Cooper Nederhood
Perspectives III – Spring 2018
May 8, 2018

DRAFT – Data and methodology

In this paper, we estimate ground-truth economic outcomes in sub-Saharan Africa from publicly available satellite imagery. We use modern neural network techniques common in applied computer vision tasks to analyze the high-dimensional satellite imagery. This technique scales well and has relatively fine-grained worldwide coverage, of key importance in data-poor developing economies. Specifically, we test the ability of satellite imagery to detect economic changes through time, rather than just cross-sectionally.

We use 30m resolution 3-band RGB images obtained from the Landsat series of satellites, operational from the 1970's to the present date. While the Landsat series is the gold-standard in publicly available satellite data in terms of quality and broad coverage, the resolution is not high enough to do specific object detection, which is possible with proprietary satellite data purchasable from private companies. However, it is well established that nighttime luminosity, a form of publicly available satellite data, is correlated with economic activity. Nighttime luminosity is a 1-63 integer at 1km resolution, so there is only 1 data point per kilometer. A Landsat image over the same kilometer will have 34x34=1,156 data points for each of its 3 band for 3,468 data points total.

The corresponding measure of the ground-truth economic outcome comes from the Demographic and Health Surveys (DHS) Program. DHS conducts detailed surveys across much of sub-Saharan Africa. While most of the survey questions reflect health outcomes, there is an "asset index" which is the first principal component of a variety of asset-related questions reflecting facts like building material of households, access to technology, etc. This is an ideal construct to measure economic outcomes because variations in the asset index should generally manifest in a different physical environment, discernible in the satellite images. The DHS survey data is conducted at a cluster level which is composed of a random sample of households within the cluster. Our analysis focused on Nigeria, which includes about 900 clusters per survey year. Nigeria has DHS survey data for the years 1990, 2003, 2008, and 2013. The city of Lagos in southwest Nigeria is one of the world's fastest growing cities in the world and is the largest city in Africa, making Nigeria an ideal subject for our analysis. Crucially, the DHS data includes corresponding GPS latitude and longitudes, allowing us to tie the ground-truth measures to the corresponding satellite imagery. The GPS information is contained in shapefiles which I have processed and merged with the tabular survey data.

We train a Convolutional Neural Network (CNN) to estimate the DHS asset index from the corresponding Landsat imagery. CNN's have emerged as the state-of-the-art in computer vision. While a standard neural net has hidden fully-connected layers, CNN's contain convolutional layers in which nodes are only connected to the local region, usually only about 3-5 pixels. This keeps the parameter count (comparatively) manageable and allows the network to identify important features wherever they may appear in the image, rather than simply focusing on the center of the image. Each convolutional layer is typically followed by a simple rectified linear unit (ReLu) which forces the convoluted layer to be non-negative. Pooling layers decrease dimensionality and allow the network to identify the dominant features in a region. After the desired convolutional and pooling layers, a final fully-connected layer maps the resulting feature vector to a single subclass among the k-classification groups.

Because of the massive computational time and data required to train a CNN, it is very uncommon to train a model from scratch. Rather, we begin with a model pre-trained on a related problem and modify and re-train this network onto our new task. Generally models are pre-trained on ImageNet, an object classification dataset consisting of millions of labeled data. This "transfer

learning" works because early layers of a CNN identify basic features like edges while later layers identify task-specific features. Even with a transfer learning approach, the roughly 900 clusters per survey year is not enough to re-train an ImageNet CNN directly on estimating the DHS asset index. Rather, we need another data-rich task to between the two. So, we instead first re-train the CNN to estimate nighttime luminosity from the corresponding Landsat image. Once we have trained the CNN to estimate nighttime luminosity, which is correlated with economic outcomes, we can remove the last fully-connected layer which maps the feature vector to a class. The resulting feature vector essentially summarizes the key features of the Landsat image which are predictive of nighttime luminosity. We then use this feature vector in a regression on the DHS asset index. We do not have enough data to train a new neural net, but I do intend to evaluate other machine learning approaches rather than just a simple regression.

Most of the work is concentrated in the first transfer learning stage. Given the complexity of the task, I am building the CNN framework first with just a single year in a subregion of Nigeria containing Lagos, and then I will scale up to the entire region and time period. While the satellite data is publicly available, it is not generally in clean, ready-to-use format. Google Earth Engine provides a Javascript API to clean and export the data. The nightimte luminosity data has one observation per year and is "stable_lights" product has been cleaned. The Landsat data has higher temporal resolution but requires greater cleaning due to cloud cover. By taking median values over a yearlong collection of images we remove the noise and match the temporal frequency of the nightlights and the survey data. Earth Engine only exports rectangular images and will break up images that are sufficiently large. Therefore, I partition Nigeria into 7 roughly equal rectangles which do not get broken up. I overlay the 1km nightlights data and the 30m Landsat data and export to .tiff files which can be read into Python as numpy arrays. To construct the training dataset, I randomly sample coordinate points from the resulting images and extract the 1x1km sub-image, yielding a 34x34 pixel 3-band Landsat image corresponding to a single luminosity score. Consistent with Jean (2016) I find that the luminosity scores have roughly 3 clusters corresponding to near-zero, medium, and high luminosity and so I create three corresponding bins, turning the CNN training into a k=3 classification problem. The sub-image is a numpy array, and I turn it back into a .png file and save it in its respective classes subfolder. I am using Tensorflow to build the model, and this is the file and folder structure it requires. I intend to sample about 300,000 1x1km sub-images across Nigeria for the final full training set. Also consistent with Jean (2016), I find that the majority of luminosity scores are 0. This does not necessarily mean that no one lives in the given area and that there is no economic conditions to estimate, but it does make learning more difficult. Jean (2016) therefore adjust their sampling to have a more even distribution of luminosity scores. My code for building the training data currently includes a parameter to drop a certain percentage of the zero luminosity scores if need be. Constructing the 10,000 image training data I am using for initial tests takes less than a minute to build, so I am confident I can scale the analysis.

Now that I have training data built and conforming to Tensorflow structure I can begin training CNN models. The major current point of uncertainty is with regard to the image size. Most CNN's trained using ImageNet use images greater than 200x200 but the Landsat images are only 34x34. The convolutional layers adjust dynamically to different input sizes and the pre-trained weights correspond to filters that are only 3x3-5x5 in size, so the mathematics of the transfer learning will still work. However, I am unclear if the vastly different resolution will make the tasks too different. Because the early layers do simple edge, color, and texture detection which is still emblematic of human development I think the technique will still apply.