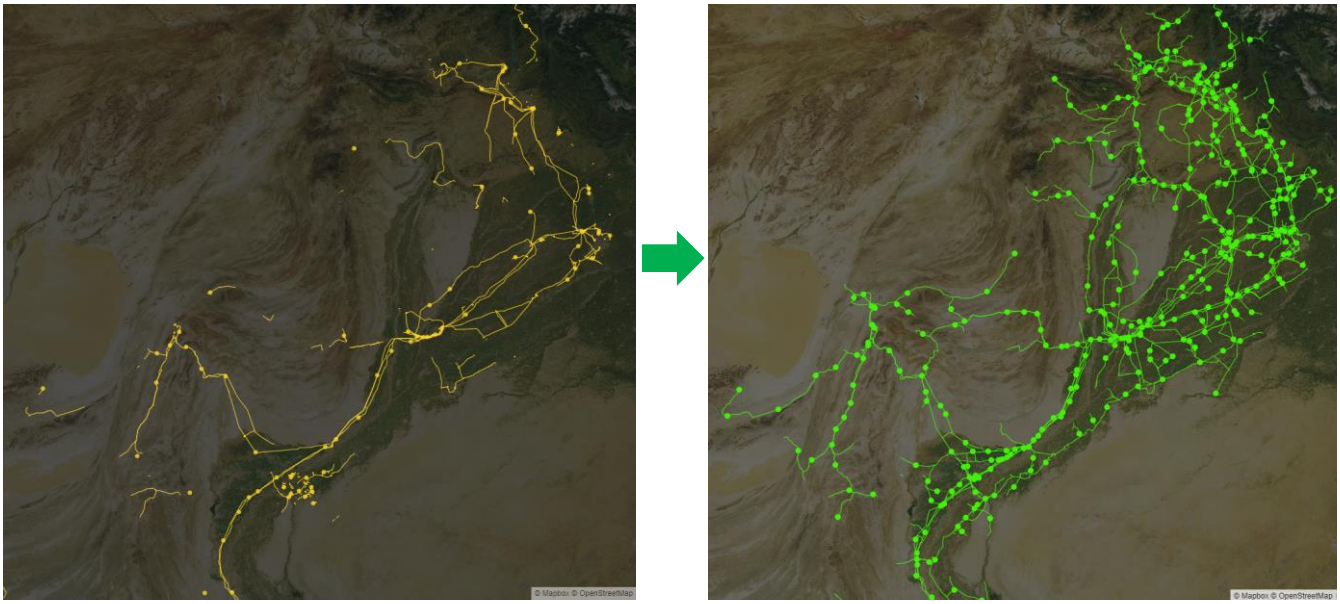
MACHINE LEARNING FOR HIGH RESOLUTION HIGH VOLTAGE GRID MAPPING

Pilot project for Nigeria, Zambia and Pakistan

June 2018



This report was prepared by [Development Seed](https://developmentseed.org) under contract to The World Bank.

It is a deliverable under the "WBG Energy & Extractives Open Data and Analytics" Project (ID: P161394), which intends to provide World Bank Group (WBG) clients, staff and other stakeholders with facilitated access to energy sector related data and advanced analytics toward informing strategic and project level decisions.

The methodology of this report is based on the previously piloted project on "Machine Learning for Africa's Grid", which was further refined based on additional tests and pilots with different ML approaches and datasets.

This project was funded and supported by the Energy Sector Management Assistance Program (ESMAP), a multi-donor trust fund administered by The World Bank.

This document is an **interim output** from the above-mentioned project. Users are strongly advised to exercise caution when utilizing the information and data contained, as this has not been subject to full peer review.

Telephone: +1-202-473-1000

Internet: www.worldbank.org

INTRODUCTION

The objective of this project was to support the SDG7 goals on energy access and renewable energy, and broader energy sector development, by providing governments, commercial developers, and researchers with access to high quality data on the location and characteristics of high voltage electricity transmission lines in WBG client countries to facilitate geospatial planning and project development.

The deliverable explores further the possibility to automatically detecting transmission lines from satellite imagery using neural network machine algorithms. It follows results from a pilot activity undertaken from March-September 2017 under the project that demonstrated the possibility to identify HV grid power poles from SAR imagery, providing a new opportunity for fast and cost-effective power grid mapping at scale (results of the pilot activity are available at http://devseed.com/ml-grid-detection/).

The project implemented machine learning based grid mapping for three countries to further refine and validate the methodology preliminary developed with the pilot, in view of developing a solid proof of concept supporting a potential scale-up in subsequent phases of the project. Data, algorithms and learnings resulting from this deliverable will be made available publicly to help inform and foster further efforts in the space.

As a natural continuation of the previous pilot project, the same company – Development Seed was contracted for implementation of this project. Development Seed is a leading creative strategy and engineering team. It develops online communications strategies for global organizations and builds open source solutions that solve complex communications challenges. They specialize in creating intuitive interfaces that effectively communicate large, complex data.

EXECUTIVE SUMMARY

The goal of the assignment was to determine and develop an open-source, cost effective and accurate method for identifying and mapping the high voltage transmission network and apply it on three pilot countries: Nigeria, Zambia and Pakistan.

Problem

Access to electricity is still limited in many developing nations. A variety of factors underlie this problem, but one core issue is that a map of the high-voltage (HV) infrastructure rarely exists. The schematics that are available are often outdated, incomplete, or split between a number of different agencies. Without a centralized map, governments or other organizations do not have the knowledge to make informed decisions on where to invest in maintenance or expansion of the electric grid. This lack of information also complicates decisions about alternative energy sources — without knowing where the conventional grid exists, it's hard to intelligently deploy alternatives like solar- or wind-power (Szabó et al., 2011). Beyond just the map itself, governments and organizations need a fast and cost-effective mapping pipeline. The HV grid is always in flux, so the ability to create an accurate snapshot at regular intervals is an important goal. Toward this goal, we partnered with the World Bank to develop a pipeline capable of efficiently mapping HV infrastructure at a country-wide scale.

Figure 2. Examples of HV towers in satellite images.

Toward solving this problem, a pipeline was built to efficiently map the high-voltage (HV) grid at a country-wide scale. This pipeline relied on both machine learning (ML) and Data Team – a group of eight professional mappers. The ML component processed satellite imagery across an entire target country and returned geospatial locations likely to contain HV towers -- the tall metal structures that support HV lines running for hundreds or thousands of kilometers. The Data Team then overlaid this information on top of satellite imagery and used it as a guide to help quicken their mapping of HV towers, lines, and substations. With this overlay, they could focus their attention on high priority areas and avoid the tedious task of reviewing entire countries worth of imagery by hand.

Using this pipeline, nearly all of the HV network in Pakistan, Nigeria, and Zambia were mapped and it was found that using the ML model increased mapping speed **33-fold** per km^2, compared to a purely manual approach. Examples of satellite images and the final workflow are below. The model, code, and mapping output are openly available, and the possibilities for further improvements of this pipeline are outlined in the Discussion section.

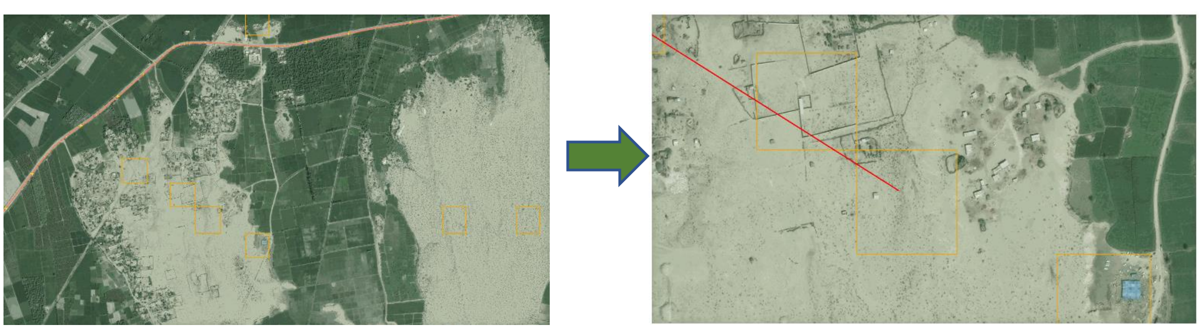


Figure 1. Tracing with ML-derived prediction overlay. Each square represents a tile that the ML model believed to contain a HV tower. The Data Team then mapped from tower to tower using this overlay as a guide. Video reflects actual speed.

Strategy

To generate a country-wide high voltage map, it was hypothesized that a two-part processing pipeline would work best. A machine learning model comprised the first component, which predicted the probability of a HV tower within individual zoom 18 satellite image tiles (or about 0.5 m/pixel). Its strength is processing speed – when running on a modern graphic processing unit (GPU), the model is capable of processing hundreds of thousands of images per hour. It was not expected to yield an extremely high accuracy, however, this first component acted as a first pass at detecting HV towers. The second component in the pipeline was made up of professional mappers. Humans are naturally more accurate at this task and can easily recognize HV towers against a wide range of backdrops from desert to dense forests. Therefore, the mappers could accurately trace from tower to tower and note questionable features that need additional validation. The goal was to combine these two components into an accurate but fast pipeline:

1. The ML component acted to rapidly produce a prioritization map (Figures 3 and 4), which then guided our Data Team of human mappers to focus on high-value areas likely to contain a tower.
2. The Data Team could then trace every component of the HV grid whether the ML model detected it or not. This ensured that their efforts were intelligently allocated instead of using a brute force approach of manually reviewing every meter of ground. It also meant that every edit added to OpenStreetMap was made by a human -- an important validation step to avoid incorrect changes to this community-driven map.

Figure 3. Find HV towers in satellite imagery with ML. The goal of the ML component was to detect as many HV towers as possible; here three of four. With this overlay, our Data Team could start tracing from these high-priority locations and fan outward to map the entire HV grid.

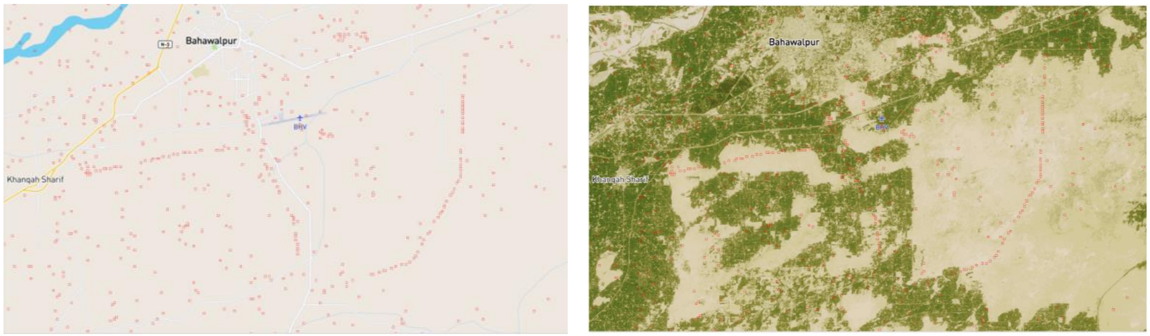


Figure 4. Example of detected HV towers in a high-level view. When viewing the ML detections on top of the map, the strings positive predictions (indicating HV lines) begin to stand out.

Some existing maps of the target countries did exist prior to this project, but in general they lacked quality. Most of the data available tracked a few lines across long distances but missed components of the grid or did not provide accurate locations. There were also some regional maps available, but these were not centralized in a single location. The three target countries under this project included Pakistan (1,009,303 km^2, 50 million zoom 18 tiles), Nigeria (927,886 km^2, 40 million zoom 18 tiles), and Zambia (777,773 km^2, 34 million zoom 18 tiles). The aim was to map HV infrastructure in these countries, but also to develop a reusable framework to reduce the time and cost associated with any future mapping efforts.

METHODOLOGY

The processing pipeline

Generating country-wide maps involved several key steps outlined below and discussed in detail:

1. Download imagery for Pakistan, Zambia, and Nigeria after obtaining country boundaries and computing tile indices.
2. Train a machine learning model to compute the probability that a HV tower was present. Then, apply the model at a country-wide scale.
3. Compile the highest probability results into a GeoJSON map overlay that indicated the most likely HV tower locations.
4. Trace HV infrastructure (specifically towers, lines, and substations) using the ML-derived overlay to focus on high-value areas.

1. Download imagery

To find all satellite imagery for each country, the country boundaries were first downloaded from an online database of Global Administrative Areas. As Pakistan has an ongoing border dispute, it was confirmed that those borders matched those matched the internal records of the World Bank. Then the indices of every satellite imagery tile that overlapped each country's borders at a specific zoom (or spatial resolution) were calculated. Tile indices simply consist of 3 numbers: X, Y, and Zoom coordinates reflecting the spatial location and pixel resolution of that tile. All relevant tiles for each country were obtained using a depth-first search algorithm: this algorithm kept a queue of tile indices stored in a last-in-first-out stack. At each iteration, the top tile was removed, it's spatial boundaries were computed, and the algorithm checked for spatial overlap between the tile boundaries and the country's boundaries. If this overlap was nonzero, the algorithm computed the four sub-tiles of the original tile (i.e., zooming in a single increment) and added these tiles to the stack. If, when the algorithm removed a tile index in the queue, it found that this tile both (1) overlapped with the country boundary and (2) was at the specified zoom (here, 18), then it appended this tile index to a text file and deleted it from the queue. Once the queue was empty, the algorithm terminated leaving a text file with all tile indices at a specific zoom that cover the country boundary. The advantage here was that it could process millions of tiles without ever storing more than 50 tile indices in the computer's operational memory (RAM). This algorithm was published as open source code upon completion of this project.

2. Machine learning

The code and model are available on Github.

With the imagery downloaded, the next step was to autonomously investigate these images for HV towers. A probability score was generated on the interval [0, 1] that a HV tower was present in each raster image using a convolutional neural network (CNN). The CNN took raster images as input, and produced a single probability per image as output. Specifically, the XCeption neural network (Chollet, 2016) was used as it has shown excellent scores on the ImageNet benchmark test set and has relatively few parameters relative to comparable CNNs (e.g., VGG and Inception). The latter fact implies faster training and prediction run-times simply because there are fewer point computations per image.

The XCeption model was trained using three small datasets -- one from each of the three target countries (Figures 5-7). The Data Team manually validated all tiles in these datasets and incorporated any required changes into OSM. This involved checking every meter of ground so that we were confident our training dataset did not miss any towers or include false positives, which would hurt our ML model performance during the inference stage (i.e., predicting on a country-wide image set). For the training data, only the Digital Globe Vivid layer was used as this was the imagery that was accessible for full country prediction. Then the actual data was constructed using Label-Maker – an open tool built previously by Development Seed to rapidly construct ML-ready datasets from OSM and an imagery source. The training procedure leveraged transfer learning: the XCeption model was initialized in Keras using weights from ImageNet training. Afterwards, only the top layer was re-trained for 2-5 epochs so that it would output the probability of two classes (HV tower present or absent), and finally opened up training to all layers to fine tune the entire model.

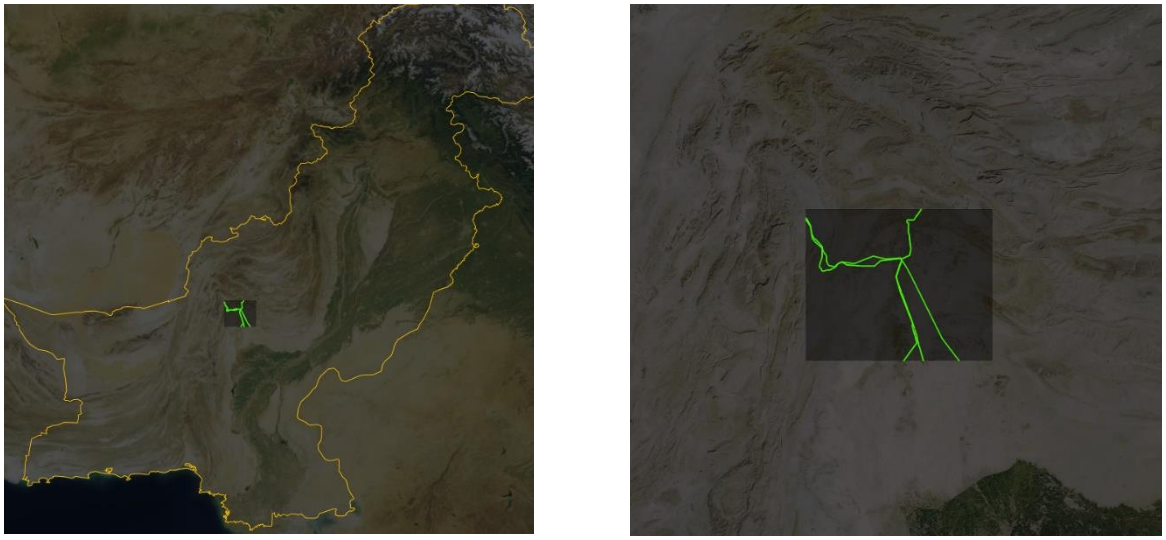


Figure 5. Training region for Pakistan. Green features represent HV infrastructure included in the training data.

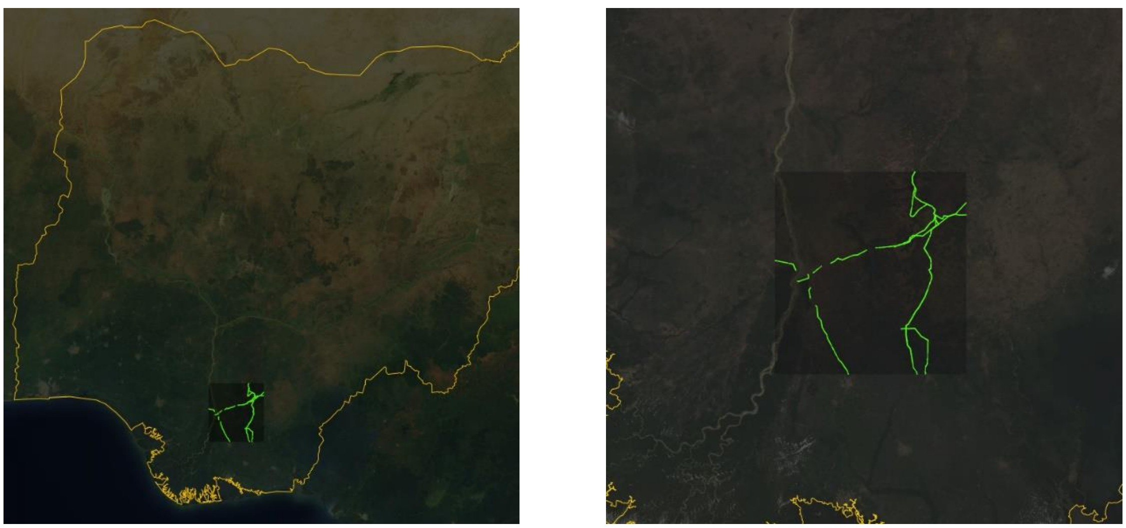


Figure 6. Training region for Nigeria. Green features represent HV infrastructure included in the training data. There are some broken lines as HV infrastructure was only included if it was visible on the Digital Globe Vivid layer.

Figure 7. Training region for Zambia. Green features represent HV infrastructure included in the training data. There are some broken lines as HV infrastructure was only included if it was visible on the Digital Globe Vivid layer.

The binary cross-entropy loss function was used in training and generally trained over 15-30 epochs of the total training dataset. Also, the Keras's utility functions were used, including early stopping and automatically reduced learning rate when validation loss hit a plateau. The dataset was augmented by using Keras's image preprocessing functions to randomly flip, rotate, and scale original images during the training procedure. Results were visualized with Tensorboard to monitor model progress. For hyperparameters, the Hyperopt library was used to intelligently iterate over different hyperparameter combinations. The full list of hyperparameters is available in config.py, but generally this included options like the optimizer, learning rate, initialization strategy, etc. An optimal model was selected based on the highest accuracy score achieved on the held-out testing data across all Hyperopt iterations.

All country-wide predictions were carried out in the inference stage on AWS EC2 instances optimized for GPU compute. Specifically, the p2.xlarge and p2.8xlarge instance types were used and were ran these as Spot Instances (as opposed to On-Demand) to reduce AWS costs. Typically, 2-3 instances were ran at a time to increase throughput. On each instance, large batches of images were processed one at a time. Each country was generally divided up into about 7-10 batches (according to the X-indices of tiles) as it was not feasible to transfer an entire country's worth of tiles from S3 to the EC2 instance. While computing probability scores, the prediction script periodically uploaded tile prediction scores (as a JSON file) to S3 while processing (around every 5,000 images). This ensured that the prediction was fault tolerant -- scores were not lost when the instance shut down unexpectedly.

3. ML-derived overlay

With the probability of a tower computed for each tile, the next step was to compile these results into a GeoJSON and provide this as a map overlay that the mappers could use on top of their satellite imagery. Therefore, a hard threshold for the probability score was selected: any image with a score at or above this limit would be designated as having a HV tower by the model and incorporated into the GeoJSON overlay for our Data Team to use during mapping. To choose this threshold, Receiver Operator Characteristic (ROC) curve was computed and the distributions of scores for positive and negative examples was plotted (i.e., containing or not containing HV towers). These tiles were included in a GeoJSON map that simply marked a square outlining every positive prediction. The threshold was set at 0.97 in an attempt to balance true and false positives. As in the previous inference step, these GeoJSON overlays were computed in batches according to the X-indices of tiles. Finally, all overlays were concatenated into one file using geojson-merge and uploaded to S3 for the mapping team to download and use within JOSM (Java OpenStreetMap; an OSM editor).

4. Tracing

Using the GeoJSON overlay, the data-team began to trace all HV infrastructure. They first created a “task” for each country using a [tasking manager](). This tool breaks a geospatial area into a grid of small squares, where any person can only work on mapping a single small region at a time (to prevent collisions; Figure 8).

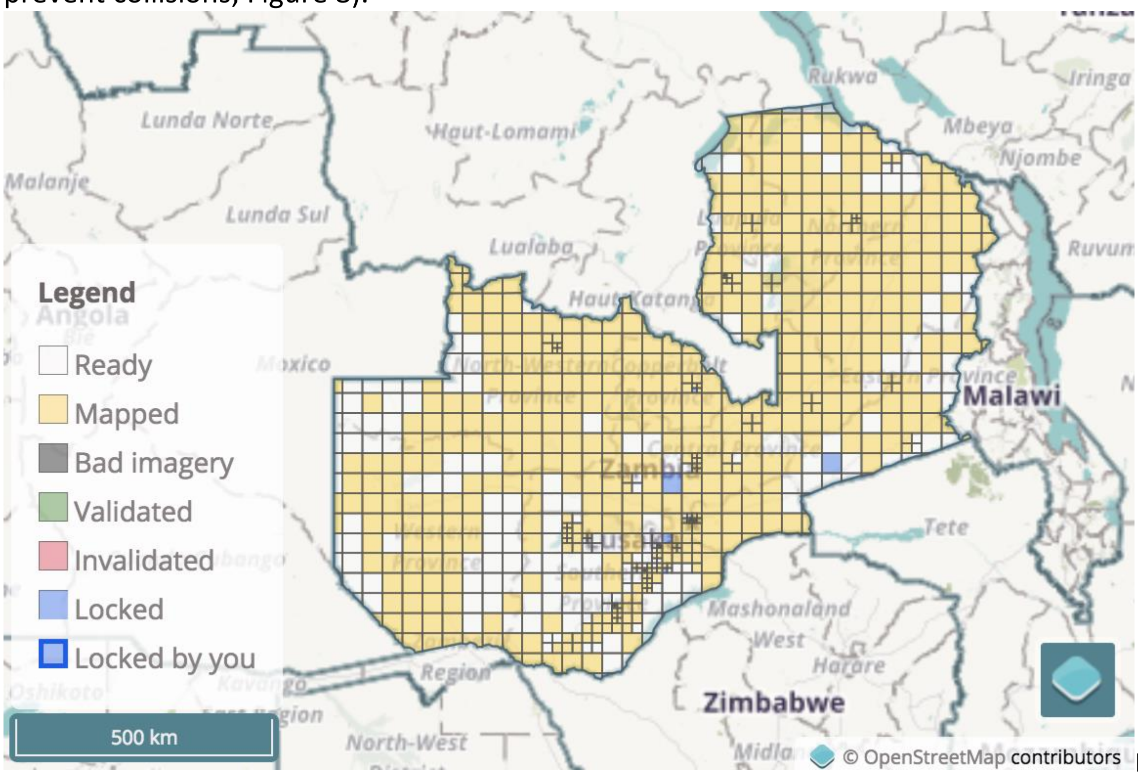


Figure 8. Example of tasking manager in Zambia. Each yellow square represents a small region that was mapped and each blue square shows a locked region where mapping in progress by a single person. Squares without any color still require attention.

Each person selected and “locked” a square before mapping all HV features that lie within this small zone using JSOM. Each square was approximately 1,100 km^2, which often contained many predicted towers distributed throughout the area (Figure 9).

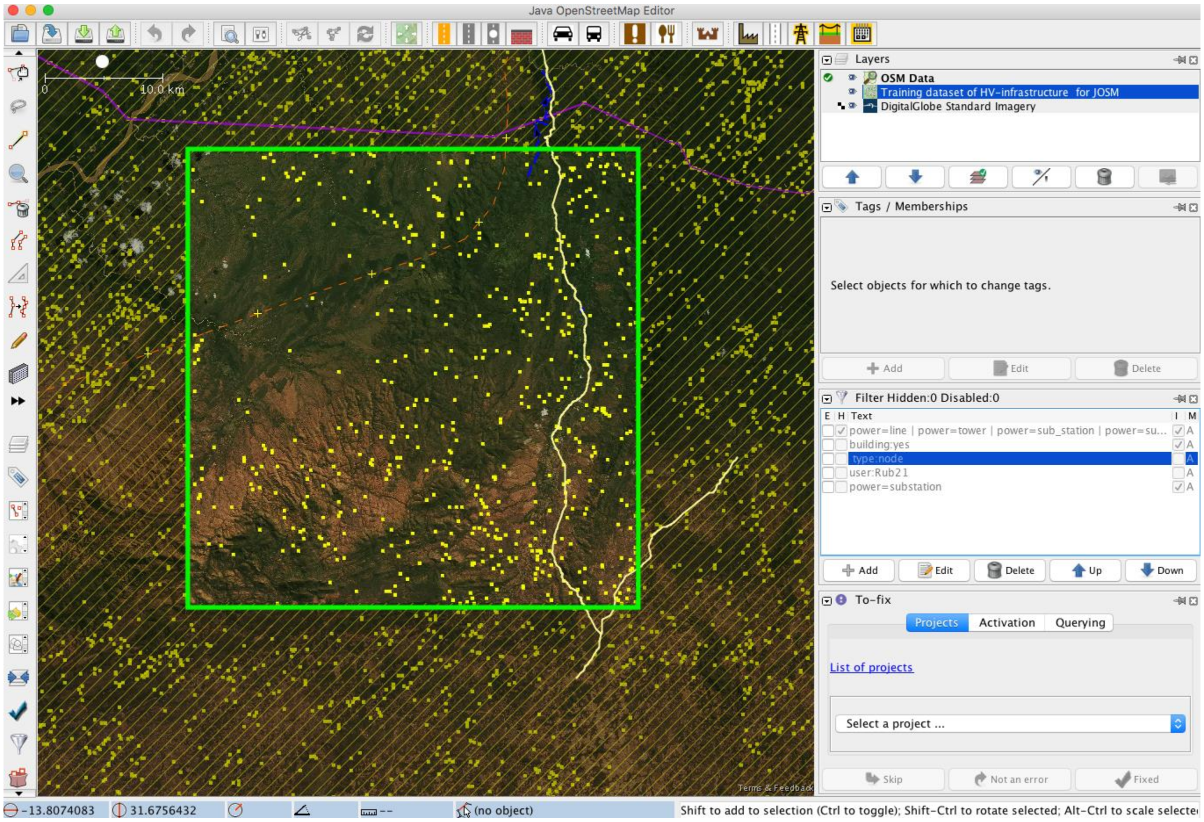


Figure 9. Example of a single locked tile from the task manager. Each dot represents a positive prediction by the model ready to be manually reviewed.

Nearly all of the manual tracing involved tracing from tower to tower. Again, the ML-derived overlay acted as a guide -- HV towers that were correctly identified showed up as strings of small boxes that are visible against the unordered background of false positives. The Data Team updated the HV infrastructure within their mapping software each morning to make sure they always edited the most up-to-date version. Tracing primarily involved adding sections of the HV network that were completely missing or adding missing towers along an existing but under-mapped HV line. In some cases, the Data Team fixed tower locations that were 50-100 meters misplaced from the correct location. The data team also did not edit the operating voltage tag for the HV lines. If the voltage existed prior to the project start, they left it in place; otherwise, it was not added.

Because it was difficult to keep track of which areas they had reviewed, the Data Team customized the [To-Fix plugin](#) available in JOSM. Rather than manually zoom and pan between each predicted tower, this allowed them to load in the ML-derived prediction map and iterate over individual predictions one at a time with the click of a button. This helped organize the review process and increased the speed of each mapper. The Data Team was tasked with tracing tower and power substation they could find (even if it was not identified in the ML portion of the pipeline). The GeoJSON overlay acted as a guide so that the mappers could efficiently hook into a section of the network and begin mapping, but it did not dictate all their mapping efforts.

The Data Team went back and validated all edits once their initial mapping pass was complete. The goal was to double check for any missing connections and verify the accuracy of all added features (e.g., HV towers and substations). As edits were made within OSM itself, all additions were immediately available for anyone to view and access. Then final statistics on the mapping rate (in of km^2 per hour and towers per hour) during the generation of training data and during country-wide prediction (i.e., before and after the ML overlay was available) were computed. These two data points allowed to calculate the relative speed of pure manual mapping and ML-assisted mapping.

RESULTS

Optimization: model training

When training deep learning models, there are many small choices to make regarding the architecture and training procedure to follow. Typically, a good strategy is to test a wide range of hyperparameters and select the model with the highest performance. The full list of possible hyperparameters tested is available in config.py, but some of the most important examples include the optimization scheme, learning rate, and non-linear activation function. To automate testing process, the Hyperopt library was used and to track each model's performance in Tensorflow's Tensorboard -- a tool for visualizing model performance during training (Figure 10). About 150 different iterations of the Xception model were tested during the early stage of this project and selected the highest performing one.

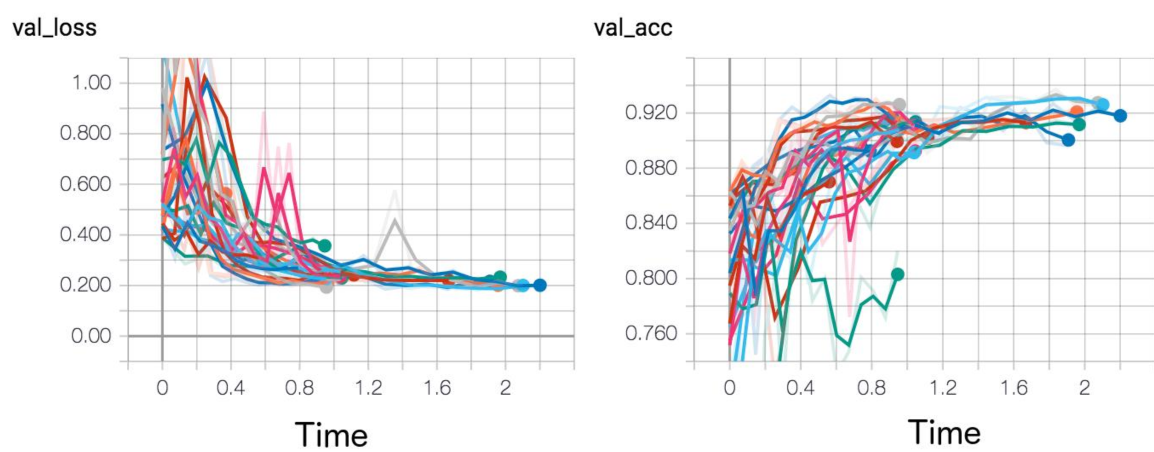


Figure 10. Validation loss and accuracy during training. Each line represents the training of one model over time viewed in Tensorboard. Left: loss on the validation data (using binary cross-entropy on the probability output) where lower is better. Right: accuracy on the validation data as proportion of images correctly classified. Here, higher is better.

Optimization: signal detection threshold

The ML model took individual images as input and provided output for each in the form of a probability between zero and one. This probability score represents the model's confidence that a HV tower was present. Therefore, this problem was framed with a perspective borrowed from signal detection theory. This field provides theoretical guidance on how to select a threshold cutoff when deciding which images are said to contain towers and which aren't. Here, this threshold defines which tiles should be included in the map overlay provided to the Data Team. ROC analysis from signal detection theory was used to inform this choice of threshold; a ROC curve gives insight into how our the true positive-rate (TPR) and false-positive rate (FPR) change for different choices of the threshold. The TPR gives the proportion of tiles truly containing a HV tower that were selected by the ML model. The FPR is the portion of model selections that were incorrect (i.e., probability of a false alarm). An area under the curve (AUC) of 0.96 was obtained in held out data from the three train regions with a threshold=0.31 (Figure 11). The probability score distributions for the held-out data in the training phase show good separation between true positive and true negatives (Figure 12).

Figures 11 and 12 indicated very high performance, but it is necessary to note that this is specific to data from the small training data set (covering about 1.05% of the combined total area of the three countries). When running this model at the country-wide scale, a substantial drop in performance was noted. Reason for this drop and potential ways to ameliorate the issue are explained in the discussion section. Briefly, it is believed that the problem lies with the fact that the training data was only created from one small region in each country.

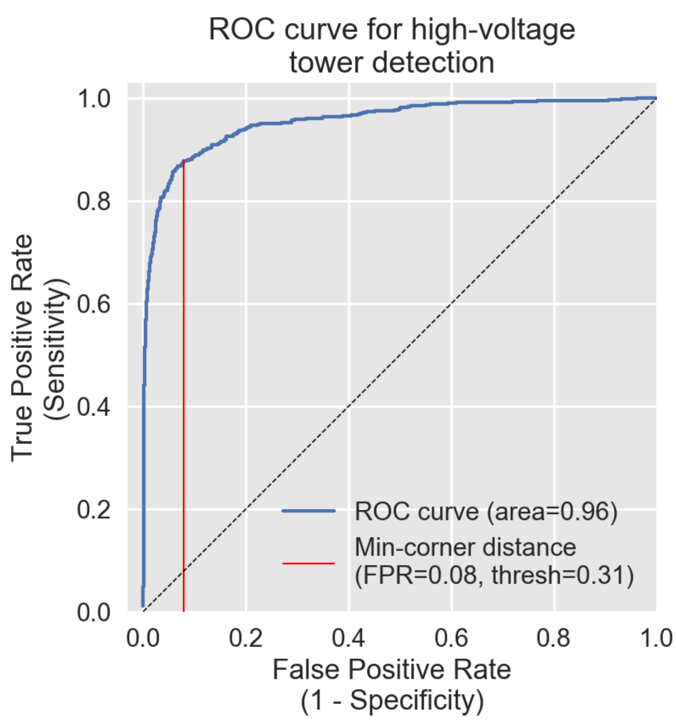


Figure 11. ROC curve for model's performance on detecting HV towers. Setting the threshold of 0.31 gave optimal corner distance.

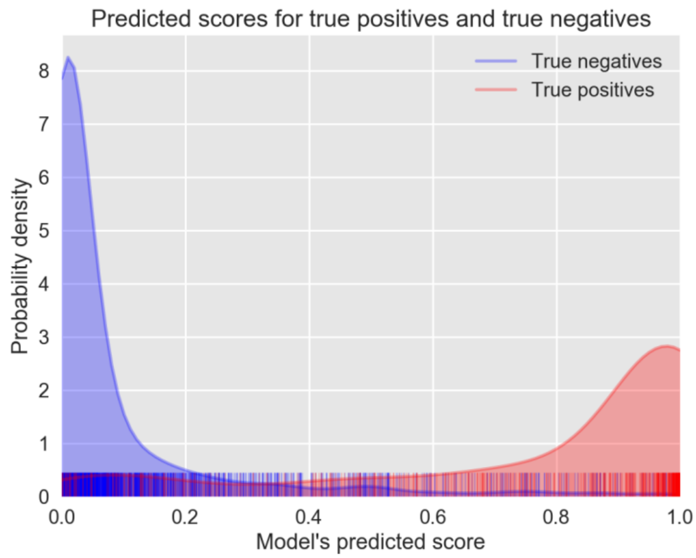


Figure 12. Assigned probability distributions for images containing (red) and not containing (blue) HV towers. The model's assigned probability scores (x-axis) are displayed as an estimated probability density function (with probability on the y-axis). Note, that the model can properly separate the majority of images in this validation data from the training set.

Country-wide predictions

After selecting an optimal model and decision threshold, country-wide imagery sets were processed across Pakistan, Nigeria, and Zambia (Figures 13-15). It is necessary to note that the thin snaking lines in the below plots indicate where HV lines exist. The same model was used across all three countries, and the model performed much better in regions of each country where the original training data was collected (likely because of the similar terrain). Outside of these areas, the model gave more false positives (appearing as large dark blobs).

Figure 13. Country-wide prediction map in Pakistan. Black dots correspond to locations where the model predicts a HV tower was present. In the mountainous desert region of Pakistan, the model performed relatively well. In the Eastern agricultural region, the model predicted many more false positives as there were no any training data obtained from that region.

Figure 14. Country-wide prediction map in Nigeria. The model made many false positives along the central savannah region of Nigeria. Here, the landscape is mostly comprised of exposed rock outcroppings and sharp hills. These rugged areas have many long cracks and grey-colored rocks that likely contributed to the false positives here. The training data came from a very different terrain -- the tropical rain forest region in the Southeast.

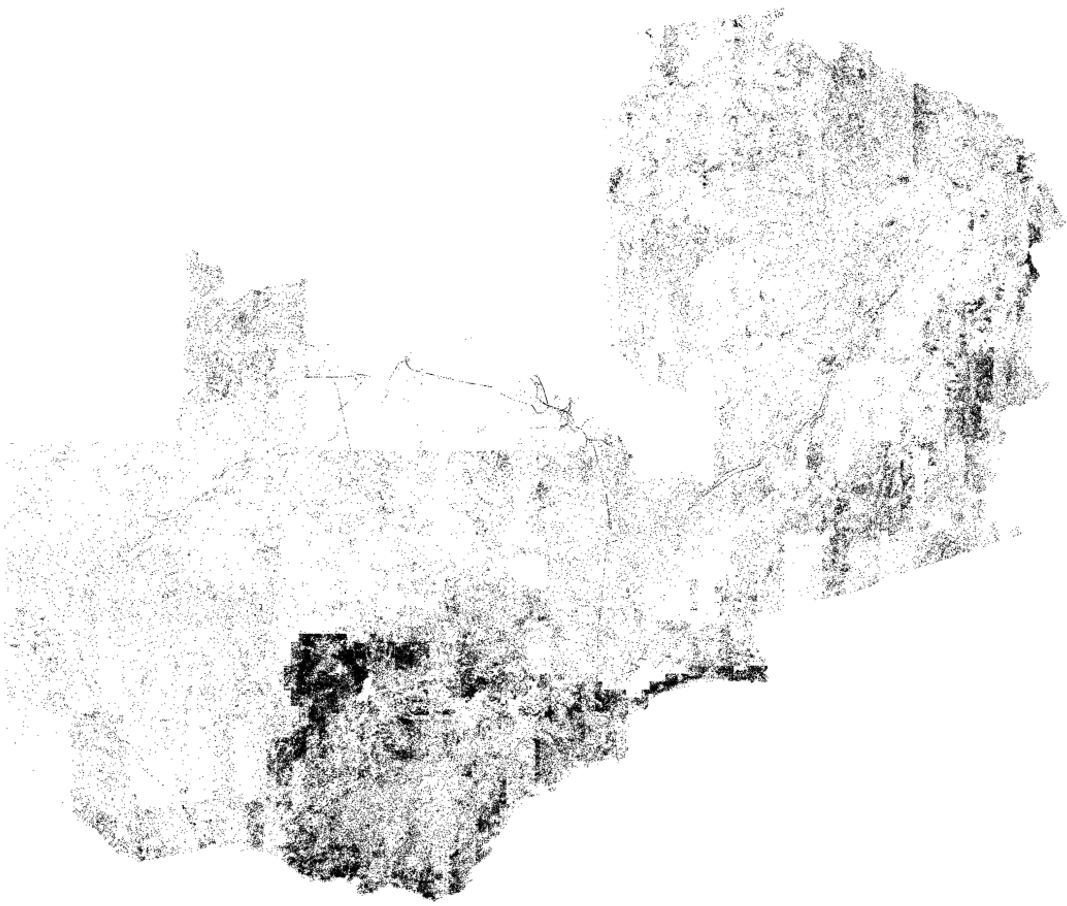


Figure 15. Country-wide prediction map in Zambia. The model performed relatively well in Zambia. In the Southern region, however, the imagery quality was relatively poor (i.e., somewhat discolored and blurry) compared to the rest of the country causing more false positive predictions. This is often indicated by sharp boundaries in prediction density. The effects of wildfires are also visible in some regions where the model performed poorly. This likely occurred because the training data did not sample from these burned regions.

Mapping output and speed

This project substantially expanded the amount of mapped HV infrastructure (Figure 16). Edits include both additions and major corrections though most edits were additions.

The Data Team was approximately 33.4x faster per km^2 with the ML-derived overlay for Pakistan, Nigeria, and Zambia (Figure 17). The total time spent on mapping and validating each country using these ML predictions is in Table 1.

Figures 18-20 show before and after maps of every country. Click on each to interact.

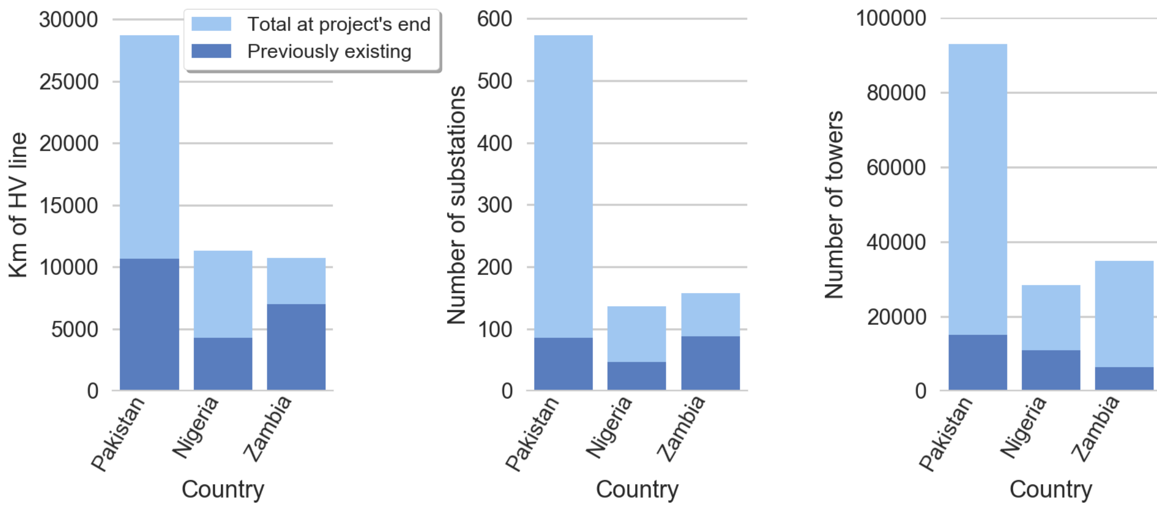


Figure 16. HV infrastructure features in OSM. Left: Total length of HV line after this project. Middle: Total number of HV towers after this project. Right: Total number of substations after this project. The bars in all plot are split into the amount of previously existing HV infrastructure (dark blue) and infrastructure added or edited during this project (light blue). In all cases, most edits were additions.

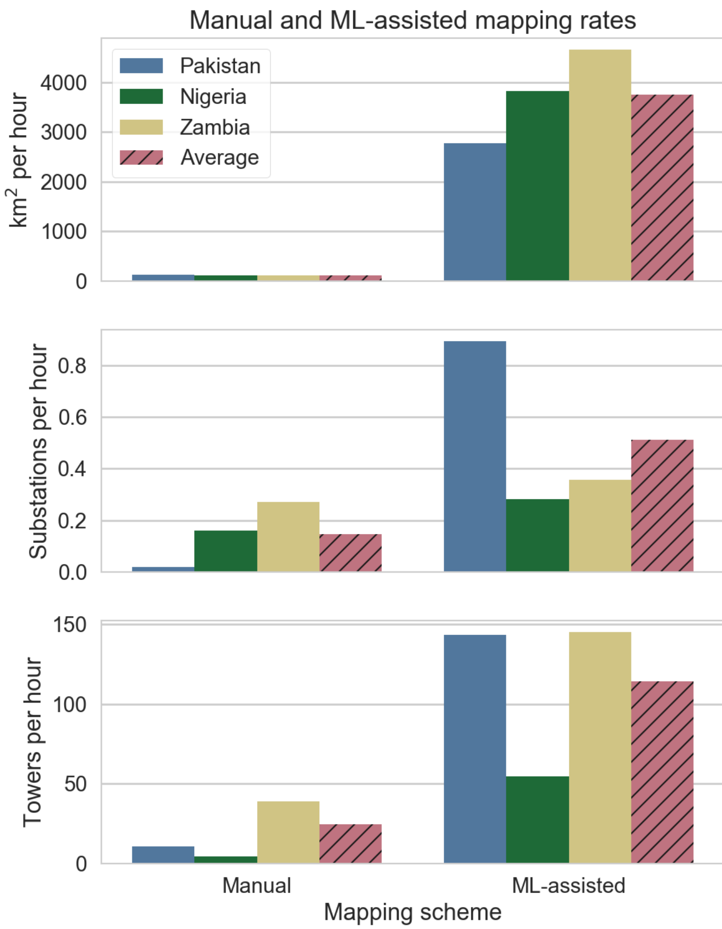


Figure 17. Mapping rate with and without ML-assist. On average, the km² per hour mapping speed increased 33.4 times, mapping speed of substations per hour increased 15.9 times, and towers mapping speed per hour increased 9.7 times. Note that these figures exclude hours spent validating (i.e., double-checking) edits. This ensured a fair comparison as no double-checking was carried out during the pre-ML mapping work to generate the training data.

Table 1. **Total person-hours spent mapping at country-wide scale.**

COUNTRY	HOURS MAPPING	HOURS VALIDATING
Pakistan	364.13	181.09
Nigeria	243.12	74.47
Zambia	167.17	29.45

Interactive demonstrations

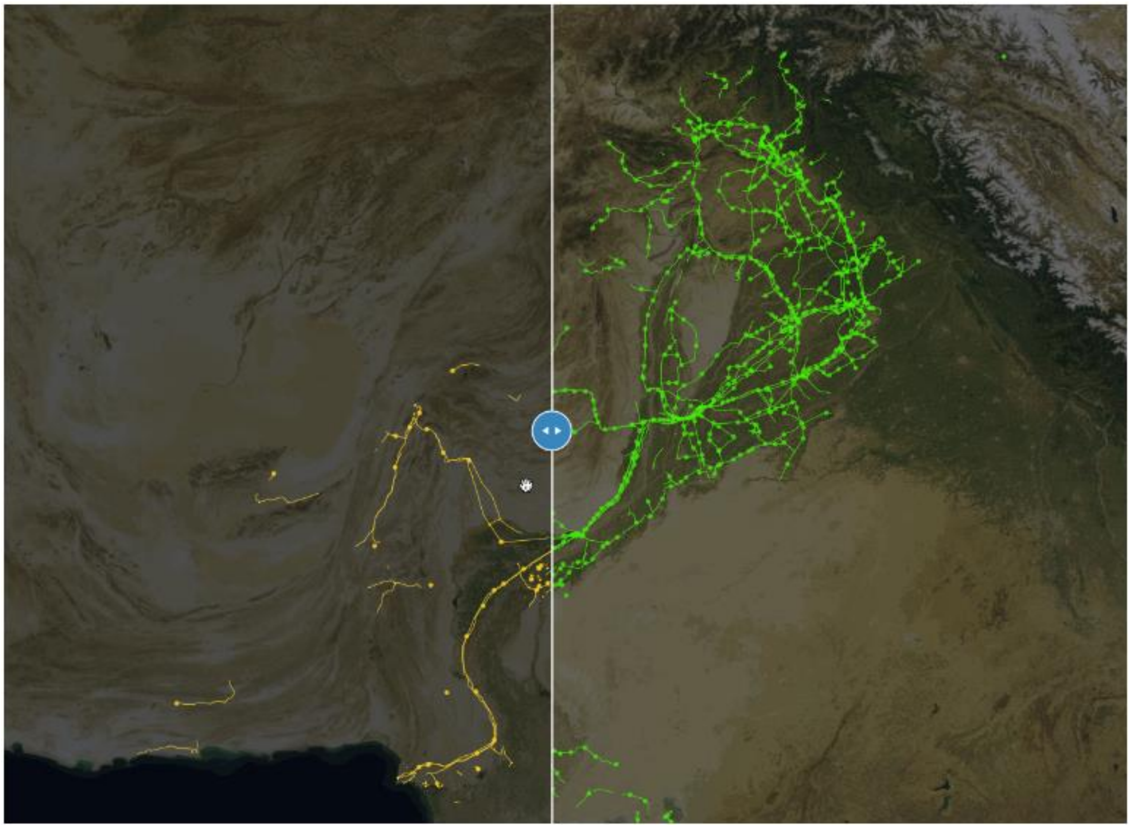


Figure 18. Before and after mapping in Pakistan. High-voltage infrastructure before (yellow) and after (green) this project was complete. Click http://devseed.com/ml-grid-docs/results/mapping-output-and-speed/ for interactive demo and zoom in to see individual towers (dots) and substations.

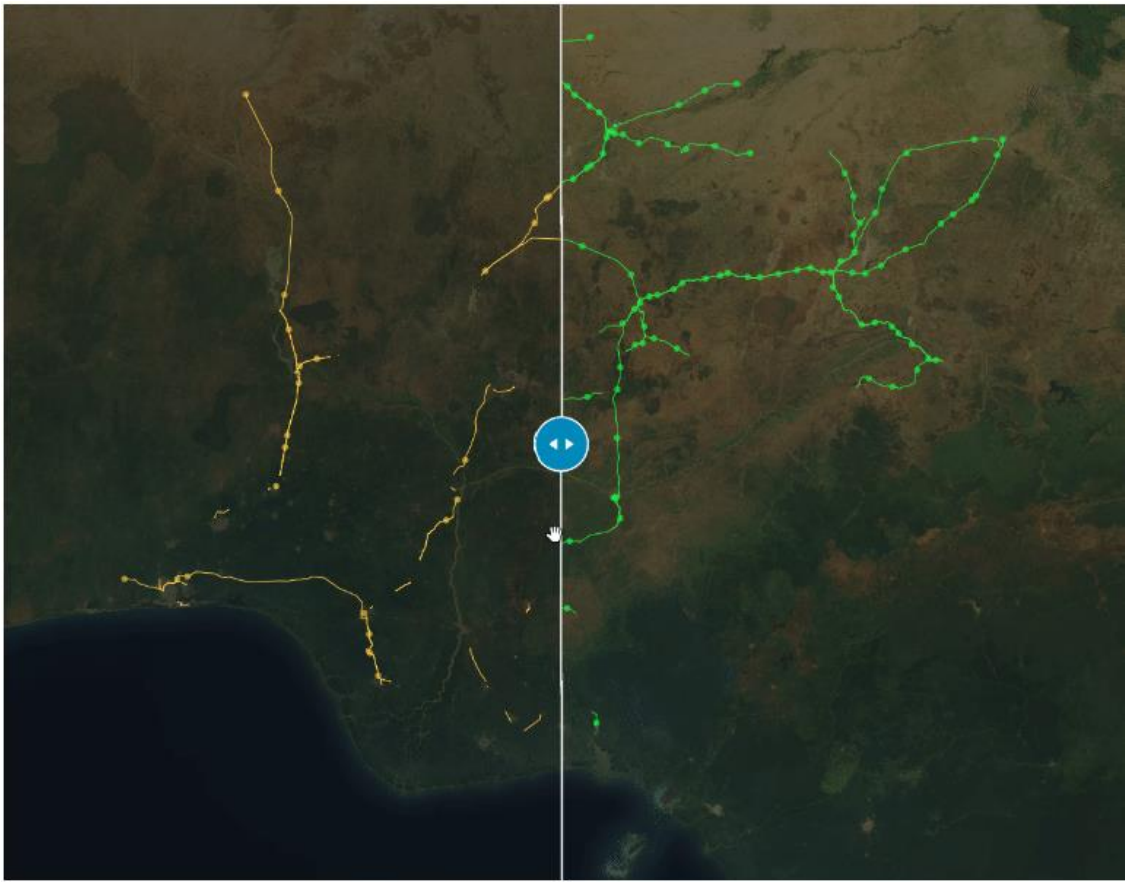


Figure 19. Before and after mapping in Nigeria. High-voltage infrastructure before (yellow) and after (green) this project was complete. Click http://devseed.com/ml-grid-docs/results/mapping-output-and-speed/ for interactive demo and zoom in to see individual towers (dots) and substations.

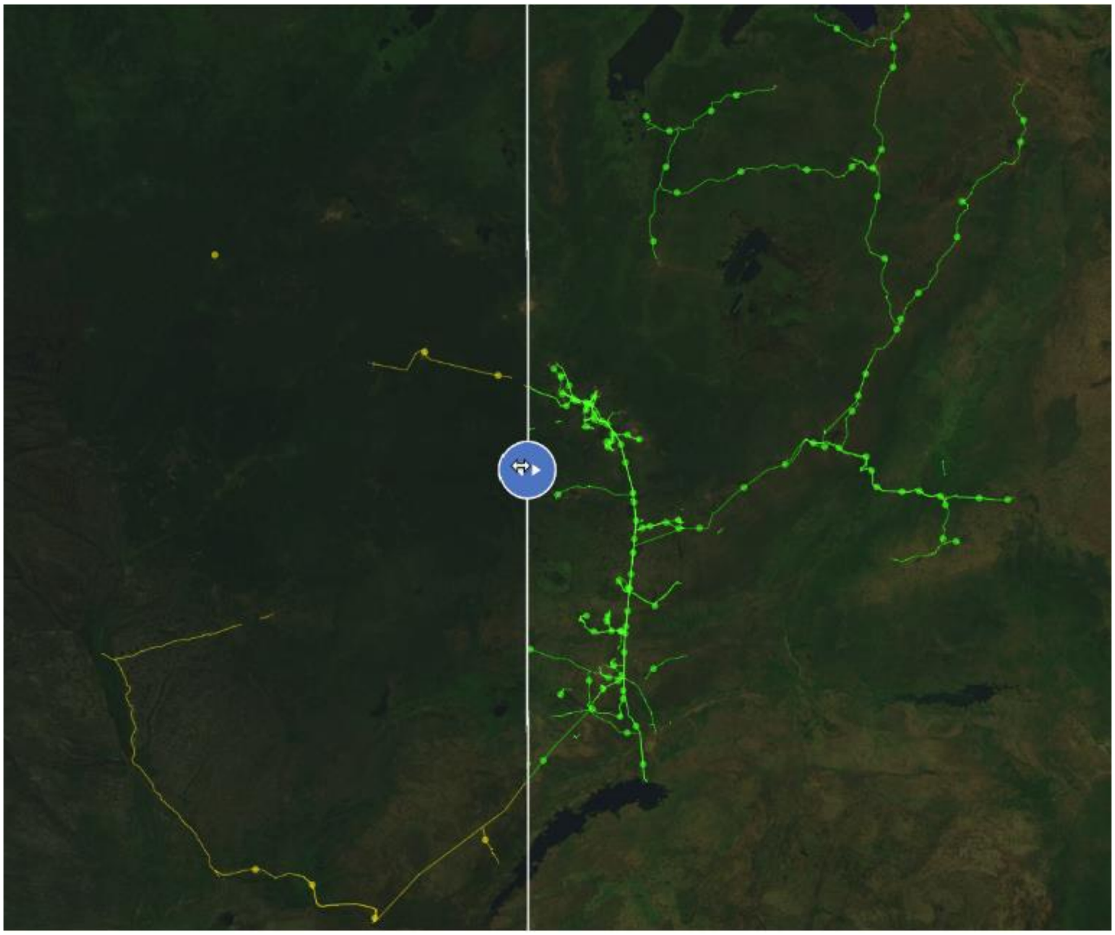


Figure 20. Before and after mapping in Zambia. High-voltage infrastructure before (yellow) and after (green) this project was complete. Click http://devseed.com/ml-grid-docs/results/mapping-output-and-speed/ for interactive demo and zoom in to see individual towers (dots) and substations.

Compute costs

All computation and data manipulation was carried out on Amazon Web Services (AWS). The AWS Elastic Compute Cloud (EC2) was used for the model training and prediction. Primarily, the p2.xlarge instances were used, which contain GPUs suitable for deep learning. To download the satellite imagery from the Digital Globe Maps API, m5.xlarge and c5.9xlarge instances were used, as these could handle large numbers of threads making the necessary HTTP requests. To store the data, AWS Simple Storage Service (S3) was used. Note that the below costs do not include any associated imagery costs, as those are difficult to estimate because of the unique contracts each organization makes with satellite imagery providers. The total compute cost was \$1,137.85 or about \$0.42 per 1,000 km^2. A more-detailed compute cost analysis is available in Table 2.

Table 2. Costs for obtaining, storing, and processing satellite imagery

SERVICE	COST
AWS EC2: downloading imagery	$59.38
AWS EC2: training and predicting	$392.14
AWS S3: storing and accessing imagery	$686.33
Total	**$1137.85**

DISCUSSION

A pipeline was created to boost human mapping speed by 33x when tracing high-voltage infrastructure. At a high level, machine learning was used to find satellite imagery tiles that were most likely to contain HV towers and then pass this information to Data Team – a group of professional mappers to trace the HV infrastructure. This strategy was tested in three countries: Pakistan, Nigeria, and Zambia comprised of a total land area of approximately 2.7 million km^2. All edits were made in OpenStreetMap, which is openly available. Individual changes to OSM are also available in the Github repo.

Throughout the course of this project, it was confirmed that neither humans nor an automated system alone are currently a feasible approach for mapping HV infrastructure. On one hand, professional mappers are very accurate and know when to ask for confirmation on difficult imagery. However, the time required to manually review an entire country is tremendous. We estimated it would have taken our Data Team about 6 months of full time effort to complete Pakistan alone. On the other hand, ML algorithms can operate with very high throughput and very little oversight once trained. Pakistan required only several days of computation time and a few hours of human effort to monitor the scripts. Nevertheless, the ML results indicated that it would be practically impossible to train an algorithm as accurate as a human. Combining them in an Intelligence Augmentation (IA) approach leveraged the strengths of both humans and machines.

The IA approach is also prudent in comparison to a pure AI strategy focused on completely replacing humans. By keeping a human in the loop, it can be made sure that all ML predictions are validated by a professional mapper before the additions are incorporated into OSM. The OSM community is (rightly) skeptical of any method that add edits without human verification as this strategy has led to issues in the past. Therefore, building a workflow utilizing the strengths of both is likely the optimal way forward. Future work should focus on improving the machine learning predictions and better incorporating those predictions into mapping editors (perhaps as plugins for standard map editors) so they are widely available to human mappers.

Future directions

The rest of this discussion section is focused on improvements for future iterations of this mapping pipeline.

1. **Improving how we handle big data**
    1. Efficiently downloading and storing large imagery datasets
    2. Matching download and prediction speeds on the fly
2. **Improving the machine learning predictions**
    1. Detecting HV substations
    2. Improving HV tower detection
    3. Using additional forms of imagery (like SAR)
3. **Integrating ML predictions into human mapping workflows more effectively**

1. Big data

Downloading large datasets

Perhaps the biggest challenge in this project was handling the tens of millions of images for each country. HV towers only become recognizable at about zoom 18 imagery (~0.5 meter/pixel resolution), which is a relatively high spatial resolution for commercially available satellite imagery. The cost associated with this high spatial resolution was associated with the need to handle a very large volume of tiles. Just the act of downloading the imagery from the Digital Globe Maps API was extremely computationally intensive. Tens of thousands of networking coroutines were run, essentially lightweight threads each downloading a single image at a time, in parallel. This was the only method allowing to obtain the country-wide image sets in a reasonable amount of time. It was also required to keep track of which files had been downloaded to avoid wasting time and resources; at nearly a hundred million image tiles, tracking this process is no longer trivial. Standard operating systems would throw an error if one would simply try to list or delete a folder with this many files.

For the next iteration, AWS's Simple Queue Service (SQS) will likely be used to tabulate all tiles that need downloading. Each entry will contain tile indices, which the downloading script can pull from asynchronously. Then, each entry in the queue is only removed if the downloading process sends back confirmation that it worked correctly. This should provide a fault tolerant and more efficient solution that could also scale if there is a need to process more countries. It will also allow for easier downloading of a portion of the total images. In many cases, the ML algorithm does not need to predict the entire country's images for the strings of HV towers to become visible to a human in an ML-generated map overlay. In the first iteration, the download script randomly skipped over images with a probability dependent on the desired download proportion (i.e., if only half of a country's images were needed, the skip probability would have been set to 0.5). Using SQS, allows randomly shuffle the queue once and then download the desired number of images directly instead of wasting computation in this skip procedure.

Optimizing download and prediction speed

It was also decided to download all imagery directly to AWS S3 for storage. This choice was useful in that there never was a need to request any imagery twice from the Digital Globe Maps API, and a full copy of all the data intended to be processed was available. It was also believed that it was possible to rapidly transfer images from S3 to EC2 instances for the inference stage, which was dependent on GPU instances that are charged by the length of time that they're reserved.

However, the storing and accessing data on S3 was both costly and slow. There are set financial costs associated with the raw volume of stored data as well as costs for each file uploaded or downloaded to and from S3. But accessing stored data was more expensive than expected simply because of the sheer number of files that were required to process. Additionally, the speed at which the stored data could be accessed was much slower than expected. This likely occurred for two reasons: first, since S3 accesses files using a key-based system, any request for a subset of imagery (using asterisk wild cards for example) required the AWS servers to iterate over all keys to find the relevant subset; again, because of the large volume of files, this was surprisingly slow — in some cases, an S3 copy request could take 1-2 hours before the download even started. Second, the effective download speed (in Mbps) was also quite slow. It became evident, that there is a fixed overhead computational cost to initiating the download of a single file. As a practical example, this means that downloading one-thousand 1kb files is much slower than downloading one 1 Mb file. Transferring data from S3 to EC2 instances (for inference) was about 100 times slower than AWS is capable of, had the same data been stored as a single file.

Future efforts should attempt to avoid S3 and focus on methods of downloading imagery directly to EC2 instances immediately before prediction. The challenge here is to match the speed that images are downloaded with the speed of prediction. Currently, the download step is 2-5x slower depending on the GPU instance used for prediction. Download speed could be increased by finding a better method of requesting images (currently done via HTTP). It may also be possible to somehow construct super-images each made up of 25 or 100 individual tiles (in 5x5 or 10x10 squares, respectively) prior to downloading them. This would address some of those constant overhead and possibly reduce the effective per-image download time. Even something as simple as combining large groups of images into single zip files might help reduce the slowdowns we experienced.

2. Improving ML output

Detecting substations

Substations are a part of the transmission and distribution system. Their purpose is to transform voltage from low to high prior to electric energy transmission and vice versa to supply transmitted electricity to consumers. The location of these substations is useful when mapping the HV network since most HV towers end or begin at these points. The Data Team estimated that they would be 15-20% faster with substation locations. It is also valuable information to developers focused on renewable energy projects. The location of substation is vital when planning potential connection points for the next generation of electricity-producing infrastructure.

For these reasons, autonomous detection of substations is an important goal for future HV mapping work. Substations were mapped as part of this project, but the ML model was not built to explicitly detect them. The Data Team identified substations while tracing HV lines to their end points, since they generally terminate at a substation. Building an ML model explicitly capable of detecting substations would accomplish two major goals. First, it would facilitate the mapping process as these substations represent important hubs within the grid's network. Knowing their location is especially useful in crowded urban environments or where HV lines run underground near residential areas. Second, substation detection would provide a much-needed tool for integrating renewable energy projects into the larger electricity network. The data acquired while mapping around thousand substations across the three full countries, it will be possible to train a ML model capable of automatically detecting substations.

Better detection of HV towers

In the beginning of this project, there was no any validated training data. The Data Team was tasked with manually reviewing three relatively small areas — one region in each country — and ensuring that every meter of ground was checked for the presence of a HV tower. This was a slow task (and exactly the problem to be solved), but necessary to generate a training set of imagery to build the machine learning model. These initial training datasets covered about 1.05% of the total area that would eventually be processed when moving to the country-wide scale.

Having the complete high-voltage infrastructure mapped in three full countries, and a vastly larger set of data would allow to train the next iteration of the model. Future iterations should start by adding training data from where the model performed most poorly. For example, the agricultural region of Pakistan is a great place to start as the gridded farmland regularly confused the ML model. In this iteration of the project, the only training data in Pakistan came from the desert mountainous region in the Western half of the country. This is likely the reason that the model generalized poorly to the more lush farmland along the Eastern border. The model was also confused in certain regions where there were issues with DG's imagery including shading or blurriness. Again, having more data with these issues present will provide training data that better represents the true data distribution and ameliorate the problem. In the future, a simple strategy might be to build training imagery from numerous small regions, representing most or all of each country's different terrain (as opposed to a few large regions as done initially). This will result in training data sets that better capture the full distribution of possible images the model might encounter when processing across an entire country.

SAR imagery

In this project optical (i.e., RGB) imagery was used to make all predictions. The initial pilot phase of this project, however, involved synthetic aperture radar (SAR) imagery – an active sensing technique that emits and receives microwave energy. Man-made structures like HV towers strongly reflect SAR imagery making them visible even in low-resolution imagery. This imaging technique is also advantageous in that it is not affected by clouds or shadows. However, SAR suffers in mountainous and forested regions because the technique measures gradients in surface height. While SAR imagery is not expected to replace the current optical approach, it may act as a power tool to validate portions of the network with high confidence.

3. Improving mapping workflow

Machine learning aside, there is room for improvement on the human component of the mapping efforts carried out by the Data Team. The concept of using machine learning to guide mapping was new and untested for the professional mappers and the ML team, so ideas were collected to streamline the process throughout this project. The Data Team found that navigating through the ML model's predictions was tedious — initially they were repeatedly zooming in to map a few towers and then zooming out to reestablish a high-level view of the predictions. This constant reorientation of the view also made it difficult to keep track of which areas had been reviewed. As discussed in the Methodology Section, the Data Team built its own To-Fix plugin within JOSM so they could click through the model's predictions with a single button. In the future, the potential to "jump" several towers at a time when mapping can be explored if only the HV lines are of interest. This could reduce mapping efforts up to 30% in areas where multiple towers in a row are visible.

Another limitation was that the ML predictions were not available to view in the tasking manager – the piece of software used to organize a group of mappers working in the same geospatial area. Initially, there was no way to view the ML predictions in this overlay, but the team could create the ability to add the ML overlay just like other map layers.

The Data Team also noted that image quality varied across entire countries. In some cases, it was difficult to accurately trace HV infrastructure due to blurry or otherwise poor-quality imagery. Future efforts should explore other sources of satellite imagery that are higher quality and up to date. An engineering consult who is familiar with typical HV network construction might also be advantageous. Someone with domain knowledge should be able to rule out ambiguous features in the imagery; examples include connections that appear to involve three HV towers, changes in the size (and possibly operating voltage) of HV towers, or areas where the HV lines appeared to bifurcate. Areas where many different HV lines came together (often near substations in cities) were also difficult to delineate as were towers in complicated urban environments. These special cases were relatively rare, but they are difficult to map from overhead imagery at zoom 18 resolution.