Detecting Mines via Satellite Imagery

Jan 1, 2018

Krishna Bhogaonker, University of California, Los Angeles

Daniel Duckworth, Google Brain

Junchuan Fan, University of Maryland, College Park

Sina Kashuk, DataKind

Ryan Vilim, Sidewalk Labs

Katie Yoshida, Foursquare

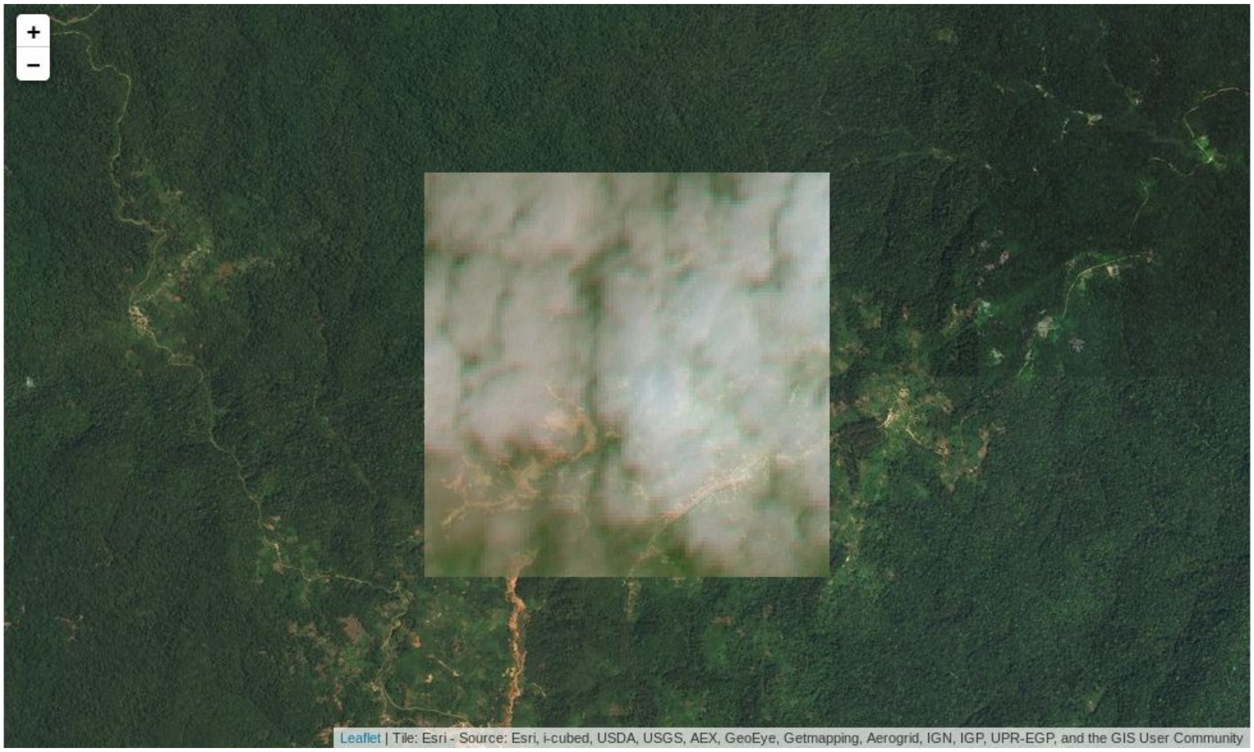
Summary

Between September and December 2017, 5 volunteers worked with DataKind to produce a proof-of-concept on the feasibility of **automatically detecting mines** via public satellite imagery. While remote sensing is a well-established method for detecting large-scale phenomena like elevation, vegetation, or temperature, it is an open question if the same data sources and techniques can effectively recover finer-grained concepts such as mines.

In this work, we conclude that **public imagery is sufficient** for detecting mining areas with high but imperfect accuracy. While results indicate that quality is too low to blindly rely on, we believe this method is the basis of a powerful tool for guiding future investigations.

Background

Irregular mining activities are known to have both positive and negative effects. For example, they provide employment to tens of millions of poor in developing countries, although their impact is mixed and hard to track, and in some cases, they have fueled conflict (eastern DRC), are often conducted under poor environmental practices (use of mercury for gold mining), and have led to mixed social outcomes (child labor but also higher family incomes). The insights gained can be passed to researchers and monitoring groups like Global Witness, who can then contact local groups to verify this acuity.



Cloudy Landsat 8 satellite image overlaid on a high resolution map. Paradiso Mine, Ituri Province.

Conclusions

TODO: What was critical to making it work? Which features? How much data? What model structure?

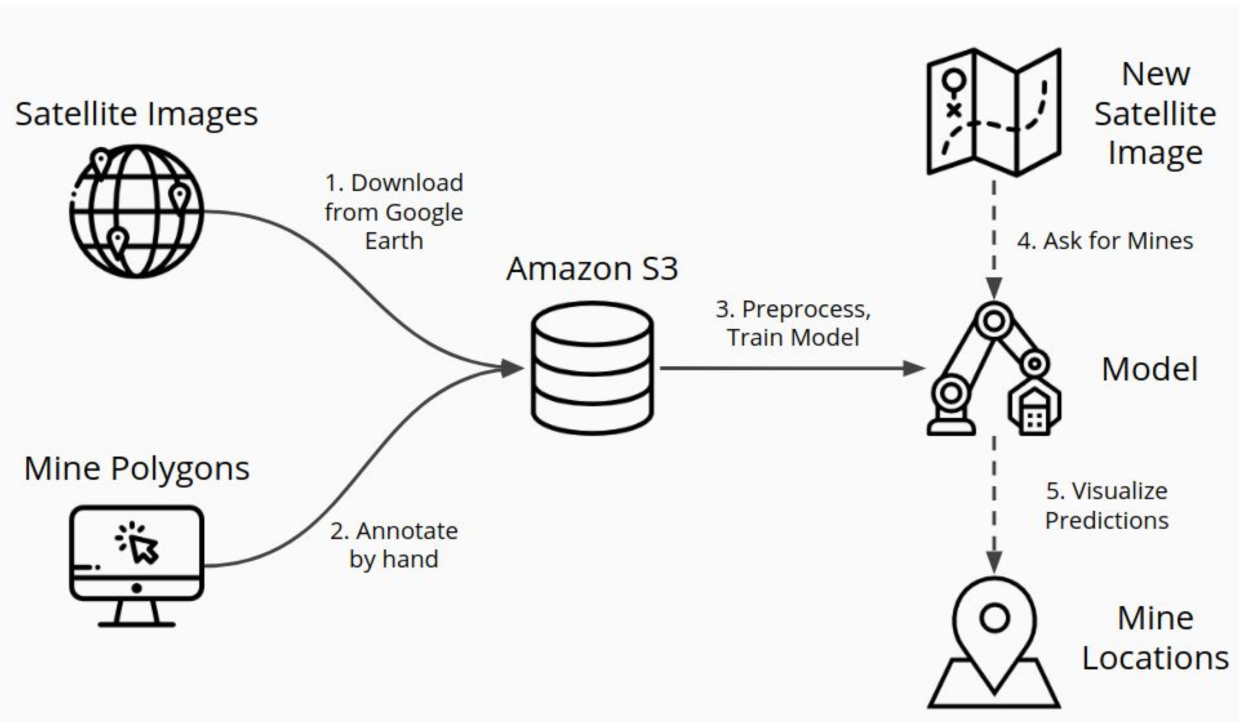
TODO: What did we think would work that didn't?

1. A Random Forest predicting mine/no mine per 30 m^2 pixel can correctly identify mine pixels with a **precision of 79.7%** and a **recall of 48.4%**. (These scores are obtained aggregating data from each site over multiple time periods; when using each time period individually we obtain precision of 25.9% and a recall of 38.4%.) For comparison, a random classifier has a precision of 0.55% and arbitrary recall.
2. Landsat Satellite imagery at **30 m^2 per pixel** spatial resolution across 11 spectral bands is sufficient for the previous results. Thermal Infrared intensity is the most valuable feature.

3. Public satellite imagery is **noisy**. Cloudiness, daylight, etc significantly impact prediction quality. It is important to average across multiple points in time to obtain accurate results.
4. In order to fit a model, it is imperative that polygons surrounding known mining sites be given. **200 labeled mining sites** was sufficient for the previous results. Typically, these must be drawn by hand using high resolution satellite imagery, such as Google Maps.
5. Convolutional Neural Networks are overly data-hungry for the small dataset available.

Technical Details

We now describe the pipeline used to produce the model and insights. The chapter below describes each section in more detail.



First, public satellite imagery is downloaded from a cloud provider (in our case, Google Earth). As raw satellite images are too large to fit on a typical home computer, small image patches around a suspected mining site are cropped out.

Simultaneously, mining sites are manually verified, and a domain expert draws polygons around visible mines by hand. The domain expert typically has significantly higher resolution imagery than is publicly available, such as that provided by Bing Maps.

Next, the two datasets are merged and presented a machine learning algorithm. A model is fit to classify the image's pixels as "mine" or "no mine", as was labeled by the domain expert. The model is chosen to maximize is accuracy on a held-out set of satellite image / mine polygon pairs.

Finally, the model is used to make predictions on new satellite images. Unlike the images used for training, we do not know if the image contains a mining site already. The model's predictions are then visualizing by domain experts for insight.

We note that the goal of this project is *determining feasibility*. As a result, the setup described below works well for a "small data" regime. Applying the same approach on larger geographical regions (e.g. provinces, countries) will require re-architecting the code. We believe the same methodology will work without modification.

Data

The data used in this project is entirely **geographic** -- that is, it is connected with location. Satellite imagery, distance to nearest body of water, and mining site locations are several such data sources used in this project.

The data used comes in one of two forms: rasterized images (e.g. pixels) and vector shapes (e.g. polygons). As predictions are made **per pixel**, all vector shape data is ultimately rasterized before being used in modelling. The data generating process can be summarized as,

1. Identify 3 km * 3 km “mining sites” in the DRC to model. One site is chosen per labeled mining site polygon (see *IPIS Mine Locations* below).
2. Split each 3 km * 3 km site into 30 m * 30 m partitions, hereafter referred to as a “pixel”.
3. Compute features (see *Sources* below) for each pixel.
4. Train a model to predict if a pixel intersects a mine, given the pixel’s features.

Based on our group’s meeting with Global Witness and domain experts, our hypothesis is that the likelihood for a image pixel containing a mining site is related with surrounding spatial features (e.g., hydrology feature, transportation feature). To incorporate the spatial relationship as a feature for our model, we combined directly observed features about a pixel (e.g. its intensity in the “blue” color range) and its relationship to other vector shapes (e.g. distance to nearest body of water).

Sources

Landsat 8, Tier 1: Landsat is a “the longest-running enterprise for acquisition of satellite imagery of Earth”[1]. Since 1972, eight satellites have been deployed to continuously capture satellite imagery of the Earth. We employed imagery from Landsat 8, as it captures the most spectral bands (11) at the highest resolution (30m by 30m per pixel) compared to other Landsat satellites, is easily accessible via Google Earth Engine, contains snapshots of the same location at multiple points in time, and is of acceptable quality at Tier 1. The intensity of each of Landsat 8’s spectral bands are used to construct 11 features per pixel, per timepoint. As for timepoints, our analysis used seven snapshots taken between October 15, 2016 and May 9, 2017.

Landsat 8	**Bands**	**Wavelength (micrometers)**	**Resolution (meters)**
Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS)	Band 1 - Ultra Blue (coastal/aerosol)	0.435 - 0.451	30
	Band 2 - Blue	0.452 - 0.512	30
	Band 3 - Green	0.533 - 0.590	30
	Band 4 - Red	0.636 - 0.673	30
	Band 5 - Near Infrared (NIR)	0.851 - 0.879	30
	Band 6 - Shortwave Infrared (SWIR) 1	1.566 - 1.651	30
	Band 7 - Shortwave Infrared (SWIR) 2	2.107 - 2.294	30
	Band 8 - Panchromatic	0.503 - 0.676	15
	Band 9 - Cirrus	1.363 - 1.384	30
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19	100 * (30)
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51	100 * (30)

* TIRS bands are acquired at 100 meter resolution, but are resampled to 30 meter in delivered data product.

In addition to the spectral bands, Tier 1 data also includes a quality assurance band (“BQA”). which contains a cloud confidence rating of each pixel, from 1 = low cloud confidence to 3 = high cloud confidence. We applied a filter to remove the high-confidence

[1] https://en.wikipedia.org/wiki/Landsat_program

clouded pixels from our dataset.[2] Pixel locations were removed on a per-timepoint basis, i.e., if a location was clouded at two timepoints, five pixel instances of the location still remained in our dataset. However, the clouded pixel was removed across all bands at that timepoint. Some images also had poor data; these pixels were also removed (similar to cloud filtering, the pixel was removed only at the poor timepoint in question but across all bands at that timepoint).

In order to account for any seasonality in the foliage, we create a feature which encodes the month the image was taken in. As we would like this to be a continuous, rather than categorical feature, we calculate it as

$$f = \cos\left(\frac{\text{Month Number}}{12.0}\right)$$

this means that January and December will have approximately the same value.

IPIS Mine Locations: The International Peace Information Service provides an extensive survey of mining locations throughout the DRC, with a particular focus on North and South Kivu. Based on this information and satellite imagery provided by Bing Maps (accessible via QGIS's OpenLayers plugin), we were able to draw polygons around visible mining sites. These polygons are then rasterized. A pixel is said to have a "mine" label if it intersects a mining polygon.

Hydrology: This dataset contains polygons surrounding all major bodies of water in the DRC. It is produced by World Resource Institute, and is the best of the publicly available hydrology datasets for DRC.

This dataset is used to calculate the distance to nearest body of water for each pixel, producing a single feature. As water is required for extracting mined minerals, we believe distance to be correlated with mining sites.

Provincial Boundaries: Subnational geographic divisions based upon the GADM Global Administrative Areas database. These polygons allow us to identify regions likely to be associated with mining via news articles. This data is used purely for exploration.

Digital Elevation: Congo digital elevation data produced by Shuttle Radar Topography Mission (SRTM).

Road Networks: Road network data from OpenStreetMaps and Diva-GIS

1. Implementation-wise, how do we go from raw data to data models can ingest?

Models

[2] The Landsat data also includes a similar confidence rating for water bodies; we did not include a water filter in our data processing, but it would be easy to implement and could be something to consider in the future.

The class of machine learning models which gave us a good mix of simplicity and predictive power was a **random forest** model. Random forests were a sensible choice because they are relatively robust to class imbalance and more difficult to overfit; this was important as mines pixels occupy only 0.55% of the Landsat images in our training set (which oversamples Landsat images which contain mines).

Other models that would be similarly reasonable are those that are not sensitive to class imbalance (e.g., gradient boosting). We would not recommend any models that are sensitive to class imbalance (e.g., logistic regression models), or models that require significantly more training data (e.g., convolutional neural networks).

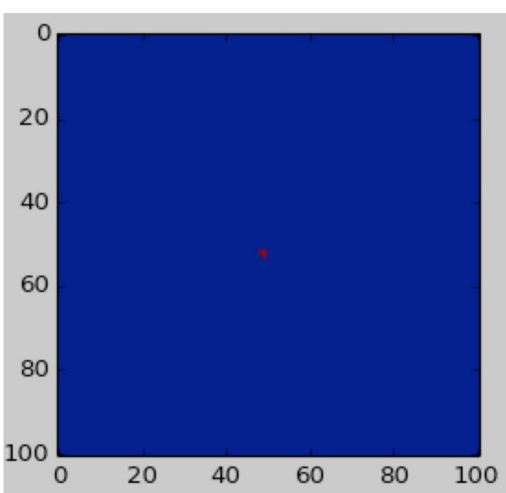
We trained the random forest model on a per-pixel basis. Predictions were made on each pixel at every time period, and the median prediction was taken for a given pixel over time (because we only want to predict that a single location has a mine or does not, over all time periods). The output of the model is a mine/no-mine prediction for each pixel location (time-invariant).

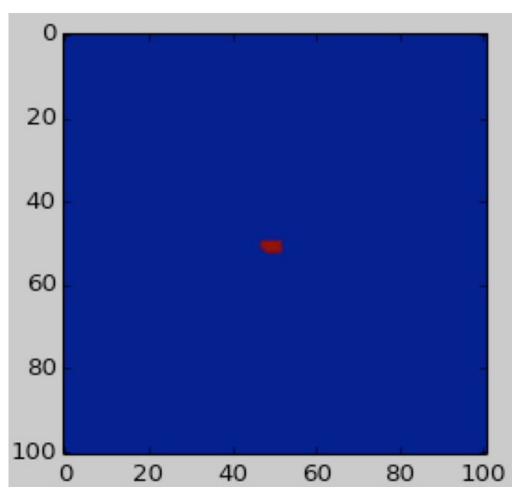
A gridsearch was used to optimally tune the random forest model parameters. We optimized for recall, but found that all metrics improved together during our gridsearch.

Images were split into cross-validation groups by hand to ensure that the same site was not seen in multiple splits (effectively sharing information between them).

Predictions

While our metrics are reported in terms of the proportion of pixels correctly identified as mines, this can be misleading. Particularly when we have aggregated the results from multiple predictions across time, the model often predicts mines the correct location, with slightly different boundaries. This means that while the model may have incorrectly classified a significant fraction of the pixels at a site, it still correctly identified the site. For example:

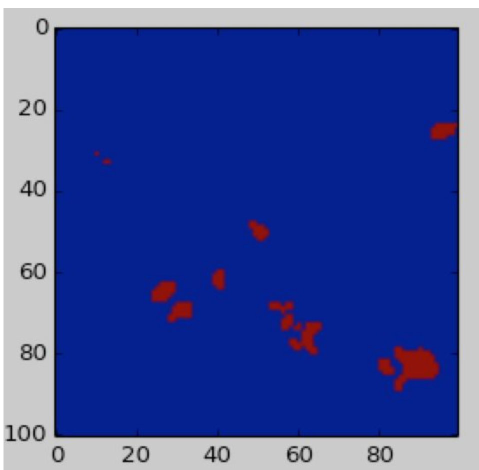
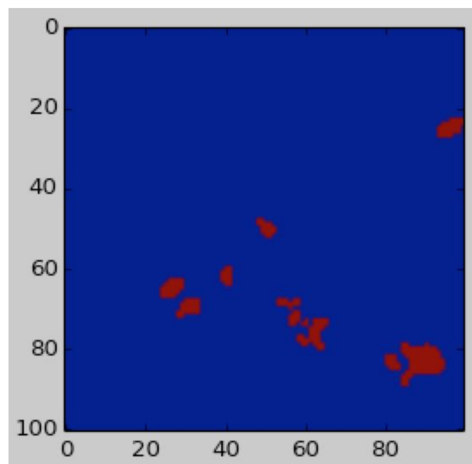




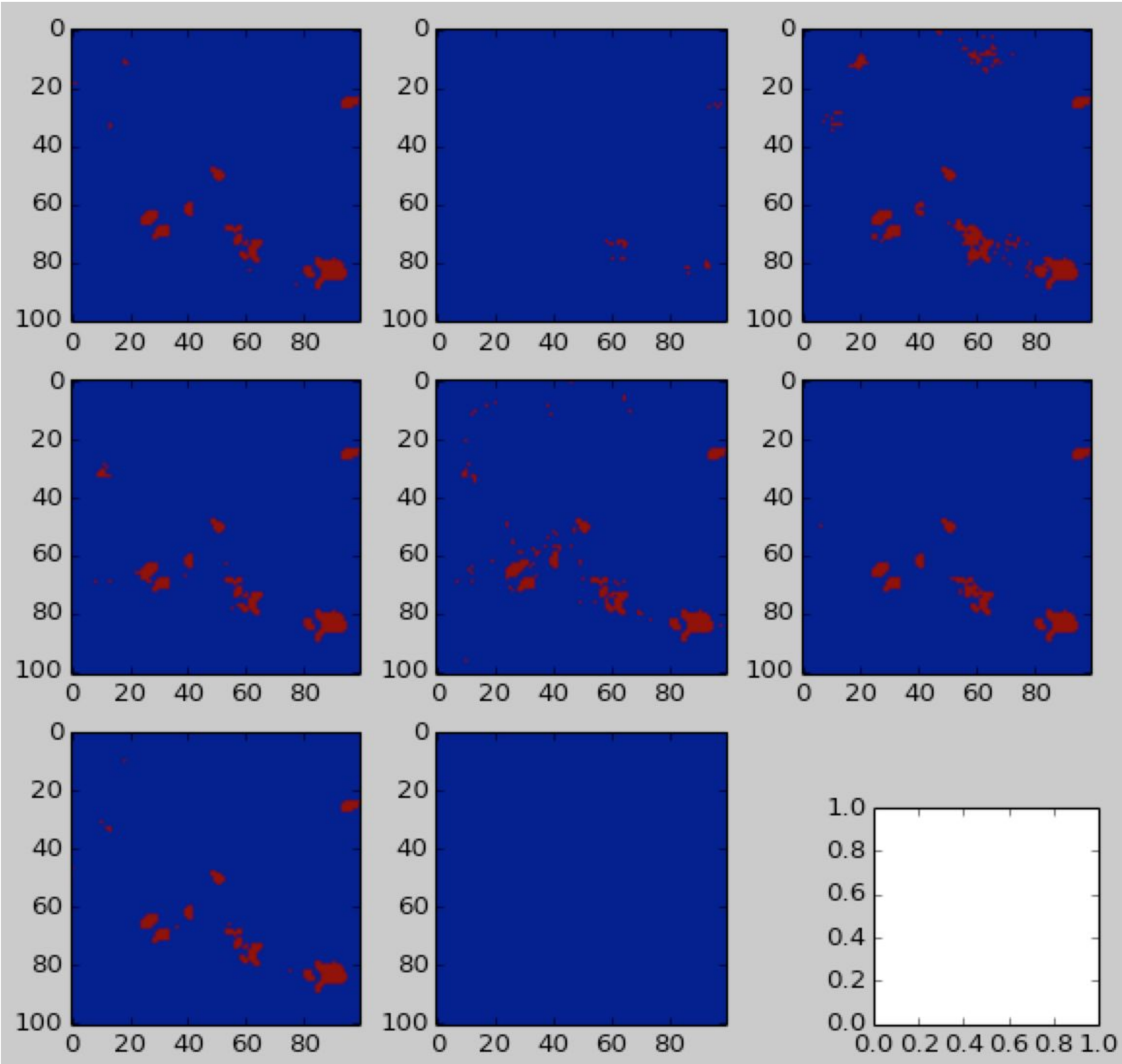
Predicted mine site (in red)	Ground truth mine site (in red)

In the above image, the predicted mine extent is shown in red on the left and the “truth” mine extent is shown in red on the right. Most of the mine site has been incorrectly classified, but the presence of a mine was still correctly identified by the model. Future work could focus on improving the metrics for which we measure the success of the models predictions to take this into account.

We find that while taking the median of our predictions across time improves the performance of our models, the model tend to produce broadly similar output for the same sites at different times, there are differences. For example below we show a mine’s actual extent (right) and the time-aggregated predicted extent (left)

	
Predicted mine site (in red)	Ground truth mine site (in red)

If we compare this to the predictions from the eight times this site was imaged over six months, we see that the predictions are often incorrect. Taking a vote (median) of all of these predictions produces a result that is closer to the true mine extent than any of the individual predictions. The benefit of aggregation might be explained in part by the possibility of clouded imagery at any given timepoint.



The predicted mine extent (in red) for eight independent measurements of the same site.

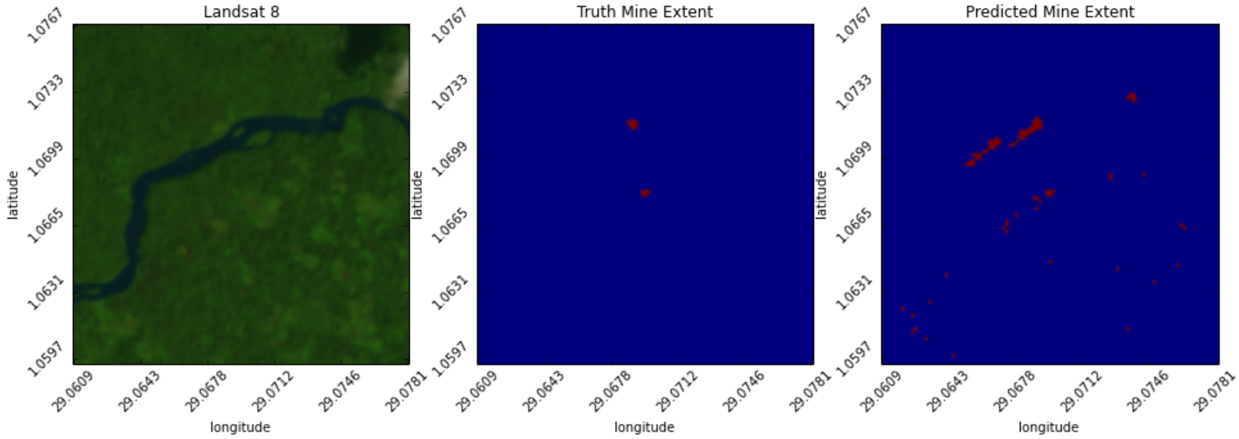
Caveats

The accuracy numbers we report here should be taken as a rough guide only. The hand labelled masks were largely judgement based and are not necessarily exhaustive. Furthermore, we have taken the “not mine” class to be everything that has not specifically been labelled as a mine. This leaves open the possibility that regions should which should be mines have been labelled as “not mines”, and vice versa. This mislabelling uncertainty will have an impact both on the machine learning model, and on the performance numbers.

Product

To showcase model quality, we developed a Jupyter Notebook for visualizing satellite imagery and predictions.

As a model assigns each pixel a probability of containing a mine, the most immediate way to understand a model's assignments is to visualize them side-by-side with a satellite image.



To give additional context, the prediction and satellite image can also be overlaid on a world map, alongside other predictions. A more complete version of this product could expand predictions over an entire region or country rather than over select locations. A Jupyter Notebook prototype for such visualizations is available on Github.

1. What insights do these predictions give?

In general, the model appears to do a reasonably good job at identifying known mines (*See 'Predictions' section above*.) . There are some instances in which it identifies areas that are not known mines, but we don't have a way to confirm that they are not mines.

2. What's keeping this setup from being a "real product" (rather than just a proof of concept)?

With this proof-of-concept, we see that the fundamental approach is sound: low-resolution satellite imagery and a small number of labeled mining sites is sufficient for building a high-accuracy mine detection model. To extend beyond a proof of concept, the primary limitations are **storage/compute** and **tooling**.

Even with the models trained today, making predictions on North Kivu alone (a single province) requires 1.98 billion predictions, and the entirety of the Democratic Republic of Congo requires 78.16 billion. To construct the latter, up to 200.2GB of satellite imagery is required.

A significant challenge in this task will be a data processing pipeline which is capable of preparing and storing the feature vectors in a timely manner. Once this happens the Random Forest model itself presents no major barrier to scalability. A simple test which timed the inference of data already pre-processed and loaded into memory found the inference step of the random forest model could perform ~200000 predictions per second. Thus the bottleneck of a production level model is likely to be feature pre-preprocessing and data storage.

Even with storage and compute, tools capable for presenting, analyzing, and communicating insights provided also need to be built. The precise form of these tools is application-dependent and out-of-scope for this project.

Recommendations

1. What other data do we wish we had?

Larger mine polygon dataset: While there is a virtually unlimited supply of satellite images, the number of labeled mining sites is much smaller. Labeling a mining site requires visually inspecting satellite imagery and outlining, by hand, visible indicators of mining. This is an extremely time-intensive process (~10 seconds per visible indicator, 1~50 indicators per mine), and is the most important limiting factor in scaling this methodology.

High resolution satellite imagery: When identifying mines by hand, a human looks for textures that are only visible via high resolution imagery, such as fallen trees or pools of water. These features are visible at 50 cm^2 resolution but not at 30 m^2.

Digital elevation model time series: According to our discussions with domain experts, topography and change of topography are important indicators for mining activities.

Therefore, using a temporal sequence of DEM can help us detect the change within a spatial region. This feature could potentially reveal valuable information about mining activities within an area.

2. What's limiting the project now? If we had more time, money, compute, what could we do that we can't do now?

Scaling up: With more resources, we would immediately be able to scale the solution to larger geographical areas. An unrealized goal of this project was to apply it to the entirety of the DRC. Unfortunately, this would require 114 GB for input features and 2.6 billion predictions. This is easily manageable on a distributed Cloud compute setup of ~100 machines, but is infeasible with our current software.

Domain Expertise: Expanding the mine polygon dataset is the single largest thing one can do to improve model quality. Hiring domain experts to assist in labeling mining sites would significantly increase the quality of our model.

Creating a Product: The modeling technology here is a starting point, but creating an actual product that solves a partners' problem is the ultimate goal.

3. If we had to do everything again, what would we do differently?
4. What questions could we *not* answer with this setup? What needs to change so we can?

Appendix

Previous Work

The methodology described here is very similar to that described in Saavedra & Romero, 2017. In particular, Saavedra also uses Landsat satellite imagery, hand-drawn mine polygons, and a random forest model. In addition to Landsat satellite imagery, the authors also use "eforestation year and ecosystem type" as inputs.

Unlike the work presented here, Saavedra's focus is on Columbia, and the conclusions are primarily econometric. The authors were unable to share their underlying data and models, so we are unable to verify the similarity of their modeling setup.

Artifacts

Code

All software for this project is available for download on Github under `datakind/ON-MiningDetection` (permissions required). The code required to prepare the data, train, and run the model is in `datakind/ON-MiningDetection/model` along with instructions on which each file does.

Data

All data used in this project is available under `s3://mining-detection`. See the following files for specific details.

2017-11-11 | Mining Images (Water, Landsat, Masks).tar.gz: Rasterized satellite images, mining polygons, and distance-to-water for each pixel. Used as training data for models. Each image is 100-by-100 pixels with 30 m^2 per-pixel resolution. Locations are centered over locations in *IPIS Mining Site Masks* dataset below. See storage_api for examples on how to load and visualization_api for examples on viewing.

2017-11-06 | IPIS Mining Site Masks.tar.gz: ESRI Shapefile containing 200 mining polygons in vector form. Polygons drawn in QGIS. Locations chosen to be close to points in the IPIS locations dataset. Drawn based on base tiles provided by Bing Maps using the OpenLayers plugin between August and November 2017.

1. What format is it in? How to load it up?

All rasterized data is stored in an on-disk format described in storage_api. See Notebook for a demo.

Models

1. Where are models stored?

s3://mining-detection/model/final_model: Serialized Scikit-Learn Random Forest model. Can be used to make predictions on new pixels. Expects each example to contain pixel intensities for Landsat 8 bands B1-B11. Stored as a pickled Python object. See scikit-learn documentation for model details.

Instructions for loading and running these models are located in the Readme.md file in the Github repository.

Product

1. How to run demos?

All visualizations produced via a Jupyter Notebook stored on Github. Requires an on-disk dataset such as *2017-11-11 | Mining Images (Water, Landsat, Masks).tar.gz*. See installation instructions for setup and use.