor platform access (Copernicus and PlanetScope). This build ships with an offline

stub so the UI and local key storage can be tested without network access. The steps below describe the full workflow, and the current offline behavior is noted where relevant.

What the Connectors tab does

- Collects and stores API keys locally
- Lets a dataset browser query sensors (S1/S2/S3 and PlanetScope)
- Builds a download queue with a save location

Step-by-step workflow

1. Open the **Connectors** tab in the right-hand inspector.
2. Choose a provider in the left panel.
3. Paste the API key and click **Save**.
4. If a **Test** or **Validate** button is shown, click it.
5. Open the **Dataset Browser** section.
6. Pick a sensor family (for example: Sentinel-2).
7. Set a bounding box or choose a saved ROI.
8. Set a date range and cloud filter.
9. Click **Search** to list scenes.
10. Select one or more scenes and click **Add to Queue**.
11. Set the **Save Location**.
12. Click **Start Download**.

Example

1. Provider: Copernicus.
2. Sensor: Sentinel-2.
3. AOI: a polygon ROI drawn in the map.
4. Date range: 2025-01-01 to 2025-02-01.
5. Cloud filter: 0–20%.
6. Action: search, add two scenes to queue, save to `samples/downloads/`.
7. Result: queued items appear in the download list with size, sensor, and target path.

Offline stub behavior in this build

- Keys are saved locally and shown as “stored”.
- Searches and downloads are disabled to keep the build offline.

- The UI still displays the full workflow so it can be demonstrated end-to-end.

Why SpectraForge Helps

SpectraForge provides a fast, repeatable path from raw EO images to clean indices, analysis layers, and exportable outputs. Custom scripts do not need to be rebuilt for every dataset. The same workflow can be applied across sensors for consistent results. A typical workflow can be done by loading a folder, letting the app auto detect bands, choosing indices, and optionally running unsupervised segmentation with ROI labeling.

UI Demo (Auto NPY Detection)

When a `.npy` dataset is opened, SpectraForge can automatically detect the sensor, load the bands, and populate the band list. If the sensor cannot be detected, it can be selected manually and the workflow can continue without restarting.



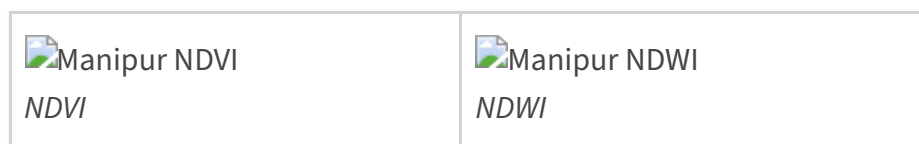
Step by Step Outputs for Manipur Sentinel-2

The full pipeline was run on real Sentinel-2 data from Manipur, and the outputs were saved below.

Step 1 — Load bands and true color



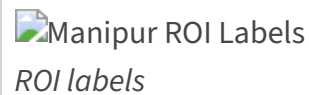
Step 2 — NDVI and NDWI indices



Step 3 — Unsupervised segmentation with 8 clusters



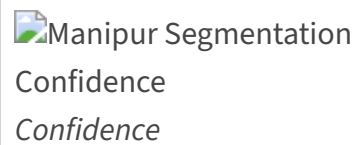
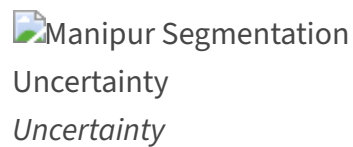
Step 4 — ROI selection and labeling



Step 5 — Uncertainty and confidence The same color rule is used for both maps: blue means low, red means high.

For uncertainty, blue means low uncertainty and red means high uncertainty.

For confidence, blue means low confidence and red means high confidence.



Saved outputs full resolution

- `samples/manipur_full/manipur_full_stack.npy`
- `samples/manipur_full/manipur_full_stack_preview.tif`
- `samples/manipur_full/manipur_full_<index>.npy`
- `samples/manipur_full/manipur_full_<index>.tif`

Color Legend for segmentation labels

Clusters are **unsupervised**. Colors map to cluster IDs in order:

- 0 → maroon #800000
- 1 → darkblue #00008B
- 2 → darkgreen #006400
- 3 → cyan #00FFFF
- 4 → darkcyan #008B8B
- 5 → magenta #FF00FF
- 6 → indigo #4B0082
- 7 → grey #808080
- 8 → peru #CD853F
- 9 → slateblue #6A5ACD
- 10 → mediumspringgreen #00FA9A
- 11 → orangered #FF4500

If you run fewer than 12 clusters, only the first N colors are used.

Color Notes for indices

Index quicklooks use a **viridis** scale: brighter colors indicate higher values.

Uncertainty calibration made easy

- Run segmentation → get probability maps (`predict_proba`)
- Use entropy + confidence to visualize uncertain regions
- Assign labels with ROI selections (no labeled data required)

Segmentation engine

If your environment has scientific stack conflicts, switch the engine to **Safe mode** in the UI.
Fast mode uses scikit-learn when available.

SpectraForge vs QGIS (Pros & Cons)

Aspect	SpectraForge	QGIS
Focus	EO segmentation + indices + uncertainty	Full GIS for all domains
Setup	One command local run	Heavier install + plugins
Unsupervised segmentation + uncertainty	Built-in, turnkey	Requires plugins/workflows
Indices	Curated EO indices + custom recipes	Many tools, but more manual setup
UI style	Modern studio layout (not QGIS style)	Traditional GIS layout

Aspect	SpectraForge	QGIS
Extensibility	Focused feature set	Huge plugin ecosystem
Geoprocessing breadth	Focused EO analytics	Broad GIS toolbox
Best for	Fast EO segmentation + research demos	Full GIS analysis & cartography

Pros of SpectraForge: fast EO-first workflow, built-in uncertainty, simple NPY export, easy to demo.

Cons vs QGIS: fewer GIS tools, smaller plugin ecosystem, less advanced cartography.

Contributions

See CONTRIBUTING.md and CODE_OF_CONDUCT.md.

Credits

Arnab Bhowmik

How to cite

If you use SpectraForge in academic work, please cite it like this:

Arnab Bhowmik. SpectraForge: Earth Observation Process-

BibTeX:

```
@software{spectraforge_2026,  
  author = {Bhowmik, Arnab},  
  title = {SpectraForge: Earth Observation Processing and  
  year = {2026},  
  version = {0.1.7},
```

```
url = {https://github.com/ArnaBannonymus/SpectraForge
}
```

Privacy note

Runs locally. No data leaves your machine.



License

Proprietary (permission required for any use)



Help

- Installing packages
- Uploading packages
- User guide
- Project name retention
- FAQs

About PyPI

- PyPI Blog
- Infrastructure dashboard
- Statistics
- Logos & trademarks
- Our sponsors

Contributing to PyPI

- Bugs and feedback
- Contribute on GitHub
- Translate PyPI
- Sponsor PyPI
- Development credits

Using PyPI

- Terms of Service
- Report security issue
- Code of conduct
- Privacy Notice
- Acceptable Use Policy

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

"PyPI", "Python Package Index", and the blocks logos are registered trademarks of the Python Software Foundation [↗](#).

© 2026 Python Software Foundation [↗](#)
[Site map](#)

[Switch to desktop version](#)

[> English](#) [español](#) [français](#) [日本語](#) [português \(Brasil\)](#) [українська](#) [Ελληνικά](#) [Deutsch](#) [中文 \(简体\)](#)
[中文 \(繁體\)](#) [русский](#) [עברית](#) [Esperanto](#) [한국어](#)