# Hugin

A Machine Learning experimentation tool for Earth Observation

Marian Neagul, Ion Nedelcu, Gabriel Iuhasz

*{marian.neagul, teodora.selea, iuhasz.gabriel}@e-uvt.ro*

*ion.nedelcu@rosa.ro*

# Who are we?

- The **SAGE Group** at **West University of Timișoara** (UVT) and **Institute e-Austria** (IeAT), Romania
  - An group focusing on Machine Learning / Deep Learning
  - Distributed Computing – With an exhaustive experience in Cloud Computing, HPC and, formerly, GRID Computing
- This work is a **collaboration** between **UVT, ROSA and TerraSigna.**
- It is partially supported by:
  - The **ESA** EOSmith project
  - The **Romanian National PN3** 432PED Project

# Why HuginEO?

- **Lack** of tools **supporting geospatial** data
    - Most tools **ignore geospatial information** (projection, coordinates)
    - **Difficult** to support **data** that is **not RGB**
    - **Missing** support for **co-registration**
- We have to **focus** on **repetitive** actions related to our **data** instead on **focus** on **algorithms** and **models**
    - Usually we find ourselves **reinventing** the same **routine** for each of our endeavours
- Some tools support **partially** the required **features** but make it **difficult** to include our own pre(post-)**processing**

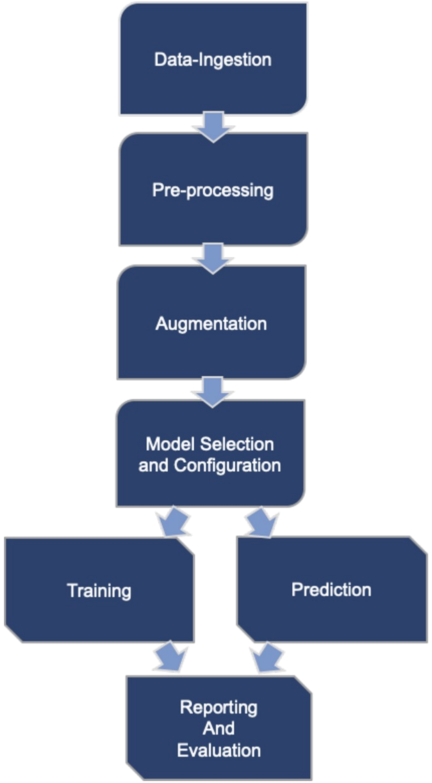
# What is Hugin?

- Our aim with **Hugin** is to provide a solution to easily handle **EO datasets** which can be used for a wide variety of common problems such as: **segmentation/classification**, **super resolution** and regression.
- Design goals can be summarized as such:
  - **Easy configuration** scheme which does **not require extensive ML background**.
  - **Simple data ingestion** mechanism which can **handle large datasets** for **out-of-core ML** solutions
  - Comprehensive **processing pipeline** which includes; pre-processing methods, a comprehensive set of state-of-the-art ML and DL algorithms, training, validation and reporting mechanisms. Pipeline extensible with your own routines
  - Provide an **easy method for predictive model instantiation** and the ability to export the resulting predictive models in a widely usable format.

# Core Functionality

- Support for:
  - TensorFlow/SciKit-Learn models
  - Models with multiple inputs/outputs
  - Client side datacubes (stackstac, xarray based)
  - Support for Zarr datasets

# What is Hugin?



- Developed in **Python** using **Keras**, **scikit-learn, rasterio, gejson, Fiona, geopandas.**
- **Features:**
  - Support for **various data sources**; Local, Google storage buckets, S3, STAC Catalogs
  - On the fly **channel input mapping** and **tiling**
    - Including support for **multiple inputs** of different sizes and sources (**co-registration**)
  - Support for **augmentation** including **morphological operators** on both input images and ground truth in the case of segmentation
  - **Easy addition** of user defined **custom components** such as; optimizers, activation functions, loss functions and last but not least user defined models.
  - **Easy usage and extension** using **callbacks** for pre and post **processing** as well as during **training**.
  - **YAML** based **configuration** aimed for supporting repeatable experiments
  - Other **features** include: Google **ML Engine Support**, **Multi-GPU** training

# Hugin Configuration

## Experiment Configuration

### Data Source Configuration

- Tilling or sub-setting / Stride
- Mapping of input data to tensors

### Model Configuration

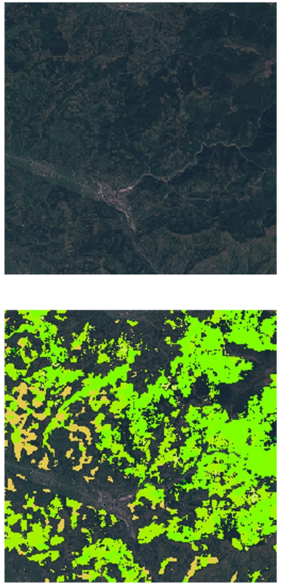
- Model Builder
- Early stopping
- Optimiser
- Metrics

### Output Configuration

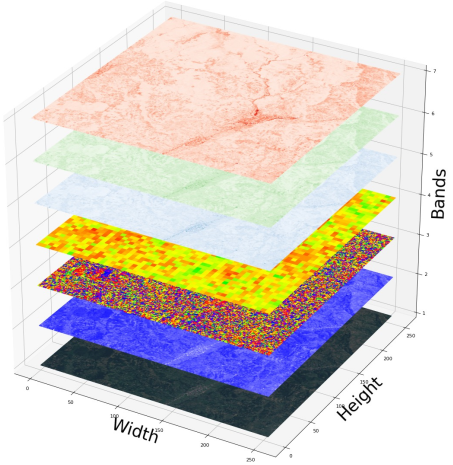
- Output format (Inference time)
- Ensemble Configuration (Inference time)

- Aims to **provide** a **reproducible** environment for **training** time experiments
- Easy to **generate automatically** from your own tools but also aimed to be **human editable**
- Used to specify **data layout** (filesystem based data loaders, S3 loaders, GCP Storage loaders)
- **Allows** you to **reference** your own **code** (callbacks, models, etc). Add **your** own **loss functions**
- Makes **integration** with external **hyper-optimization** tools more easy

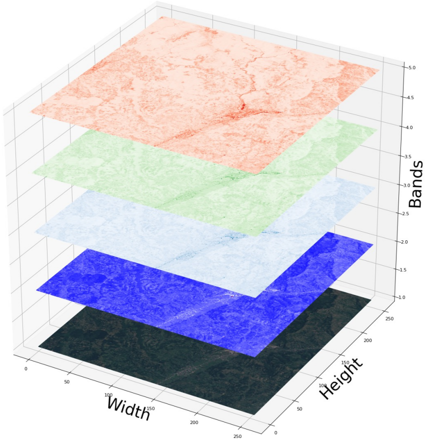
# Hugin - Data Ingestion and pre-processing



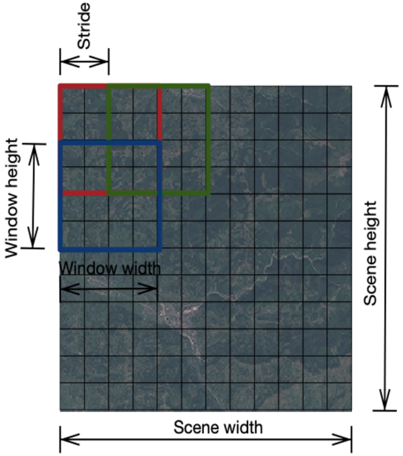
Input
Images and GTI
Data Fusion



Load bands
from Images

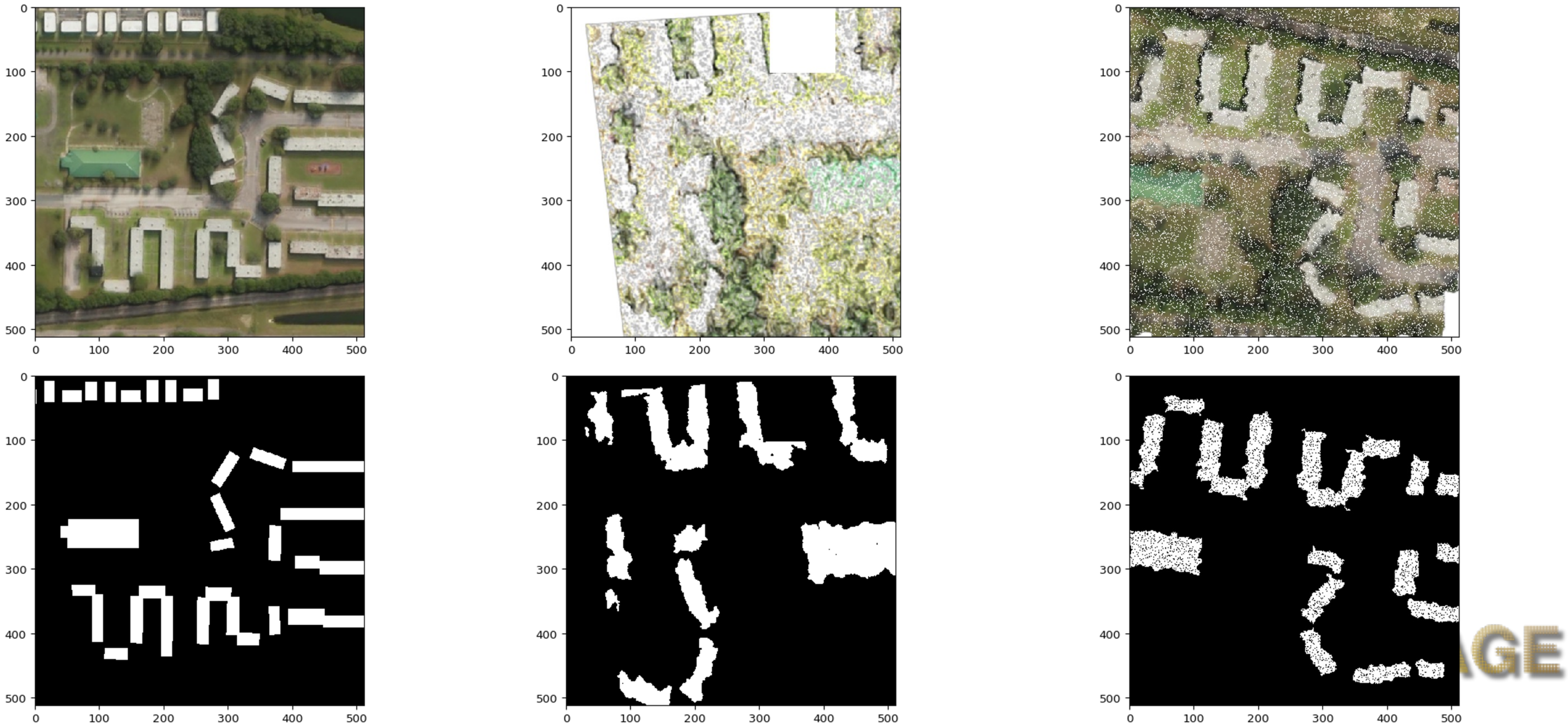


Dynamically
map bands to
input tensors

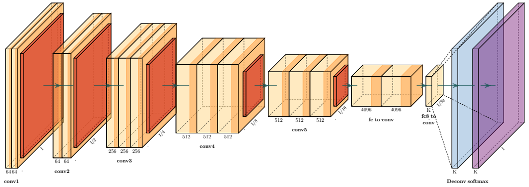


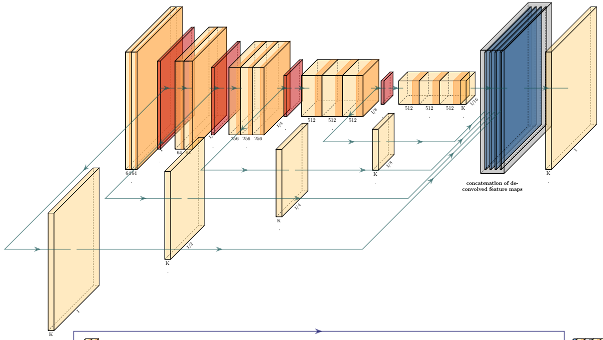
Dynamic tiling
of input tensor

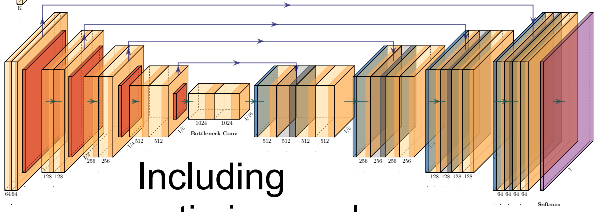
# Hugin - Augmentation



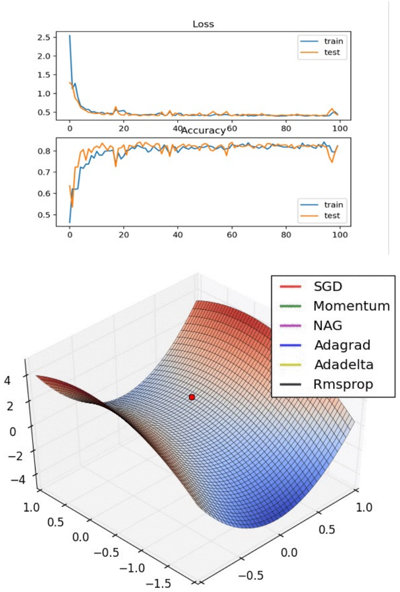
# Hugin - Training and Inference





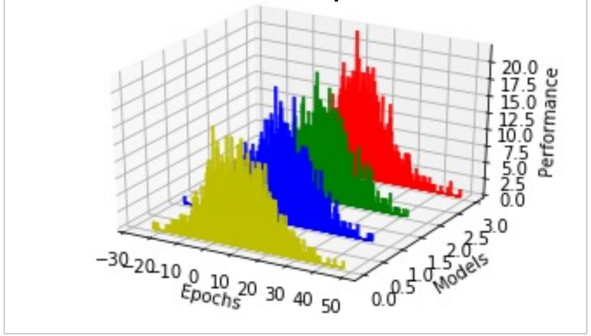


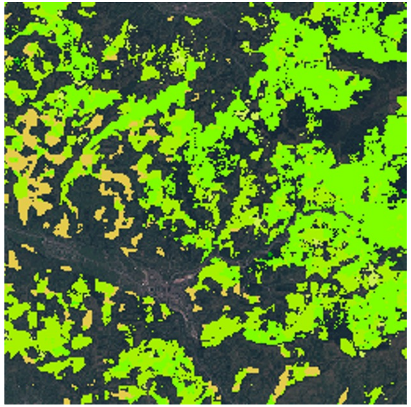
Including optimizer and callbacks



Training

Create ensamble predictions



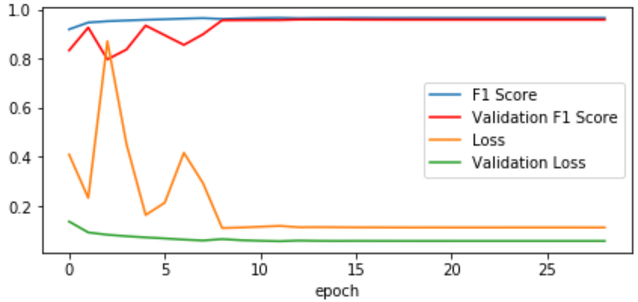


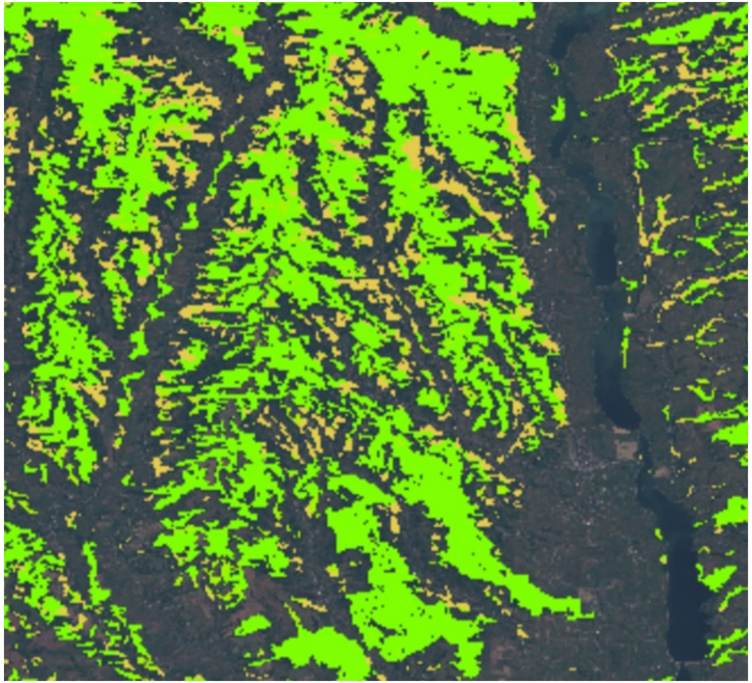
Inference Raster

SAGE

# Use cases of Hugin – Forestry (Playground)

- **Forestry segmentation** using S2 and **CORINE** data
  - Identify **forested** areas using an Deep Learning **model** (U-Net)
  - **Transfer learning** from SEN12MS (S2 only)
  - **Input**: RGB and NIR
  - **Output**: Class ID’s for each pixel
    - Rendered as GeoJSON





# Conclusions

- **Hugin** is **aimed** at making the uptake **easier** for state of the art **Machine Learning** technologies in the **EO community**

- We aim to **support** your **EO** needs by providing a **common** and **extensible** data **processing** pipeline suited **for Machine Learning applications**

- Hugin **facilitate data fusion** by **providing** the required facilities for **consuming** data in **different resolutions**, **coordinate systems and data types**

- Hugin was extensively validated on Romanian usecases/datasets, and the SpaceNet Competition

- **Serverless** deployments (via **Google ML Engine**) facilitates the processing of **large scale datasets**

# Where we go from here? (Future Work)

- **Hyper-Parameter** optimization
  - **Grid** and **Random** search **working**
  - **Bayesian** and **Genetic** algorithm **methods** planed
- Methods for **topology evolution**
  - **Neuroevolution** and **parametric** methods (see Google AutoML)
- Support for other Deep Learning and Machine Learning tools: **PyTorch**

# Acknowledgments

- This work was primarily supported by **ESA** through the **ML4EO** Project, **EOSmith** and the Romanian National Research Authority through the **FUSE4DL** Project.

- Also, with **computing resources** generously offered by:
  - **Amazon Web Services**
  - **Google Cloud Platform**
  - **CloudFerro via ESA NoR**

- **GPU** resources offered by **West University of Timisoara**
  - Faculty of Mathematics and Informatics

- Partially motivated by the SpaceNet competition

# Thank you!

Any questions?