Intro to Google Earth Engine
and
Crop Classification based on Multispectral Satellite Images

Ivan Matvienko, MSc-2 IST

Intro2Intro2
Google Earth Engine

Google Earth Engine

- Satellite images:
  - Landsat
  - Sentinel
  - MODIS
- Big variety of features:
  - Climate and Weather data
  - Satellite images
  - Geophysical data (Cropland, Land cover, …)
- Access to Cloud Services:
  - Google Compute Engine
  - Google Cloud ML

How to work with it

- Interactive web browser JavaScript IDE
  - Easy prototyping and Instant visualization of results
  - Utilization of Google Cloud Services

- Python API
  - Not so easy to use
  - You cannot use Numpy with GEE =(

GEE Capabilities

- Big built-in library of processing functions:
  - Image Processing and Partially NLA (basically Eigen-analysis):
    - Convolutions, Gradients, Edge Detection e.t.c
    - Automatic Resampling, Scaling, Projecting, Mosaicking
  - Basic ML methods for:
    - Regressions, Decision Trees, Classification, Clusterization, TF Models to deploy
  - Basic Statistical methods

- 4 main operations:
  - Filter
  - Select
  - Join (combination of elements from different datasets)
  - Reduce (aggregation over bands, time)

Demo

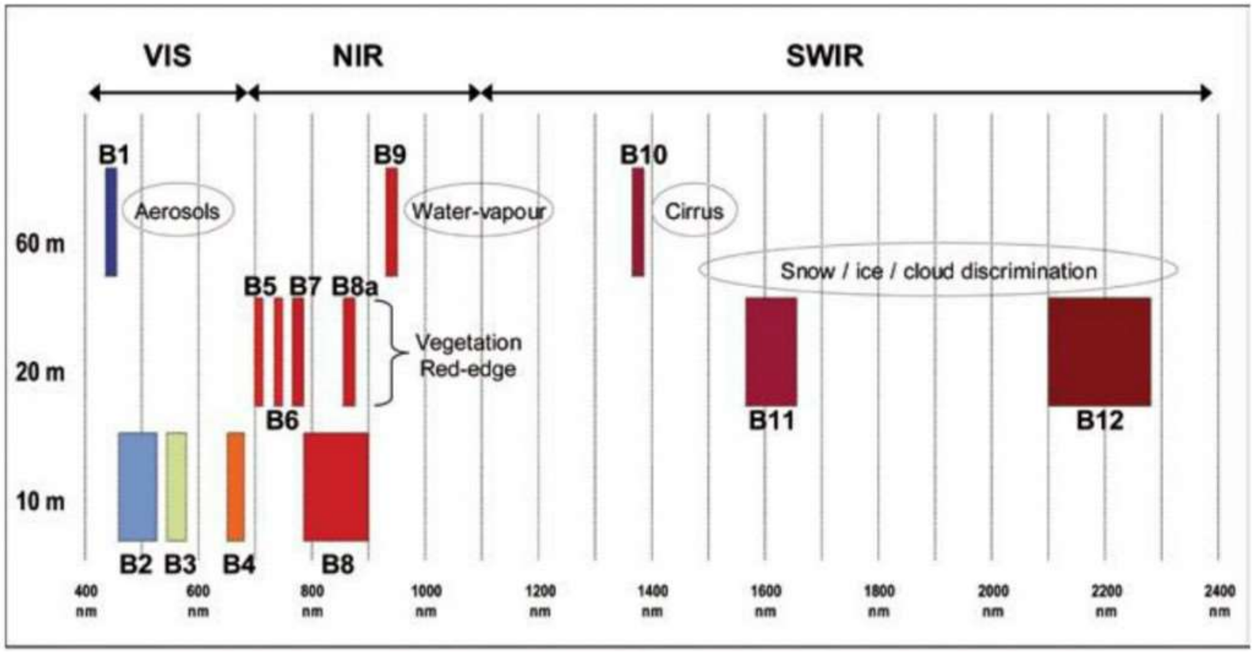
Crop classification using multispectral satellite images

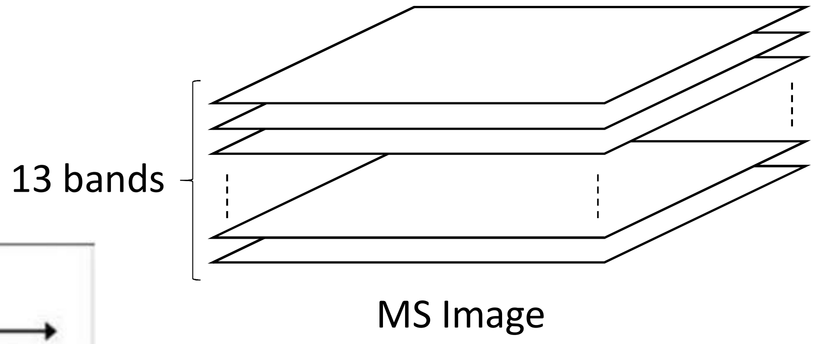
Problem Statement

- Small dataset of Sentinel2 L1C MSI with annotated and labeled agricultural fields
- Goal: classify every pixel of input image to certain crop class
  - Wheat
  - Sunflower
  - Corn
  - …

Multispectral Satellite Images

- Sentinel 2B
  - 13 bands (L1C) / 12 bands (L2A)





Data Processing Levels

- Level 0: Reconstructed, Unprocessed Instrument and Payload data (raw data)
- Level 1A: Level 0 time-referenced, radiometric and geometric calibration
- Level 1B: Level 1A processed to sensor units (TOA reflectance)
- Level 2: Derived geophysical variables at the same resolution and location as Level 1 source data (BOA/surface reflectance)
- Level 3: Variables mapped on uniform space-time grid scales
- Level 4: Model output and result of analysis

Data Processing Levels

- Level 0: Reconstructed, Unprocessed Instrument and Payload data (raw data)
- Level 1A: Level 0 time-referenced, radiometric and geometric calibration
- Level 1B: Level 1A processed to sensor units (TOA reflectance)
- Level 2: Derived geophysical variables at the same resolution and location as Level 1 source data (BOA/surface reflectance)
- Level 3: Variables mapped on uniform space-time grid scales
- Level 4: Model output and result of analysis

Data Processing Levels

- Level 0: Reconstructed, Unprocessed Instrument and Payload data (raw data)
- Level 1A: Level 0 time-referenced, radiometric and geometric calibration
- Level 1B: Level 1A processed to sensor units (TOA reflectance)
- Level 2: Derived geophysical variables at the same resolution and location as Level 1 source data (BOA/surface reflectance)
- Level 3: Variables mapped on uniform space-time grid scales
- Level 4: Model output and result of analysis

Level 1 vs. Level 2

Process Workflow

- Image Preprocessing
  - Applying Sen2Cor utility: L1 -> L2 satellite images (TOA -> SR)
  - Mosaicking of multiple images into one
  - Upsampling of low-resolution images (60m and 20m) to high-res (10m)
  - Noise Removal (opt)

- Dataset Processing
  - Data Augmentation (Affine Transform, Rotation, Crop, Cutouts, Flips)
  - Feature Engineering (Vegetation Indices, Vector norms, Neighbor Pixels, Convs)

- Model Building and Training

Classical ML Approach

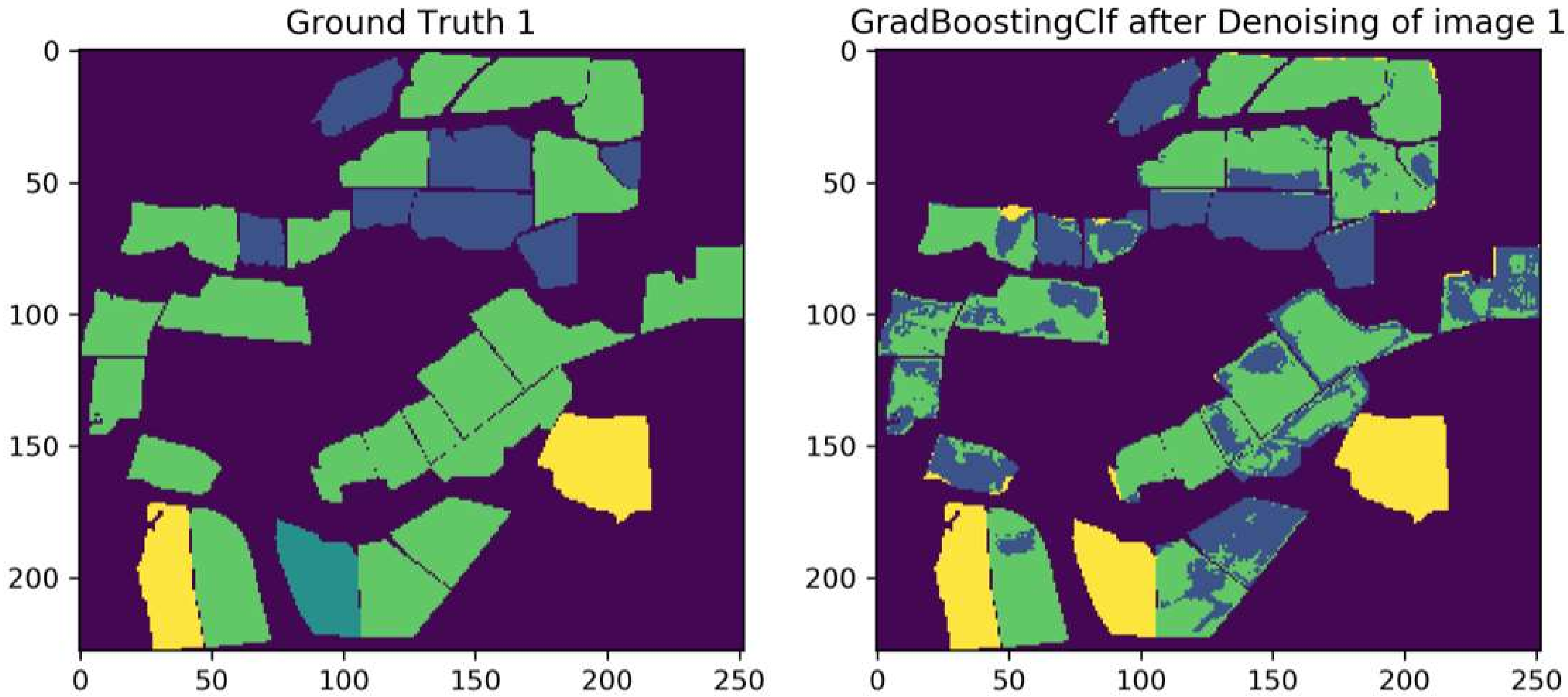
- SVM (4 hours for 250x250 image
- Decision trees
  - Random Forest (67.0% Tacc & 60.2% VAcc)
  - Gradient Boosting (72.5% TAcc & 65.2% VAcc)

- Implemented Laplacian-Gaussian Mixture denoising*
  - Improved results for ~2-3%

- Drawbacks:
  - Low accuracy (as well as other metrics)
  - Granularity of output

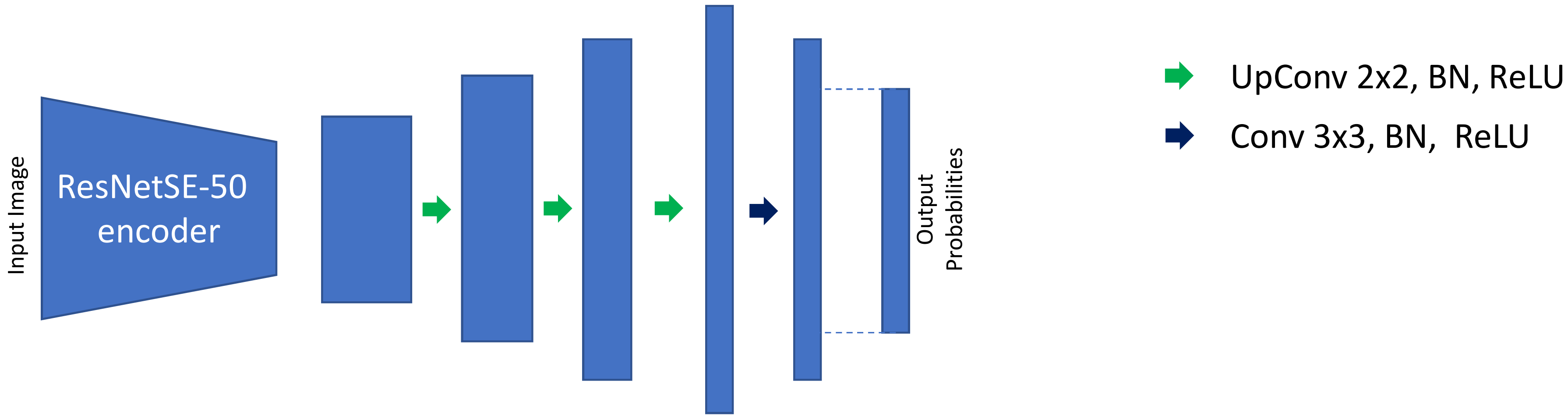
*A Robust PCA Approach With Noise Structure Learning and Spatial–Spectral Low-Rank Modeling for Hyperspectral Image Restoration - Wenfei Cao , Kaidong Wang, Guodong Han , Jing Yao, and Andrzej Cichocki

Classical ML Approach

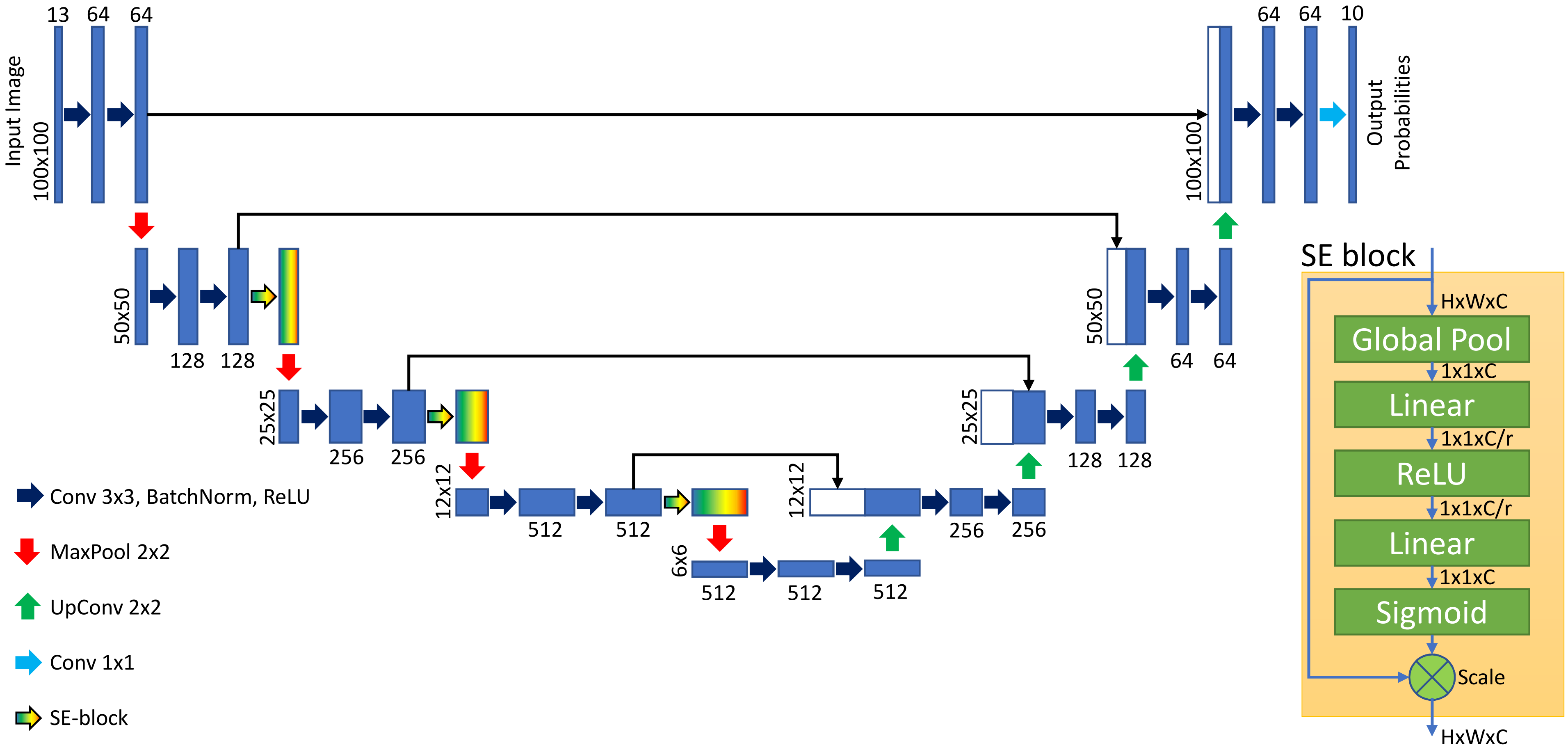


Deep Learning Approach

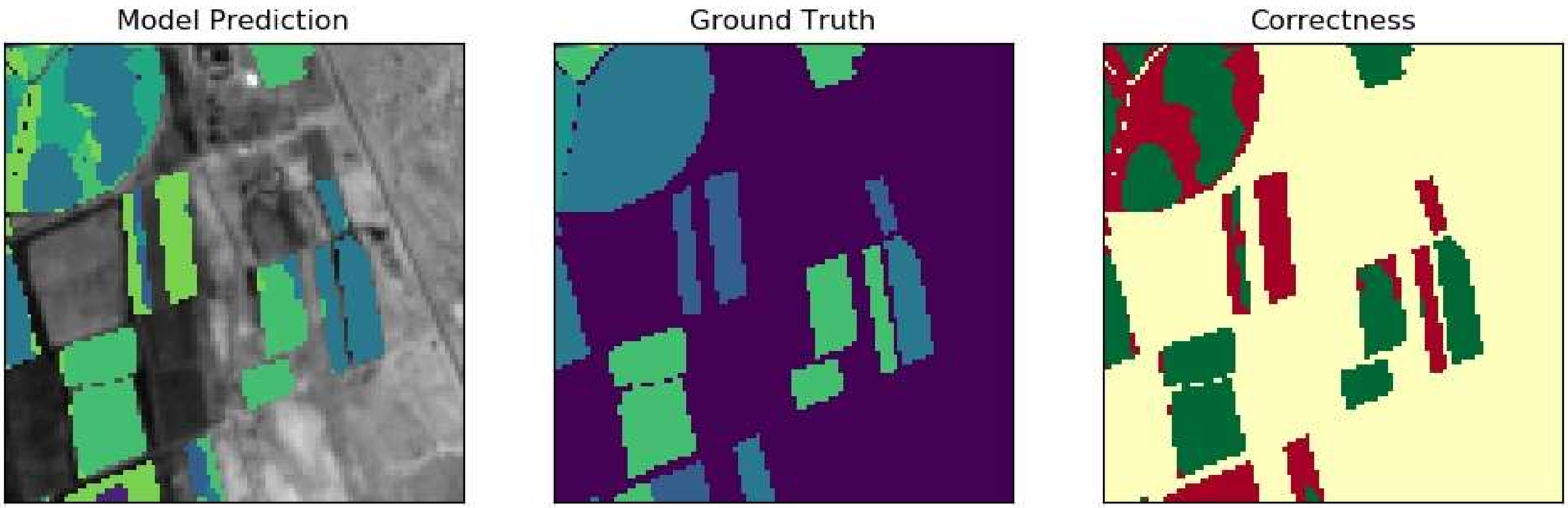
- U-Net architecture with SE blocks(74.2% TAcc & VAcc 73.9%)
- ResNetSE-101/50 encoder and Convolutional decoder (doesn't train)
  - For some reason loss does not decrease over epochs



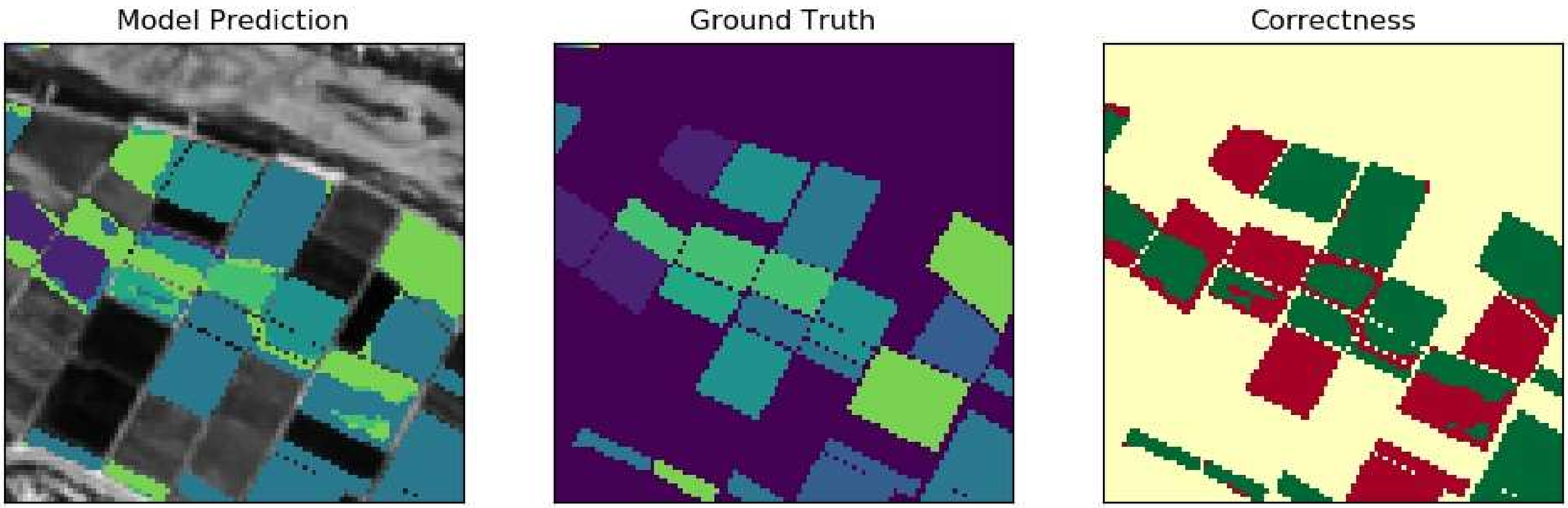
U-Net with SE Blocks Architecture



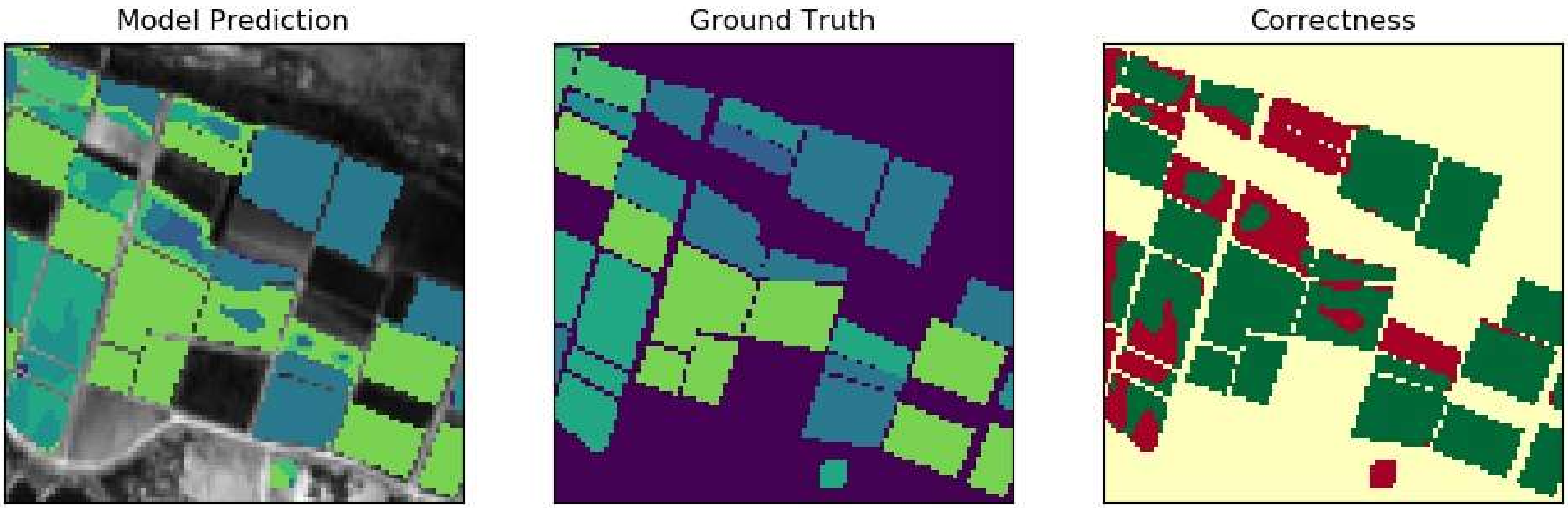
U-Net with SE Blocks Architecture



U-Net with SE Blocks Architecture



U-Net with SE Blocks Architecture



Parcel (SuperPixel) based Classification

- Apply Aggregation algorithm to the image (SLIC for example)
- Classify every parcel using much simpler CNN

Thanks