Machine Learning Model Specification for Cataloging Spatio-Temporal Models

Making Geospatial Models Searchable and Usable

Francis Charette-Migneault, **Ryan Avery**, Brian Pondi, Joses Omojola, Simone Vaccari, Parham Membari, Devis Peressutti, Jia Yu, and Jed Sundwall. 2024. "*Machine Learning Model Specification for Cataloging Spatio-Temporal Models (Demo Paper)*". In 3rd ACM SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data (GeoSearch'24), October 29–November 1 2024, Atlanta, GA, USA. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681769.3698586

PAPER

CRIM Wherobots TERRADUE STAC

Challenge of Cataloging Spatio-Temporal Models

Discovering Assets

SpatioTemporal Asset Catalog (STAC) has a robust ecosystem for describing, querying, and fetching geospatial data.

More and More Models

Lots of models are being trained on different kinds of geospatial data to solve various problems.

Problem Statement

These models are being published without the necessary information to find and run them on the right data.

Challenges deep-dive

Data dependencies and constraints

Models trained on geospatial data are typically constrained to a sensor domain, a geographic domain and temporal domain.

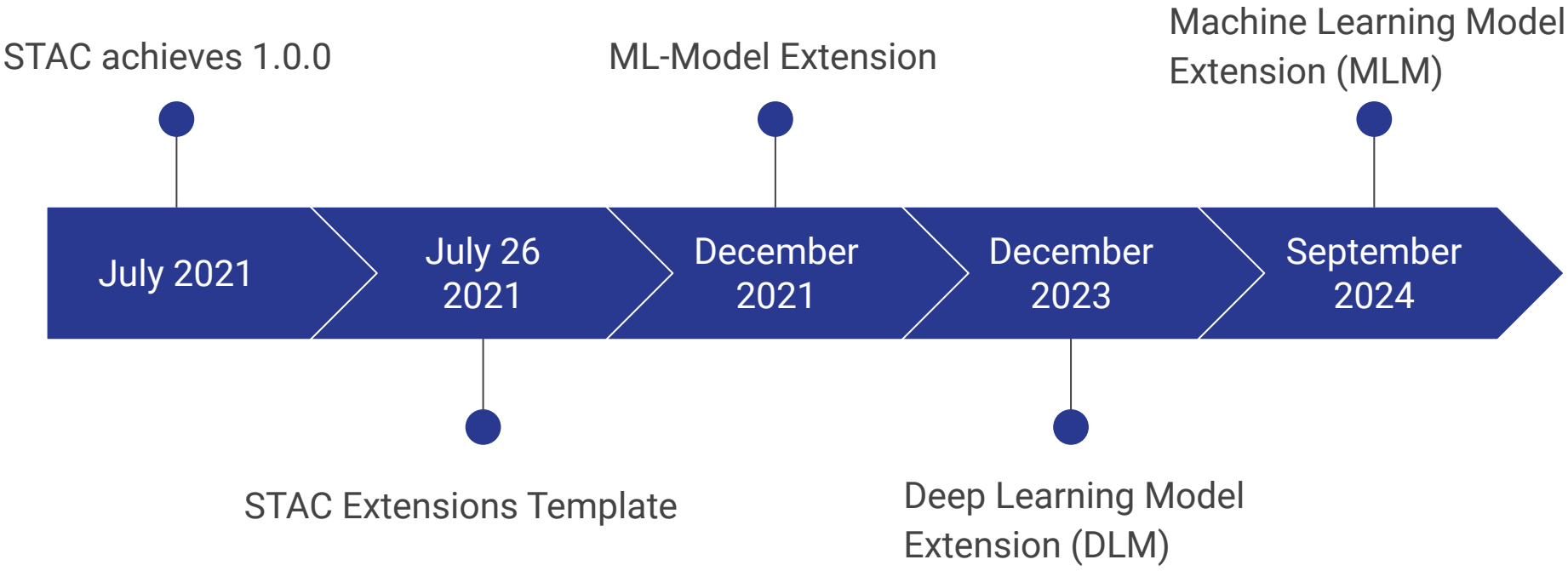
Complex input requirements

Models have different requirements for the input shape, data type, and data preparation steps to reproduce model inference.

Complex runtime requirements

Models have diverse dependencies on higher level language libraries, lower level libraries, and hardware-specific tooling.

Timeline of ML standards with STAC



From 2021 - 2024, 76 STAC Extensions are indexed at https://stac-extensions.github.io/

Solution

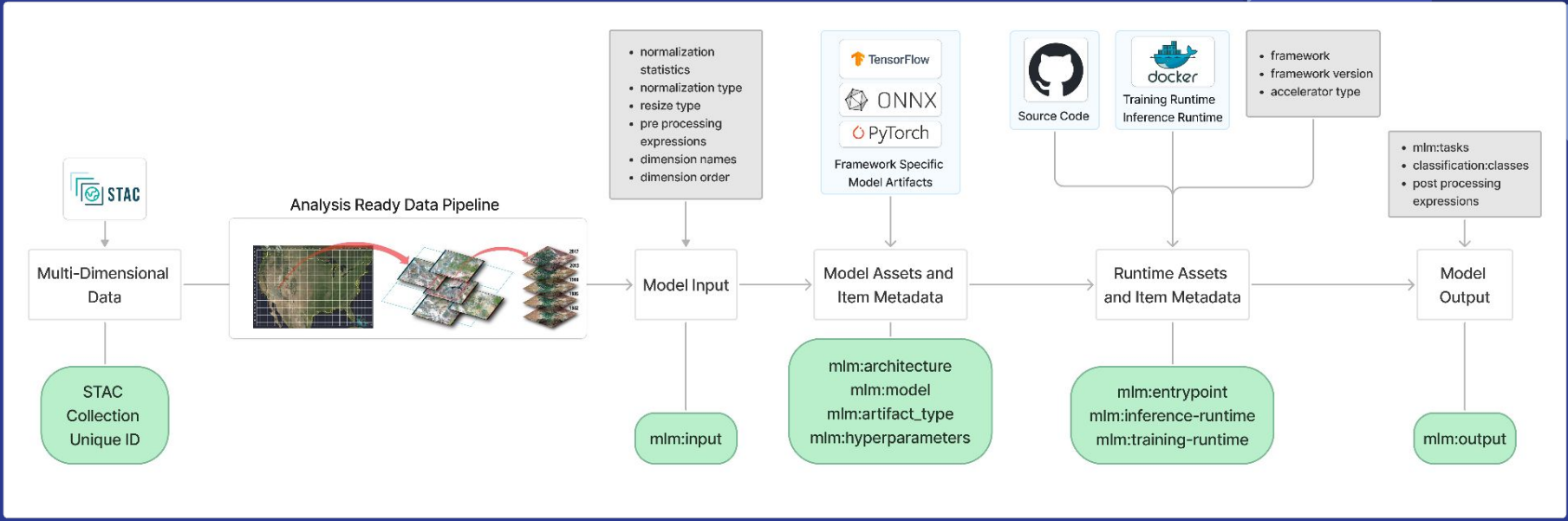
A STAC Extension for Machine Learning Models

stac-extensions/mlm

A Model Card for Geospatial

- Interoperable with STAC Core and Extension schemas
- ML Framework agnostic
- Multiple implementation examples from different organizations

MLM Features



Sets norms for geospatial and temporal ML model publishing

Relates models back to geospatial collections (training/validation/test datasets)

Objective: make searching, finding and reusing geospatial models near-zero effort

MLM Specification Structure

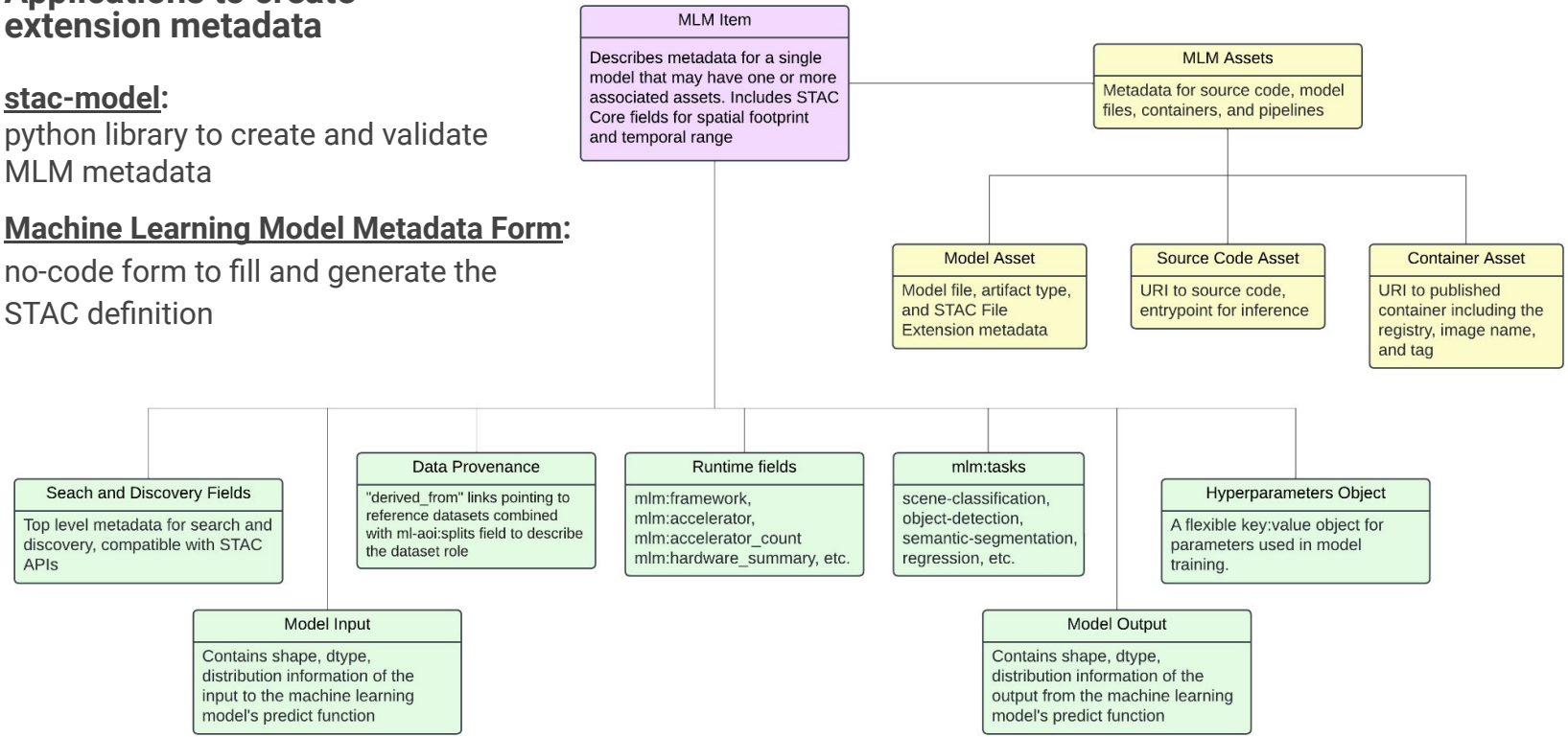
stac-model

MLM Form

Applications to create extension metadata

stac-model:
python library to create and validate MLM metadata

Machine Learning Model Metadata Form:
no-code form to fill and generate the STAC definition



STAC Extensions + MLM

Related STAC extensions	
Label Extension	Classes (name/index only), Annotations, ML Tasks, Methods
ML-AOI Extension	Area of Interest, Splits (Train, Validate, Test)
Classification	Classes (detailed) for supervised ML models
Processing	Level of processing (L4 = Model Output), Data Lineage, Expression/Software
Raster / EO / STAC 1.1 Bands	Relevant for which Bands to use as Model input
Stats	Statistics about Item counts, relevant for reference datasets and collections
Scientific	Reference scientific work, Publication Paper
File & Storage	Checksum and location of Model weights / checkpoints
Version	Model revisions (e.g.: experiment runs)

STAC MLM Example

STAC and Extension Metadata

```
{
  "stac_version": "1.0.0",
  "stac_extensions": [
    "https://stac-extensions.github.io/mlm/v1.3.0/schema.json",
    "https://stac-extensions.github.io/raster/v1.1.0/schema.json",
    "https://stac-extensions.github.io/file/v1.0.0/schema.json",
    "https://stac-extensions.github.io/ml-aoi/v0.2.0/schema.json"
  ],
```

Spatio-Temporal Metadata

```
  "type": "Feature",  "id": "model-multi-input",  "collection": "ml-model-examples",
  "geometry": {
    "type": "Polygon",
    "coordinates": [ [
      [ -7.882190080512502, 37.13739173208318 ], [ -7.882190080512502, 58.21798141355221 ],
      [ 27.911651652899925, 58.21798141355221 ], [ 27.911651652899925, 37.13739173208318 ], [ -7.882190080512502, 37.13739173208318 ]
    ] ]
  },
  "bbox": [ -7.882190080512502, 37.13739173208318, 27.911651652899925, 58.21798141355221 ],
  "properties": {
    "description": "Example model that employs multiple input sources with different combination of bands and some inputs without any band at all.",
    "datetime": null, "start_datetime": "1900-01-01T00:00:00Z", "end_datetime": "9999-12-31T23:59:59Z",
```

Model Metadata

```
    "mlm:name": "Resnet-18 Sentinel-2 ALL MOCO",
    "mlm:tasks": [ "classification", "semantic-segmentation" ],
    "mlm:architecture": "ResNet",
    "mlm:framework": "pytorch",
    "mlm:framework_version": "2.1.2+cu121",
    "file:size": 43000000,
    "mlm:memory_size": 1,
    "mlm:total_parameters": 11700000,
    "mlm:pretrained_source": "EuroSat Sentinel-2",
    "mlm:accelerator": "cuda",
    "mlm:accelerator_constrained": false,
    "mlm:accelerator_summary": "Unknown",
    "mlm:batch_size_suggestion": 256,
```

Inputs

```
    "mlm:input": [
      { "name": "RGB", "bands": [ "B04", "B03", "B02" ], "input": { "shape": [ -1, 3, 64, 64 ], "dim_order": [ "batch", "channel", "height", "width" ], "data_type": "uint16" },
        "norm_by_channel": false, "norm_type": null, "resize_type": null },
      { "name": "NDVI", "bands": [ "B04", "B08" ],  "input": { "shape": [ -1, 1, 64, 64 ], "dim_order": [ "batch", "ndvi", "height", "width" ], "data_type": "uint16" },
        "pre_processing_function": { "format": "gdal-calc", "expression": "(A - B) / (A + B)" }
      },
      { "name": "DEM", "description": "Digital elevation model. Comes from another source than the Sentinel bands. Therefore, no 'bands' associated to it.",
        "bands": [], "input": { "shape": [ -1, 1, 64, 64 ], "dim_order": [ "batch", "dem", "height", "width" ], "data_type": "float32" }
      }
    ],
```

Outputs

```
    "mlm:output": [
      { "name": "vegetation-segmentation", "tasks": [ "semantic-segmentation" ], "result": { "shape": [ -1, 1 ], "dim_order": [ "batch", "class" ], "data_type": "uint8" },
        "classification_classes": [
          { "value": 0, "name": "NON_VEGETATION", "description": "background pixels", "color_hint": null },
          { "value": 1, "name": "VEGETATION", "description": "pixels where vegetation was detected", "color_hint": [ 0, 255, 0 ] }
        ],
        "post_processing_function": null
      },
      {
        "name": "inverse-mask", "tasks": [ "semantic-segmentation" ], "result": { "shape": [ -1, 1 ], "dim_order": [ "batch", "class" ], "data_type": "uint8" },
        "classification_classes": [
          { "value": 0, "name": "NON_VEGETATION", "description": "background pixels", "color_hint": [ 255, 255, 255 ] },
          { "value": 1, "name": "VEGETATION", "description": "pixels where vegetation was detected", "color_hint": [ 0, 0, 0 ] }
        ],
        "post_processing_function": { "format": "gdal-calc", "expression": "logical_not(A)" }
      }
    ]
  }
}
```

https://github.com/stac-extensions/mlm/tree/main/examples

Model Weights and Input Bands

```
{
  "assets": {
    "weights": {
      "href": "https://huggingface.co/torchgeo/resnet50_sentinel2_rgb_moco/blob/main/resnet50_sentinel2_rgb_moco.pth",
      "title": "Pytorch weights checkpoint",
      "description": "A Resnet-50 classification model trained on Sentinel-2 RGB imagery with torchgeo.",
      "type": "application/octet-stream; application=pytorch",
      "roles": [
        "mlm:model",
        "mlm:weights"
      ],
      "raster:bands": [
        {
          "name": "B02 - blue", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B03 - green", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B04 - red", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B08 - nir", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        }
      ]
    }
  },
```

Reference Dataset (training data)

```
  "links": [
    {
      "rel": "derived_from",
      "href": "https://earth-search.aws.element84.com/v1/collections/sentinel-2-l2a",
      "type": "application/json",
      "ml-aoi:split": "train"
    }
  ]
}
```

Discussion

- **Initial focus of MLM is to support FAIR principles**
  - Findable
  - Accessible
  - Interoperable
  - Reusable

- **Adaptable to multiple applications, data formats, variable input/output structures**

- **Leverages existing STAC extensions to maximize reusability and cross-compatibility with existing datasets, source data retrieval, citing related work references, etc.**

- **Improves integrity, provenance tracking and trust involving ML processing pipelines and spatio-temporal data for decision making**

Contact

Francis Charette-Migneault
@fmigneault
francis.charette-migneault@crim.ca

Ryan Avery
@rybavery
ryan@wherobots.com

Annex: STAC MLM Example

STAC and Extension Metadata


```
{
"stac_version": "1.0.0",
"stac_extensions": [
  "https://stac-extensions.github.io/mlm/v1.3.0/schema.json",
  "https://stac-extensions.github.io/raster/v1.1.0/schema.json",
  "https://stac-extensions.github.io/file/v1.0.0/schema.json",
  "https://stac-extensions.github.io/ml-aoi/v0.2.0/schema.json"
],
```


Spatio-Temporal Metadata


```
"type": "Feature",  "id": "model-multi-input",  "collection": "ml-model-examples",
"geometry": {
  "type": "Polygon",
  "coordinates": [ [
    [ -7.882190080512502, 37.13739173208318 ], [ -7.882190080512502, 58.21798141355221 ],
    [ 27.911651652899925, 58.21798141355221 ], [ 27.911651652899925, 37.13739173208318 ], [ -7.882190080512502, 37.13739173208318 ]
  ] ]
},
"bbox": [ -7.882190080512502, 37.13739173208318, 27.911651652899925, 58.21798141355221 ],
"properties": {
  "description": "Example model that employs multiple input sources with different combination of bands and some inputs without any band at all.",
  "datetime": null, "start_datetime": "1900-01-01T00:00:00Z", "end_datetime": "9999-12-31T23:59:59Z",
```


Model Metadata


```
  "mlm:name": "Resnet-18 Sentinel-2 ALL MOCO",
  "mlm:tasks": [ "classification", "semantic-segmentation" ],
  "mlm:architecture": "ResNet",
  "mlm:framework": "pytorch",
  "mlm:framework_version": "2.1.2+cu121",
  "file:size": 43000000,
  "mlm:memory_size": 1,
  "mlm:total_parameters": 11700000,
  "mlm:pretrained_source": "EuroSat Sentinel-2",
  "mlm:accelerator": "cuda",
  "mlm:accelerator_constrained": false,
  "mlm:accelerator_summary": "Unknown",
  "mlm:batch_size_suggestion": 256,
```

Annex: STAC MLM Example

```
  "mlm:input": [
    { "name": "RGB", "bands": [ "B04", "B03", "B02" ], "input": { "shape": [ -1, 3, 64, 64 ], "dim_order": [ "batch", "channel", "height", "width" ], "data_type": "uint16" },
      "norm_by_channel": false, "norm_type": null, "resize_type": null },
    { "name": "NDVI", "bands": [ "B04", "B08" ],  "input": { "shape": [ -1, 1, 64, 64 ], "dim_order": [ "batch", "ndvi", "height", "width" ], "data_type": "uint16" },
      "pre_processing_function": { "format": "gdal-calc", "expression": "(A - B) / (A + B)" }
    },
    { "name": "DEM", "description": "Digital elevation model. Comes from another source than the Sentinel bands. Therefore, no 'bands' associated to it.",
      "bands": [], "input": { "shape": [ -1, 1, 64, 64 ], "dim_order": [ "batch", "dem", "height", "width" ], "data_type": "float32" }
    }
  ],
```

Inputs

```
  "mlm:output": [
    { "name": "vegetation-segmentation", "tasks": [ "semantic-segmentation" ], "result": { "shape": [ -1, 1 ], "dim_order": [ "batch", "class" ], "data_type": "uint8" },
      "classification_classes": [
        { "value": 0, "name": "NON_VEGETATION", "description": "background pixels", "color_hint": null },
        { "value": 1, "name": "VEGETATION", "description": "pixels where vegetation was detected", "color_hint": [ 0, 255, 0 ] }
      ],
      "post_processing_function": null
    },
    {
      "name": "inverse-mask", "tasks": [ "semantic-segmentation" ], "result": { "shape": [ -1, 1 ], "dim_order": [ "batch", "class" ], "data_type": "uint8" },
      "classification_classes": [
        { "value": 0, "name": "NON_VEGETATION", "description": "background pixels", "color_hint": [ 255, 255, 255 ] },
        { "value": 1, "name": "VEGETATION", "description": "pixels where vegetation was detected", "color_hint": [ 0, 0, 0 ] }
      ],
      "post_processing_function": { "format": "gdal-calc", "expression": "logical_not(A)" }
    }
  ]
}
```

Outputs

Annex: STAC MLM Example


```
{
  "assets": {
    "weights": {
      "href": "https://huggingface.co/torchgeo/resnet50_sentinel2_rgb_moco/blob/main/resnet50_sentinel2_rgb_moco.pth",
      "title": "Pytorch weights checkpoint",
      "description": "A Resnet-50 classification model trained on Sentinel-2 RGB imagery with torchgeo.",
      "type": "application/octet-stream; application=pytorch",
      "roles": [
        "mlm:model",
        "mlm:weights"
      ],
      "raster:bands": [
        {
          "name": "B02 - blue", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B03 - green", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B04 - red", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        },
        {
          "name": "B08 - nir", "nodata": 0, "data_type": "uint16", "bits_per_sample": 15,
          "spatial_resolution": 10, "scale": 0.0001, "offset": 0, "unit": "m"
        }
      ]
    }
  },
  "links": [
    {
      "rel": "derived_from",
      "href": "https://earth-search.aws.element84.com/v1/collections/sentinel-2-l2a",
      "type": "application/json",
      "ml-aoi:split": "train"
    }
  ]
}
```


Model Weights and Input Bands

Reference Dataset (training data)