Mapping land use and land cover changes faster and at scale with deep learning on the cloud

Zhuang-Fang Yi [1], Drew Bollinger [1], Devis Peressutti [2]

1. Development Seed https://developmentseed.org/

2. Sinergise https://www.sinergise.com/

Introduction

Policymakers rely on Land Use and Land Cover (LULC) maps for evaluation and planning. They use these maps to plan climate-smart agriculture policy, improve housing resilience (to earthquakes or other natural disasters), and understand how to grow commerce in small communities. A number of institutions have created global land use maps from historic satellite imagery. However, these maps can be outdated and are often inaccurate, particularly in their representation of developing countries.

We worked with the European Space Agency (ESA) to develop a LULC deep learning workflow on the cloud that can ingest Sentinel-2 optical imagery for a large scale LULC change detection. Sentinel-2 has high temporal and spatial resolutions, is openly licensed, and can be freely downloaded from ESA or other service providers. The workflow, **Deep LULC**, aims to create automated, accurate, fast, scalable LULC maps to support decision making around natural resource management and urban resilience, especially in developing countries.

Our current workflow can be broken down into three steps:

1. Generating training data;
2. Training deep learning models on the cloud;
3. Predicting LULC over a new area of interest (AOI)

Methodology

The designed workflow is an end-to-end workflow that sits on top of two comprehensive tools, SentinelHub, and eo-learn. **Sentinel Hub** is a cloud based GIS platform for distribution, management and analysis of satellite data. **Eo-learn** is an earth observation processing framework for machine learning in Python. **Deep LULC** seamlessly link earth observation data with machine learning, and has a dynamic U-Net under the hood that allows user to train a LULC model quickly.

Deep LULC takes in the labeled LULC and associated AOI in shapefiles, set up a task to fetch cloud-free, time series imagery stacks within the defined time interval by the users. It will pair the satellite imagery tile with it's labeled LULC mask for the supervised deep learning model training on the cloud. On the deep learning model training side, we're using Dynamic UNet from Fast.ai. Fast.ai is a deep learning algorithms python package that lets users to train and test the best practices neural nets with their own data With dynamic U-Net under the hood of **Deep LULC, u**sers can swap-in a variety of models to be used as the UNet encoder. This allows users to quickly experiment and switch the models according to their task, AOI, diversity of data and desired LULC classes. Once a well-performing model is trained, it can be exported as a Tensorflow/Pytorch serving docker image to work with our cloud-based model inference pipeline, Chip n' Scale. Chip n' Scale is a open-source package created by Development Seed. It's a Queue Arranger helps users run machine learning models over satellite imagery at scale on the cloud. The inference pipeline can automatically scale with the number of images to be processed.

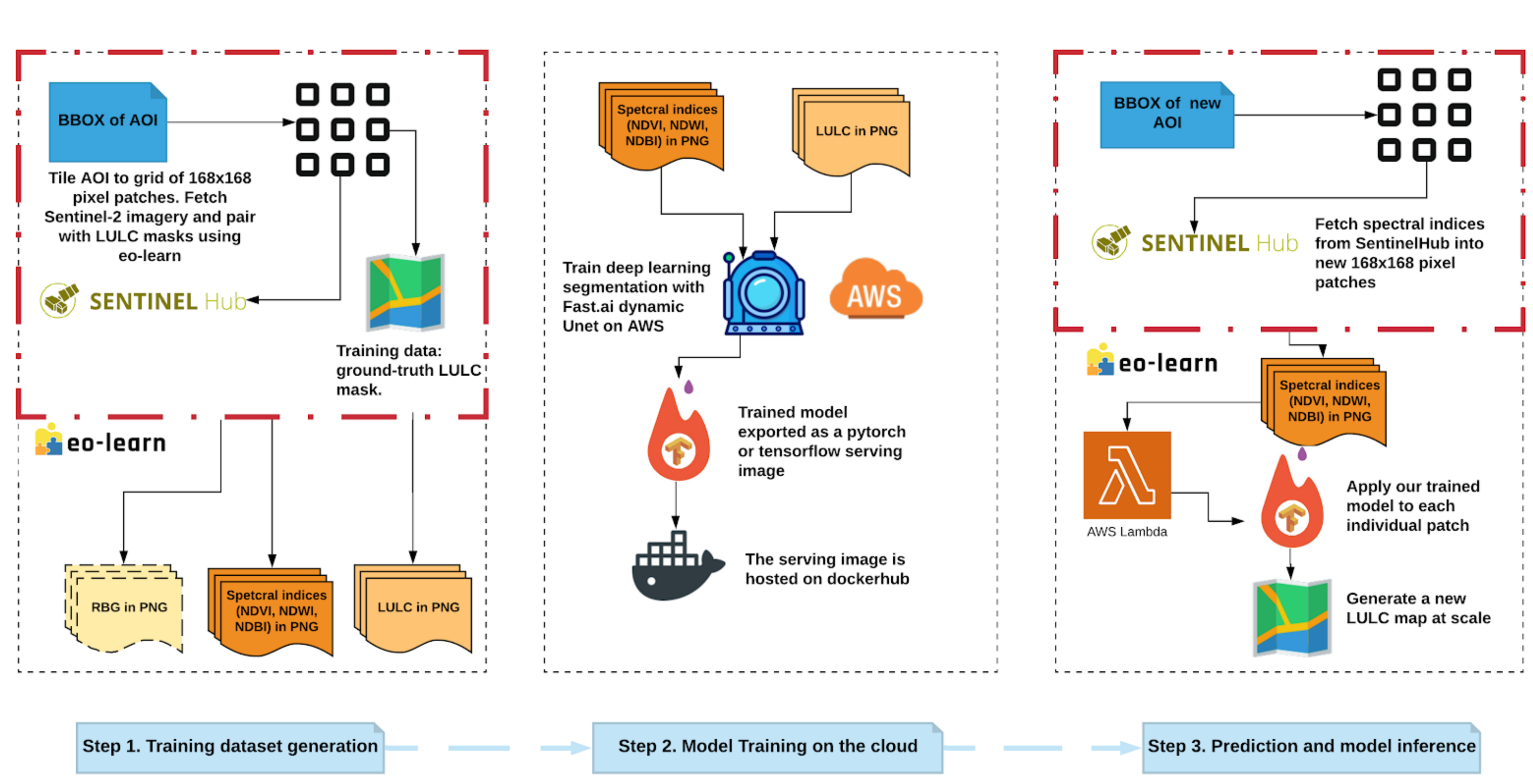


Figure 1. The deep learning pipeline that fetches and creates training data for LULC modeling on the cloud. It can be scaled up with our current open-sourced and cloud-based pipeline Chip n' Scale.

•For the training dataset generation, we removed too cloudy scenes, and data was saved as eopatches. A stacked numpy ndarray attached with spatial information. We went through:

– Check the ratio of the valid data for each patch and for each time frame
– Keep only time frames with > 99 % valid coverage (no clouds)

•Concatenate BAND, NDVI, NDWI, NDBI info into a single feature called FEATURES

•Perform temporal interpolation (filling gaps and resampling to the same dates)

– Create a task for linear interpolation in the temporal dimension
– Provide the cloud mask to tell the interpolating function which values to update

•Perform erosion

– This removes artefacts with a width of 1 px, and also removes the edges between polygons of different classes

•Random spatial sampling of the EOPatches

– Randomly take a subset of pixels from a patch to use in the machine learning training
– Burn LULC labeled data and NDVI, NDWI and NDBI into PNGs.

•Split PNGs for training/validation

– Split the the images into a training and validation set

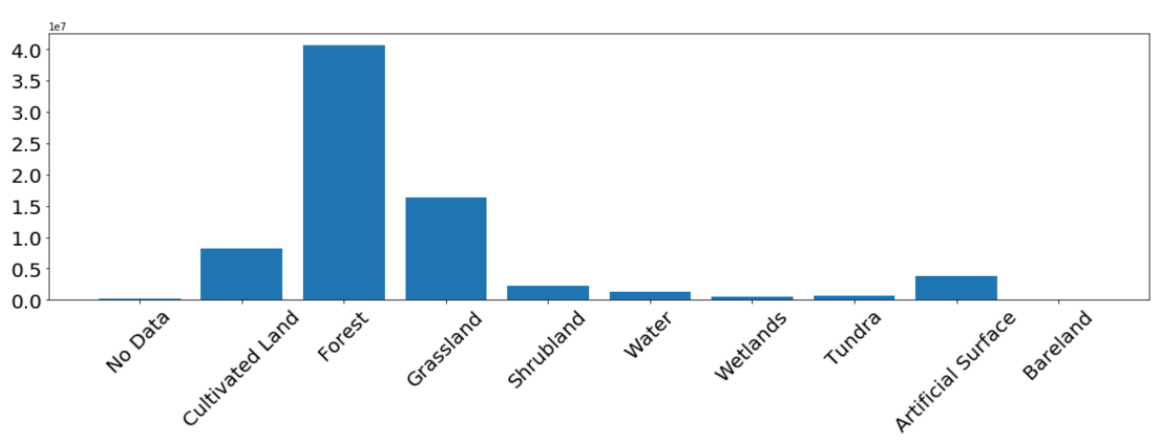
Results

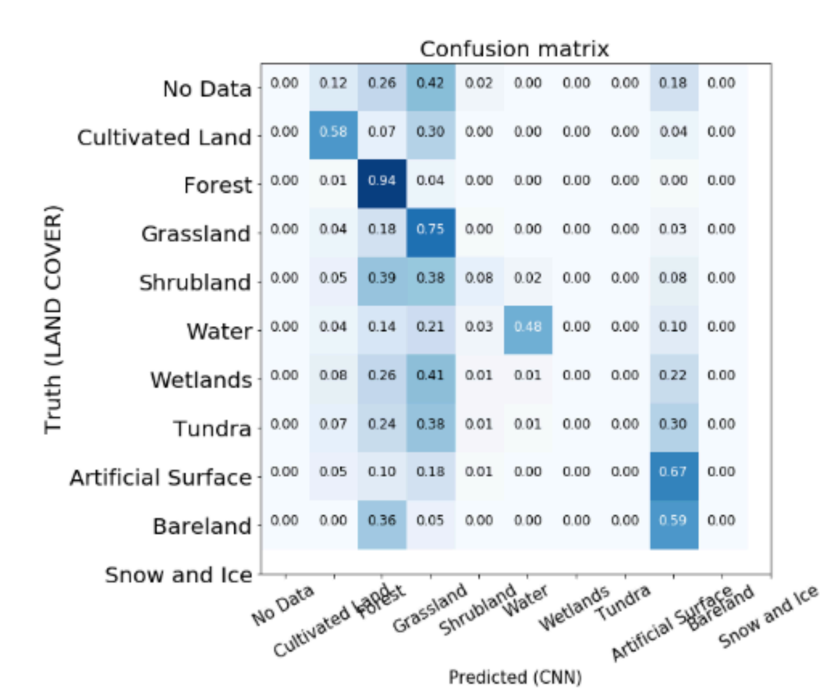
The model was trained for 20 hours, 200 epochs, with ResNet50 as the dynamic U-Net encoder. The training accuracy was around 0.83

epoch	train_loss	valid_loss	IOU
1	0.574539	0.536570	0.821021
2	0.568564	0.537619	0.821263
3	0.599832	0.540106	0.819846
4	0.582904	0.531692	0.822887
5	0.583801	0.533170	0.822409
6	0.588943	0.539949	0.820633
7	0.559916	0.531932	0.822631
8	0.581134	0.546181	0.819532
9	0.597752	0.550947	0.819598
10	0.596975	0.668728	0.815831
11	0.619852	0.532728	0.821346
12	0.594922	0.542480	0.820092
13	0.597265	0.543689	0.821547
14	0.583579	0.541271	0.820120
15	0.581756	0.538158	0.819519
16	0.609799	0.573464	0.817476
17	0.582316	0.555822	0.817495
18	0.619981	0.567032	0.815915
19	0.607304	0.535262	0.821584
20	0.630332	0.538771	0.821299
21	0.591000	0.536753	0.821243
22	0.606162	0.543471	0.820706
23	0.581930	0.552527	0.818242
24	0.575535	0.538081	0.821798
25	0.563449	0.531827	0.824028

The model was trained for 20 hours, 200 epochs, with ResNet50 as the dynamic U-Net encoder. The training accuracy was around 0.83.

However, the classes imbalance issue is pretty standout, e.g. forest has the lagest pixel pool but wetland, water etc. pixels are pretty small. When train the Deep LULC with U-Net, we found the classes have less pixel struggle and obtain less validation accuracy.





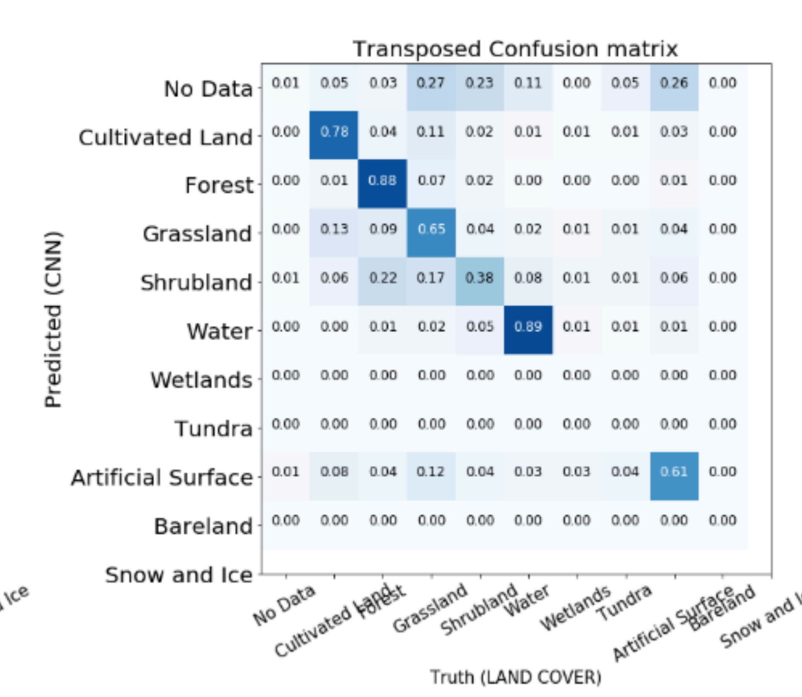


Figure 2. The pipeline fetches training data, Sentinel-2 optical imagery and labeled 10 classes LULC, for the AOI (in green) that covers 1296 km2 in Slovenia. A UNet using ResNet50 as the encoder was trained, and the model prediction was done over a new area (in red), and the LULC classification is shown on the right.

Conclusion

Land, forests, and water are intimately connected to how people live. Changes in land use are heavily influenced by human activities (e.g. agriculture, deforestation, human settlement expansion) and have been a great source of greenhouse gas emissions. Sustainable forest and land management practices vary from region to region, which means having flexible, scalable tools will be critical. With these tools, we can empower analysts, engineers, and decision-makers to see where contributions to climate-smart agricultural, forestry and urban resilience programs can be made. The pipeline, written into Jupyter notebook, is open sourced under eo-learn examples now.

Comparing to traditional tree-base and machine learning supervised learning algorithm, e.g RandomForest, Support Vector Machine and XGBoost, Deep learning LULC method is more sensitive to class imbalance. Model performs poorly to the classes have less data in the training dataset. However, a deep learning LULC workflow is more scalable once a good-performed model is trained and select with cloud GPU machines' computation powers.

Acknowledgements

We thank ESA fund this work.