Predicting asset ownership in Africa using satellite imagery

Nicolas Suarez Chavarria
SUNet ID: nsuarez
Department of Economics
Stanford University
nsuarez@stanford.edu

Edoardo Yin
SUNet ID: edoyin
Department of Computer Science
Stanford University
edoyin@stanford.edu

Othman Bensouda Koraichi
SUNet ID: othmanb
Graduate School of Education
Stanford University
othmanb@stanford.edu

1 Introduction

For many parts of the planet, especially in low income countries, on-the-ground data about local development levels is often nonexistent. However, when it exists, it tends to be limited or out of date. Country-level estimates are based on representative household surveys, but those tend to be too sparse to conduct meaningful spatial analyses of how developed different areas within a country are. As a result, granular data on infrastructure access and economic development is sparse, yet many important questions could be addressed if one knew how outcomes are distributed spatially within a country.

Granular measurements of infrastructure access, asset ownership or economic activity can be valuable inputs for researchers and policymakers. Having more accurate spatial data can help policymakers better identify poor or vulnerable groups when formulating public policy, and it can even help them flag remote communities in rural areas, so generating very granular measures could be useful.

This project aims to predict asset ownership in Africa making use of satellite imagery and ground truth data from the Demographic and Health Surveys (DHS). We train a ResNet model (He et al., 2016), a sub-class of a Convolutional Neural Network (CNN), feeding it Landsat-8 Surface Reflectance images to predict asset ownership levels over 6.72 km by 6.72 km patches of land in Africa. We use transfer learning to train our model to perform a regression task, starting our model with weights originally used to classify images.

2 Related work

There is a large body of research in economics which uses satellite images or machine learning predicted outcomes as a proxy for local economic development, including articles by Chen and Nordhaus (2011), Henderson et al. (2012), Michalopoulos and Papaioannou (2014) , Moscona et al. (2020) and Canning et al. (2022).

There has been some work on recent years combining computer vision algorithms with satellite images to map infrastructure access and poverty levels in African countries: Graetz et al. (2018) use satellite imagery to predict educational attainment in Africa, Oshri et al. (2018) use Landsat imagery to predict access to electricity, water, sewage system and other outcomes across Africa, M Rustowicz et al. (2019) use satellite imagery to predict crop types in Africa, Xie et al. (2016) uses nighttime lights data to predict poverty levels in Uganda, while Jean et al. (2016) and Yeh et al. (2020) use Landsat and night lights imagery to predict poverty levels in Africa.

Yeh et al. (2020) does something very similar to what we do. They use all 7 channels of Landsat imagery, and they also use nighttime lights imagery to train their model, so even though we have a similar sample size to theirs, their dataset is more rich. However, the authors in Yeh et al. (2020) trained country specific models with the purpose of generating asset ownership rankings withing countries, whereas we trained our model so it can generates more precise predictions for out-of-sample countries.

3 Dataset and features

3.1 Demographic and Health Surveys (DHS) surveys

To obtain our labels, we use all the household DHS surveys for African countries that have been published since 2014 that have a matching file with the geocoordinates of each cluster. We geocoded 12,511 locations across 24 countries, combining data from 26 DHS surveys between 2014 and 2021.

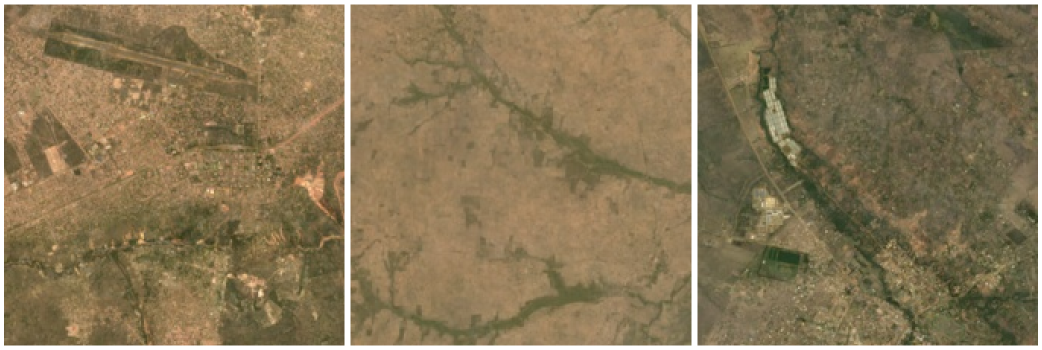
Our variable of interest is the score in the wealth index obtained by each household. This score is produced using Principal Component Analysis over some of the survey questions, including but not limited to access to drinkable water, sewerage, electricity, and ownership of farming and non-farming assets. In some cases we have more than one village per cluster, so we average their wealth indices to geocode them.

3.2 Satellite imagery

The satellite imagery dataset was constructed using the Google Earth Engine API. For each DHS cluster with a geocoded location, we define a patch of 6.72 km $\times$ 6.72 km centered in our location, and we retrieve an image for the patch using the Landsat 8 Surface Reflectance Tier 1 Collection (USGS, 2013). We use 3 bands from this collection: Red, Green and Blue surface reflectance. We preprocess each of our images by adding a cloud mask per pixel and then computing the per pixel and band mean composite of all the available images for the year when the DHS cluster was surveyed. Since Landsat 8 has a resolution of 30 meters and our patches have dimensions 6.72 km $\times$ 6.72 km, our images contain 224×224 pixels, and 3 channels.

We retrieved images for 12,426 locations across all Africa. In Figure 1 we show some examples of the cloud masked imagery we produced.

Figure 1: Examples of images for different countries



(a) Image in Angola **(b)** Image in Chad **(c)** Image in Kenya

3.3 Cross validation and data augmentation

As we mentioned before, we have 12,426 datapoints across all Africa. For cross-validation purposes, we split our dataset in the following way: we assign 80% of our sample to our training set, 10% to the validation set and 10% to the test set. In Table 1 we report the exact size for each set.

Table 1: Training, validation and test set sizes

Set	Training	Validation	Test
Observations	9,941	1,243	1,242

Regarding data augmentation, we apply several transformations to our satellite images: with certain probabilities we apply horizontal and vertical flips to our images, we rotate them in multiples of 90 degrees, and we apply a Gaussian Blur with a 9 by 9 kernel. We also normalized our imagery using the recommended mean and standard deviation for a ResNet-18 pre-trained model. More details about these transformation will be provided in the coming sections.

4 Methods

For this task, we plan to train a Convolutional Neural Network (CNN), specifically a Residual Network. A Convolutional Neural Network is a type of Neural Network generally used for computer vision tasks, where instead of having regular fully connected layers, a CNN contains convolutional layers. These layers are defined by applying a small filter centered around different pixels of an image to produce an activation map. These layers are useful in computer vision tasks because not only they can be used to learn spatial patterns within an image irrespective of their location within the image, but also they have less parameters than a fully connected layer. This last part is important for computer vision tasks, because an image could easily contain more than 100,000 elements if we count all the channels, so using fully connected layers would require training a lot of parameters.

A Residual Network (ResNet) is a special case of a CNN with a special arquitecture: as He et al. (2016) mentioned, deep convolutional network with a lot of layers should theoretically perform better than shallow networks, because they could produce a loss smaller or equal than a shallow network. However, in practice, deep networks performed worse than shallow network. To overcome this, they designed a Residual Network, build around what they called "Residual Blocks": these are regular blocks of convolutional layers connected with activation functions, with the difference that the input of the block is then added to the output of the stacked convolutional layers at the end. The idea behind this is that these residual blocks will help a deep CNN to avoid the problems associated with very deep networks, so if some of the final layers are not helping the model performance, their weights will be set to 0 and the block will become an identity mapping.

We will train a ResNet-18 model, a Residual Network proposed by He et al. (2016) that contains 18 convolution layers grouped in 8 residual blocks, and used ReLU activation functions[1]. Since ResNet models were originally developed to compete in the ImageNet Large Scale Visual Recognition Challenge, the last layer of the original model is a linear layer connected to a softmax layer that classifies images into 1,000 classes. However, for the purposes of our paper, we modify the last linear layer so now it outputs only one number that will represent our predicted asset ownership, and we measure our loss as our root mean squared error (RMSE). To train our model we use transfer learning, so we start our ResNet-18 network with pretrained weights used for the ImageNEt competition. The idea behind this is that even if the original weights were trained to classify images, the different layers of the model already learned to recognize some basic shapes, and our model can use those shapes to learn visual features in our images that are correlated with asset ownership. To do this, we scaled our images with the satellite imagery scale factor, so each element of our images is between 0 and 1, and then normalized the images using the mean and standard deviation required for a pretrained Resnet-18 model.

We also train linear regression, Lasso regression and Ridge regression models to compare the performance of our model against them.

5 Experiments and results

As we mentioned before, we trained linear regression, Lasso regression and Ridge regression models so we can compare their performance against the performance of our ResNet models. In Table 2 we report the RMSE and the R-squared coefficient for our three regression models in our training, validation and test sets. We can see that linear regression and Ridge regression perform incredibly well in our training set, with R^2 values close to 1. However, their performance is the validation and test sets is terrible: they even have negative R^2 values, which means that their predictions have less explaining power than the ground truth average for that set. These performance numbers are indicative of our baseline models overfitting, which is not surprising if we consider that they are trained using 9,941 observations with 150,528 features (224 by 224 pixels for 3 channels). Lasso regression performs poorly in all 3 sets.

After having established the performance of our baseline models, we can now discuss the results of our experiments: In order to tune the hyperparameters of our models, we experimented modifying several components of our model:

- Number of epochs for training.
- Changing our mini-batch training size.
- Applying independent vertical and horizontal flips to our images with certain probabilities.
- Rotating our images in degrees multiples of 90 (0, 90, 180 and 270) with certain probabilities.
- Applying different degrees of Gaussian Blurring to our images.

[1]The model also contains a max pooling filter, an average pooling filter and batch norm layers, but given the limited space we have here, we will not explain what these layers and filters do.

- Freezing the top n convolutional layers of our model.
- Changing our optimizer (Stochastic Gradient Descent or Adam).
- Changing our learning rate.
- For SGD, changing our momentum (momentum makes our gradient a moving average of our previous gradients).

Table 2: Performance metrics for baseline models

Metric	**Set**		
	Training	**Validation**	**Test**
Linear regression:			
RMSE	0.008	2.352	2.403
R^2	1.000	-0.734	-0.683
Lasso regression:			
RMSE	1.733	1.786	1.853
R^2	0.000	0.000	-0.001
Ridge regression:			
RMSE	0.071	2.310	2.359
R^2	0.998	-0.673	-0.622

For each experiment, we report the best validation set RMSE and the batch-average training set RMSE of that epoch. In Panel A of Table 3 we can see the results of our 16 experiments: Experiment 1 is our benchmark model (no frozen layers, no data augmentation), were we observe a validation RMSE of 1.409, which is better than what our baseline models achieved. However, just adding data augmentation in experiment 2 decreased our RMSE substantially to 1.134. In the following 4 experiments we tried freezing between 1 and 4 layers, but this didn't improved our validation set performance. After that, we modified our data augmentation techniques, batch sizes, optimizers and their parameters. After all this, our best experiments were experiments 10 and 11.

Table 3: Results of experiments

Experiment number	**Training Batch Size**	**Number of Epochs**	**Number of frozen layers**	**Horizontal and vertical flip probability**	**Probabilities of rotating images in 0, 90, 180 and 270 degrees**	**Gaussian Blur**	**Optimizer**	**Learning rate**	**Momentum**	**Best validation set RMSE**	**Training set average RMSE**
Panel A. Experiments:											
1	500	100	0	-	-	-	SGD	0.001	0.9	1.409	0.296
2	500	100	0	0.4	[0.4,0.2,0.2,0.2]	Yes	SGD	0.001	0.9	1.134	0.594
3	500	100	1	0.4	[0.4,0.2,0.2,0.2]	Yes	SGD	0.001	0.9	1.253	1.171
4	500	100	2	0.4	[0.4,0.2,0.2,0.2]	Yes	SGD	0.001	0.9	1.236	1.016
5	500	100	3	0.4	[0.4,0.2,0.2,0.2]	Yes	SGD	0.001	0.9	1.200	1.057
6	500	100	4	0.4	[0.4,0.2,0.2,0.2]	Yes	SGD	0.001	0.9	1.187	0.863
7	500	100	0	0.4	-	Yes	SGD	0.001	0.9	1.162	0.503
8	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.001	0.9	1.114	0.606
9	250	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.001	0.9	1.105	0.551
10	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	Adam	0.001	-	1.043	0.804
11	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.01	0.9	0.998	0.425
12	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	Adam	0.01	-	1.250	1.406
13	500	100	0	0.5	[0.25,0.25,0.25,0.25]	-	SGD	0.001	0.9	1.174	0.537
14	250	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.01	0.9	1.073	0.482
15	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.01	0	1.125	0.797
16	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.01	0.5	1.059	0.594
Panel B. Extra epochs for best experiments:											
11	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	SGD	0.01	0.9	1.006	0.467
10	500	100	0	0.5	[0.25,0.25,0.25,0.25]	Yes	Adam	0.001	-	0.976	0.555

For our first 16 experiments we didn't modify the number of epochs we use. Because of this, in Panel B of Table 3 we train our best models (the best models of experiments 11 and 10) for 100 extra epochs. After training for 100 extra epochs we only observe modest changes in validation test RMSE, with the model from experiment 11 getting an slightly higher RMSE, and the model from experiment 10 obtaining an slightly smaller RMSE. The best model from experiment 10 was originally reached after 69 epochs, and when we trained it for 100 extra epochs, it hit its new best after 11 epochs, so we think training a model for a total of 200 is enough to converge to a relatively good set of parameters.

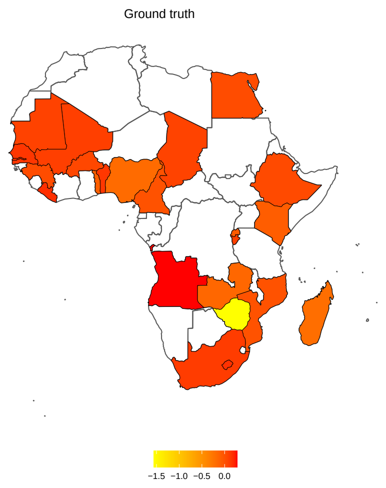
Based on this, we decided our best model is the model of experiment 10 after retraining it for 100 epochs. In table 4 we report the RMSE and R^2 of our final model for our training, validation and test sets: We can see that our model performs much better than our baseline models in the validation and test sets, and we obtain good R^2 values in the validation and test sets. However, our R^2 in the training set is very high, so our model might be overfitting our training data.

Table 4: Performance metrics for best experiment

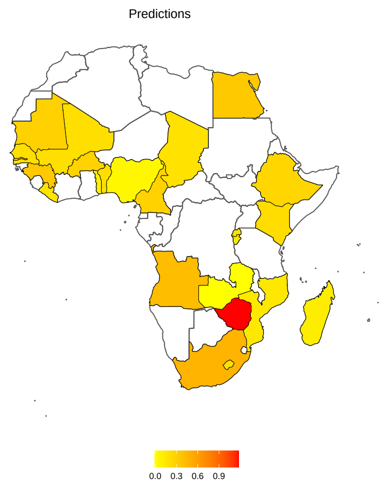
Metric	**Set**		
	Training	**Validation**	**Test**
RMSE	0.478	0.976	1.309
R^2	0.948	0.591	0.501

Finally, to assess the performance of our model, in Figure 2 we plot the country average asset ownership for pixels in our test set. We can see that our predicted country averages are quite off from the original ground truth: most of the ground truth averages are negative, but we predict on average positive values of asset ownership. This is consistent with our test set RMSE, meaning that our predictions are on average 1.3 units off from our ground truth values.

Figure 2: Average asset ownership per country for test set observations



(a) Ground truth country averages



(b) Predicted country averages

6 Concluding remarks and future work

In this project we trained a ResNet-18 model with Landsat-8 satellite imagery to predict asset ownership in Africa. We used geocoded DHS surveys to generate a sample of 12,426 labeled images covering 6.72 km by 6.72 km in 24 African countries. We modified the original ResNet-18 architecture so we can train it for a regression task, and we used transfer learning techniques to take advantage of the pretrained ResNet-18 weights.

We experimented freezing layers of our model, using data augmentation techniques, using different optimizers and learning rates. Based on our experiments, our best model was a ResNet-18 trained for 200 epochs, with a training batch size of 500 images, several data augmentation transformations applied to our images, and optimized using the Adam optimizer with a learning rate of 0.001. The performance of this model is much better than the performance of our baseline models, but the model might be overfitting.

Our model is probably overfitting, the predictions are not that related to our ground truth, and we obtained lower R^2 values than Yeh et al. (2020), so there is room for improvement here. Future work on this might include expanding our sample size by including more years with DHS surveys, using more channels of our satellite images, and increasing the regularization in our model, adding dropout probabilities to our nodes and tuning that hyperparameter.

Appendix

Contributions

- Nicolas produced the geocoded DHS dataset, wrote the code to download imagery with Google Earth Engine, converted the images to Pytorch tensors and produced the dataloaders, wrote most of the code to define Pytorch models, train them and do the experiments.
- Othman wrote part of the code for the baseline models, helped with train/test/validation split and helped with data augmentation and freezing layers in our experiments.
- Edoardo wrote part of the code for the baseline models.

References

Canning, D., Mabeu, M. C., & Pongou, R. (2022). Colonial origins and fertility: Can the market overcome history? *Available at SSRN 4070963*.

Chen, X., & Nordhaus, W. D. (2011). Using luminosity data as a proxy for economic statistics. *Proceedings of the National Academy of Sciences*, *108*(21), 8589–8594.

Graetz, N., Friedman, J., Osgood-Zimmerman, A., Burstein, R., Biehl, M. H., Shields, C., Mosser, J. F., Casey, D. C., Deshpande, A., Earl, L., et al. (2018). Mapping local variation in educational attainment across africa. *Nature*, *555*(7694), 48–53.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Henderson, J. V., Storeygard, A., & Weil, D. N. (2012). Measuring economic growth from outer space. *American economic review*, *102*(2), 994–1028.

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *353*(6301), 790–794.

M Rustowicz, R., Cheong, R., Wang, L., Ermon, S., Burke, M., & Lobell, D. (2019). Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 75–82.

Michalopoulos, S., & Papaioannou, E. (2014). National institutions and subnational development in africa. *The Quarterly journal of economics*, *129*(1), 151–213.

Moscona, J., Nunn, N., & Robinson, J. A. (2020). Segmentary lineage organization and conflict in sub-saharan africa. *Econometrica*, *88*(5), 1999–2036.

Oshri, B., Hu, A., Adelson, P., Chen, X., Dupas, P., Weinstein, J., Burke, M., Lobell, D., & Ermon, S. (2018). Infrastructure quality assessment in africa using satellite imagery and deep learning. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 616–625.

United States Geological Survey. (2013). *Landsat 8 surface reflectance tier 1 collection* [Accessed September, 2022].

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. *Thirtieth AAAI Conference on Artificial Intelligence*.

Yeh, C., Perez, A., Driscoll, A., Azzari, G., Tang, Z., Lobell, D., Ermon, S., & Burke, M. (2020). Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature communications*, *11*(1), 1–11.