AP-Latam Technical Report Q1

ap-latam.dymaxionlabs.com

July 2018

1 Introduction

Poverty is a major issue in Latin America. Slums and informal settlements grow quickly, and there is a need for more up-to-date data for policy maker decisions. If governments had updated information about slums and their growth, they could give the affected families better life conditions and change their future. Having updated information greatly helps to improve health, education and security for the children who currently live there.

In order to carry out an accurate survey of precarious areas, exhaustive coverage of the territory is required. This task requires having costly logistical and material resources, which in turn undermines periodicity and scope.

Our solution, AP-Latam[4], tries to reduce survey costs by analyzing high resolution satellite imagery to detect potential areas of informal settlements growth. The end result is a geospatial dataset of areas that could contain informal settlements. By having more updated potential slums growth location, survey users can prioritize areas to cover and make better decisions.

In this report we describe our methodology and the application of our solution to an extensive area of Buenos Aires, Argentina.

2 Methodology

The method consists of a binary classifier of image tiles. Each image is classified as to whether it contains an informal settlement or not.

To build the dataset for training and validation, the classifier takes a vector file of polygons of previously-known informal settlements, and takes fixed-size tiles of images by sliding a window across the entire satellite image. For each image tile, it checks if the tile intersects with some polygon and tags it appropriately.

To make predictions over new images, it slides a window over the new image and builds a new vector file of polygons of the size of each positively-tagged

(a) Tiles tagged as *true* (b) Tiles tagged as *false*

Figure 1: Example of images tiles tagged as either true or false, whether they contain a informal settlement

image tile. The resulting dataset is post-processed to remove polygons with small probability and dissolve them into bigger polygons.

In our tests with Buenos Aires, as a final post-processing step, we used official open datasets of street blocks[5][2] and crossed them with the polygons file.

2.1 Imbalanced classes

A binary classifier of informal settlements has imbalanced classes, that is, images tagged as positive (areas that contain settlements) are much less frequent than images tagged as negative (areas that do not contain settlements). To decrease bias and avoid overfitting, we subsample the negatives by taking a set of images of size proportional to the size of positives. We tried subsampling with a proportion of 4 and 8 and settled with 4.

2.2 Data augmentation

To help prevent overfitting and make the model generalize better, we perform data augmentation on the image tiles. For now we only do horizontal and vertical flipping, but there are other random transformations that we could apply to augment our dataset, like hue and brightness randomization (to account for differences in atmosphere corrections) and rotations.

2.3 Fine-tuning

A large amount of data is needed to build a functional convolutional neural network model. In practice, it is common to reuse a pretrained network. However,

Figure 2: Example of random transformations over an image tile

most pretrained networks work for a different set of labels, and were not trained with satellite images, so for this use case it is necessary to retrain some of the top layers to improve prediction.

The methodology used here was to fine-tune a ResNet-50[3] network with our satellite imagery. The procedure is roughly as follows:

1. Instantiate the convolutional base of ResNet-50.
2. Add a fully-connected model on top, with a standard SGD optimizer and validating with the binary cross-entropy loss function.
3. Freeze the layers of the model up to the top 70 layers.
4. Retrain the model.

We used Keras library for data augmentation, training and prediction[1].

2.4 Post-processing

The resulting dataset after prediction over sliding windows is a set of small fixed-size squares, with a prediction probability associated. To refine the results we apply the following:

1. **Median filter**: remove squares with low probability and a small number of neighbours.
2. **Dissolve overlapping squares**: if the sliding window step size is smaller than the size of the windows, it may end up with overlapping squares, so this step dissolves them into a single polygon with a mean probability values between the values of each connected squares.

As mentioned before, we also used a dataset of blocks and calculated the intersection between the squares and blocks, and if sufficient squares covered a block, we picked them to form a new dataset of blocks that contain potential informal settlements. The resulting dataset has better prediction accuracy mainly because roads and other areas are not considered.

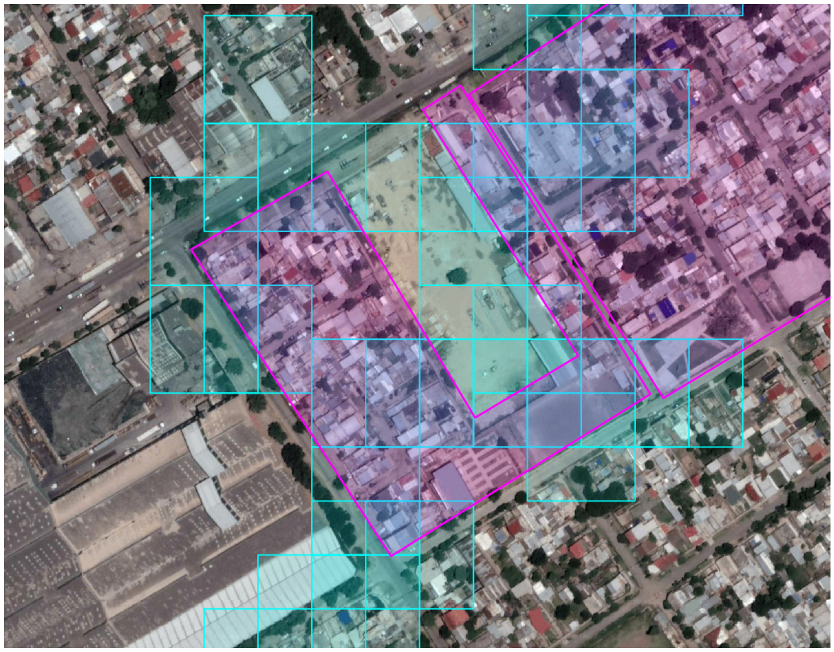

Figure 3: Image tiles detected as positives, and intersecting blocks

3 Results

Results for Buenos Aires were uploaded to our website (both as downloadable data and the online map), and to PANH[6].

These datasets have been released as public domain data, using the same license that OpenStreetMap uses for its data, the Open Data Commons Public Domain Dedication and License[1]. From AP-Latam website the user can download the datasets as GeoJSON files, one for each area and image acquisition date. The user can also explore an online map with the latest dataset generated.

Source code for building and using classifier to create datasets has also been released as open source in a Github repository[2] with a BSD-2 license. Instructions on how to train the classifier and predicting over new images are available there.

4 Other work

We internally tried using Kubernetes in Google Cloud Platform for deploying jobs for training and prediction, which turned to be quite practical for scaling up experiments using multiple instances at the same time. In the future we

[1] https://opendatacommons.org/licenses/pddl/1-0/
[2] https://github.com/dymaxionlabs/ap-latam

will consider using Kubeflow[3], a toolkit for simplifying deployments of machine learning workflows on Kubernetes.

We wrote a Dockerfile and configured Docker Hub for continuous deployment and containerization of the tool. We also started using TravisCI for continuous integration testing.

5 Future work

During the next quarter, we will be working on a monthly report in order to monetize the product.

We have already started developing a different detection algorithm based on semantic segmentation (U-Net architecture), that is known for outperforming sliding-window convolutional networks. Other instance-aware segmentation methods show promising results for detecting building outlines and roads.

Another option to consider is the use of OpenStreetMap data for roads and building types.

References

[1] François Chollet et al. Keras. https://keras.io, 2015.

[2] Buenos Aires Data. Parcelas, ciudad autónoma de buenos aires. https://data.buenosaires.gob.ar/dataset/parcelas.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] Dymaxion Labs. Ap-latam. https://ap-latam.dymaxionlabs.com/, 2017.

[5] República Argentina Ministerio de Economía. Cartografía digital cnphyv 2010 de la provincia de buenos aires. http://www.estadistica.ec.gba.gov.ar/dpe/Estadistica/censo2010/cartografia.html.

[6] República Argentina Ministerio del Interior. Plataforma abierta nacional del hábitat. https://panh.mininterior.gob.ar/h.

[3] https://github.com/kubeflow/kubeflow