Incorporating Spatial Context and Fine-grained Detail from Satellite Imagery to Predict Poverty

Jae Hyun Kim
Stanford University
jakekim@stanford.edu

Michael Xie
Stanford University
xie@cs.stanford.edu

Neal Jean
Stanford University
nealjean@stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

ABSTRACT

The lack of accurate poverty data in developing countries and the high cost of obtaining this data pose major challenges for making informed policy decisions and allocating resources effectively. The rise of inexpensive, high-resolution remote sensing data such as satellite imagery can provide global-scale information, given methods for extracting useful insights from this unstructured data. In this paper, we present a machine learning approach to predicting real-valued asset indices in five countries in Africa which uses satellite imagery from multiple resolutions to incorporate both spatial context and fine-grained detail. Following the model developed by Xie et al. [17], we train convolutional neural network models to predict nighttime light intensity from daytime imagery, simultaneously learning features that are useful for asset prediction. We show that for many different resolutions, single-resolution models trained using this transfer learning method can extract features that are good predictors of asset measures. Further, we demonstrate that models trained on different resolutions extract semantically different features that are often conceivable by human sight. Finally, we propose two multi-resolution models that combine spatial context and fine-grained details of satellite images to outperform the single-resolution models as well as the previous state-of-the-art.

1. INTRODUCTION

Data scarcity in the developing world poses a major challenge for making informed policy decisions and allocating resources effectively [4, 15]. Countries with advanced economies, such as the United States, have sufficient resources and institutional infrastructure to be able to collect detailed household survey data on a regular basis, but these capabilities do not exist in many of the world's poorest nations. According to data from the World Bank, only a few African countries have conducted more than two nationally representative surveys since the year 2000 [16]. This lack of reliable data severely hinders our ability to identify areas of greatest need and to understand the complex dynamics of poverty.

With the proliferation of passively collected data from cell phones and other distributed sensors, a variety of creative solutions to this data shortage problem have been offered. In a recent paper, Blumenstock et al. [2] predict the distribution of poverty in Rwanda using proprietary cell phone data. Other approaches based on call record data have been applied to create poverty maps in developing countries [7, 12].

Remote sensing data is also becoming increasingly accurate and inexpensive. Given methods of extracting useful insights from unstructured data, remote sensing data such as high-resolution satellite imagery can provide valuable information at a global scale. Xie et al. [17] use publicly available satellite images to predict poverty, circumventing the lack of labeled data through a transfer learning approach. Using nighttime light intensity prediction as a data-rich proxy task, they train a convolutional neural network (CNN) to learn to identify image features such as roads, urban areas, and terrains that are predictive of poverty.

We improve upon this approach by incorporating satellite imagery of multiple resolutions. Different resolutions contain semantically different information, with lower resolutions providing coarser information about spatial context (e.g., rural/urban distinctions, major roads, nearby population centers) while higher resolutions capture more detailed features of the area of interest (e.g., individual buildings, building materials, cars).

In this paper, we train residual CNN networks (ResNet [6]) to predict higher-resolution nightlight intensities from the VIIRS dataset using daytime satellite images of multiple resolutions, in comparison to the older VGG architecture trained using images of a single resolution and the lower-resolution DMSP nightlight data used in Xie et al. [17]. Through this transfer learning proxy task, our models learn to extract low- and high-resolution features from satellite images that can be generalized to poverty prediction. The trained CNNs serve as feature extractors, and the resulting image features are used to predict real-valued asset indices. We assess the performance of two types of models: "in-country" models that are trained on and evaluated using data from the same country, and "out-of-country" models that are trained on data from one country and evaluated using data from different countries. By using satellite imagery from multiple resolutions, we are able to capture both spatial context and high-resolution detail.

We find that for many different resolutions, the single-resolution models trained using the transfer learning approach can extract features that are good predictors of asset measures. For locations where multiple resolutions are available, we also demonstrate that by combining image features from different resolutions of imagery, we can improve significantly on the performance achieved by any individual single-resolution model. Our multi-resolution models outperform the previous state-of-the-art, and often compare fa-

vorably with predictions made using expensively collected survey data.

2. BACKGROUND

2.1 CNNs and residual learning

Convolutional neural networks trained on the ImageNet image recognition challenge have had remarkable success in object classification, semantic segmentation, and other fundamental computer vision tasks [13]. A convolutional neural network (CNN) is a hierarchical model which includes multiple layers of convolution operations over the input. The convolution operator is translation invariant, which is important for identifying features in images [3]. For image data, the initial layers typically identify low-level features such as edges, while additional layers identify high-level features such as objects and textures [18]. The top layer encodes the input as a feature vector, which is then used as the input to a final classifier. Therefore, we can view a CNN as a mapping $\mathcal{F} : \mathcal{X} \to \mathbb{R}^n$ from the space of input images $\mathcal{X}$ to their feature vector representations.

The success of CNNs has also launched various attempts to repurpose the learned feature representations for generic tasks. Donahue et al. [5] show that deep features learned from training on the ImageNet dataset directly achieve good performance for object recognition, domain adaptation, subcategory recognition, and scene recognition.

We use a residual network (ResNet) developed by He et al. [6], which uses a gradient "highway" to train a much deeper architecture without gradient decay problems. This architecture achieved the best performance on the 2015 ImageNet challenge for object recognition and allows for faster convergence during training. In residual learning, we allow the neural network to approximate the residual function $\mathcal{F}(x) = \mathcal{H}(x) - x$, where $\mathcal{H}(x)$ is the underlying mapping to be fit and x is an input. Implicitly, we assume that $\mathcal{H}(x)$ can be more easily approximated by $\mathcal{F}(x) + x$ than directly. Formally, a residual layer is defined

$$y = \mathcal{F}(x, W) + W_s x$$

where W are the residual weights and W_s is a transformation to match the size of $W_s x$ with $\mathcal{F}$. Note that W_s is chosen as the identity mapping wherever possible and is not optimized in that case. Each layer with residual learning requires the input from an earlier layer as part of its calculation, mitigating the gradient degradation problem. Intuitively, this ensures that adding layers can only improve performance, since we can set the residual weights of the additional layers to zero to learn an identity mapping from the shallower network.

2.2 Transfer learning

Transfer learning uses knowledge from a related task to improve performance on the target task (see Pan and Yang [11]). As in Xie et al. [17], we define a transfer learning problem $\mathcal{P} = (\mathcal{D}, \mathcal{T})$ as a domain-task pair. Transfer learning is used to improve the learning of a target predictive function in the target problem $\mathcal{P}_T$ by using knowledge from a source problem $\mathcal{P}_S$. The transfer of knowledge forms a *transfer learning graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a directed acyclic graph with vertices $\mathcal{V} = \{\mathcal{P}_1, \ldots, \mathcal{P}_v\}$ and edges $\mathcal{E} = \{(\mathcal{P}_{i_1}, \mathcal{P}_{j_1}), \ldots, (\mathcal{P}_{i_e}, \mathcal{P}_{j_e})\}$. For each problem $\mathcal{P}_j \in \mathcal{V}$, we transfer the knowledge gained through solving the parent problems in the graph $\cup_{(\mathcal{P}_i, \mathcal{P}_j) \in \mathcal{E}} \mathcal{P}_i$.

Figure 1: Daytime images of the same location corresponding (from left to right) to resolutions of zoom level 14, 16, and 18 respectively.

3. NIGHTTIME LIGHT PREDICTION

3.1 ImageNet initialization

Let the domain of natural images be $\mathcal{D}_{NI}$ and the ImageNet classification task be $\mathcal{T}_{IN}$. The ImageNet challenge problem $(\mathcal{D}_{NI}, \mathcal{T}_{IN})$ is the first source problem in our transfer learning graph pipeline seen in Fig. 2, as we transfer the knowledge learned from the ImageNet problem to the nighttime light intensity prediction problem by initializing our models with parameters trained with ImageNet. Although the ImageNet domain consists of natural images that are fundamentally different from the aerial view of satellite images, low-level features such as edges and corners are still generally useful.

3.2 Daytime satellite imagery

In the intermediate transfer learning problems, we learn a predictive function for the nighttime light intensity prediction task $\mathcal{T}_{NL}$ from the domain of daytime satellite images $\mathcal{D}_{SI_r}$ of resolution (zoom) level r obtained from the Google Static Maps API. We use zoom levels $r \in R = \{14, 16, 18\}$, which correspond to resolutions of 9.55 m, 2.39 m, and 0.60 m per pixel respectively. Therefore, 224-by-224 pixel images cover areas of 2.14 km by 2.14 km, 0.54 km by 0.54 km, and 0.13 km by 0.13 km respectively.

At the highest resolution (zoom level 18), landscape features such as individual buildings and rooftops can be seen clearly. These details are useful for inferring quantities of interest (e.g., consumption expenditures or asset ownership) at a fine-grained level. Below this resolution, we generally cannot see building outlines or cars. At the lowest resolution (zoom level 14), larger-scale features, such as roads and other types of infrastructure and natural terrains, dominate the variation in the images. Since the lower resolutions are able to cover more land area in each image, they also capture the surrounding spatial context—a small farm in the middle of a desert likely has a different economic outlook than one that is located on the outskirts of a large city.

3.3 Nighttime light intensities

To obtain nighttime light intensity labels, we use data from the Visible Infrared Imaging Radiometer Suite (VIIRS) satellite at a 15 arc-second resolution (∼0.5 km), twice the resolution of the 30 arc-second Defense Meteorological Satellite Program (DMSP) data used in Xie et al. [17]. The VIIRS data is also more recent than the DMSP data, which is from 2013, allowing for the daytime satellite images and

nighttime light labels to be matched temporally. The majority of our daytime imagery was taken in 2015, so we use 2015 VIIRS data. However, while the DMSP data is processed to filter out noise from ephemeral light sources, the VIIRS data is not. We process the VIIRS data by averaging nighttime light intensity values over 2015, weighted by the number of observations in each month. These annual weighted averages are then used as ground truth nighttime light values.

As in Xie et al. [17], we discretize the nighttime light intensities into classes by observing the distribution of nighttime lights across Africa. We bin the intensities into 3 classes, with the classes ranging from 0 to 8, 8 to 35, and 35 to 200 $nW/cm^2 - sr$. The lowest light intensity class includes natural terrains, the middle intensity class includes areas with some human activity such as rural areas, and the highest intensity class includes most of the urban areas. For each class, we sample 50,000 images and augment the data by random rotations. We sample an additional 10,000 images from each class for the test set.

4. POVERTY MEASURE PREDICTION

In our final target problem, we predict an asset-based measure of wealth using $\mathcal{D}_{SI_R} = \cup_{r \in R} \mathcal{D}_{SI_r}$, the space of satellite imagery data with multiple resolution levels R. For the poverty prediction task, $\mathcal{T}_P$, we use an asset index for five countries (Nigeria 2013, Tanzania 2010, Uganda 2011, Malawi 2010, Rwanda 2010) drawn from the Demographic and Health Surveys (DHS). The index is computed as the first principal component of survey data about ownership of assets [8]. While the asset index cannot be used directly to construct benchmark measures of poverty, asset-based measures are thought to better capture households' longer-run economic status and are measured with relatively less error since they are directly observable to survey administrators [14].

The DHS survey data is collected at the household level, but the locations of the households in the survey are provided only at the "cluster" level, roughly corresponding to villages in rural areas and wards in urban areas. To preserve the anonymity of the participants, these cluster locations also include random noise, up to 2 km in each direction for urban clusters and up to 5 km for rural clusters. Cluster-level wealth measures are computed by averaging the asset index of all households belonging to each cluster. In the poverty measure prediction problem, we aim to predict the real-valued cluster-level asset index given a set of input satellite images.

4.1 Extracting features

A CNN model takes a satellite image as input and outputs a feature representation of the image. After training each model on the nighttime light prediction problem, we can extract image features from new input images by evaluating the model on the new image. For each DHS survey cluster location, we sample a set of images which are used to construct a feature vector for the cluster. For urban clusters, we extract features from 256 images in a 4 km by 4 km bounding box centered on the DHS cluster location. For rural clusters, we extract from 400 images in a 10 km by 10 km bounding box centered on the cluster location. The difference in sampling areas arises from the different amounts of noise added to rural and urban cluster locations. We then average the 256 or 400 feature vectors for each cluster into a final feature vector for the cluster. For locations where Google Maps imagery was incomplete, we use as many images as possible (up to 256 or 400) for computing the cluster image features.

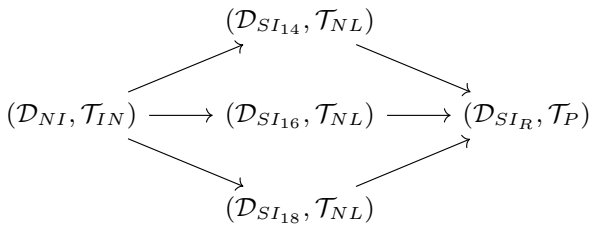

Figure 2: Transfer learning graph first depicting the transfer of knowledge from the domain of natural images ($\mathcal{D}_{NI}$) and the ImageNet classification task ($\mathcal{T}_{IN}$) to the nighttime light prediction task ($\mathcal{T}_{NL}$) with multiple resolution levels of input satellite images ($\mathcal{D}_{SI_r}$ for $r \in R = \{14, 16, 18\}$), then depicting the transfer of knowledge to a poverty measure prediction task ($\mathcal{T}_P$) with a multi-resolution domain of satellite images ($\mathcal{D}_{SI_R} = \cup_{r \in R} \mathcal{D}_{SI_r}$).

4.2 Making predictions

The cluster image features are then used to train ridge regression models that can predict the average household asset index for each cluster. In our experiments, we first reduce the feature dimension to 100 using principal component analysis (PCA) to speed up computation while still capturing almost all of the variation in the data. Regularization in the ridge model guards against overfitting, a potential challenge given the relatively small number of training examples in the survey datasets (< 1000).

We evaluate asset prediction models through in-country and out-of-country tests. For in-country models, we use DHS data and daytime imagery from one country to train a model that is then evaluated on a test set from the same country. An example of this would be to train a model using images and survey data from Uganda, then test that model on test survey data from Uganda. An out-of-country model uses survey data and daytime imagery from one country to train a model, then applies that model to predict assets from daytime imagery in a different country. An example of this would be to take the model that was trained on Uganda and use it to predict asset-based wealth measures in Tanzania, Nigeria, Malawi, or Rwanda.

5. MODELS

5.1 Single-resolution models

In the single-resolution setting, we train ResNet-50 models (which each have 50 layers) to predict nighttime light intensity classes using input daytime satellite images of a single resolution. The locations for the training set are sampled randomly from the entire African continent, and the model is trained on the 3-class VIIRS nighttime light intensity labels described in section 3.2. Through this proxy task,

the ResNet-50 models learn to map each input image into a 2048-dimension feature vector representation that is useful for predicting nighttime light intensities. We train three separate single-resolution models using zoom 14, zoom 16, and zoom 18 images, which we will refer to as the Zoom14, Zoom16, and Zoom18 models respectively.

5.2 End-to-end model

One way to use information from multiple resolutions is to combine the single-resolution models from all three resolutions and concatenate their output feature vectors. We train an end-to-end neural network model that takes as input 3 images with different resolutions (zoom 14, 16, 18) for each location and outputs a 6144-dimensional feature vector. The three ResNet-50 models are initialized with the final parameters of the trained single-resolution models and are then trained jointly on the nighttime light intensity prediction problem. In the process of training, the three single-resolution models are able to optimize how they identify features in relation to the other models, potentially allowing models of different resolutions to focus on features that are unique to their resolution instead of trying to find features that can predict nightlights using information from only one resolution.

5.3 Neural network model

Another way to combine the information from multiple image resolutions is to train a shallow neural network model which takes in the concatenated features from the multiple single-resolution models and outputs a combined feature vector that also has fixed dimensionality. After training the single-resolution models, we fix the 6144-dimensional concatenated features as input and train this shallow network on the nighttime light prediction problem. This network has one 2048-dimensional hidden layer with a ReLU nonlinearity, and therefore the learned feature representation is 2048-dimensional. The network is implemented using TensorFlow [1] and trained using the ADAM optimizer [10] with initial learning rate 1e-6. Dropout regularization with a dropout probability of 0.5 is employed with weight decay of 0.005. In the training process, the neural network model optimizes a nonlinear transformation of the concatenated multi-resolution features on the transfer learning task. One drawback of this model is that while the regression coefficients of the End-to-end model can easily be interpreted as feature contributions from different resolutions, the neural network model is not as interpretable.

6. RESULTS AND DISCUSSION

6.1 Nighttime light prediction

The single-resolution ResNet-50 modls are trained with Caffe and initialized from the parameters learned from the pre-training process on the ImageNet dataset [9]. Random mirroring is used for data augmentation, and stochastic gradient descent (SGD) with momentum is run for 150,000 iterations using batch sizes of 16 (4 epochs) to fine-tune the parameters on the nightlight prediction task. The learning rate is initially set to 1e-6, and then dropped by a factor of 0.5 every 10,000 iterations. We use a momentum parameter of 0.9 and a weight decay of 0.0005, with other hyperparameters set to the same values used in the pre-training process. We choose the weights corresponding to the minimum of the moving average of validation accuracy over 5,000 iterations as the final model parameters.

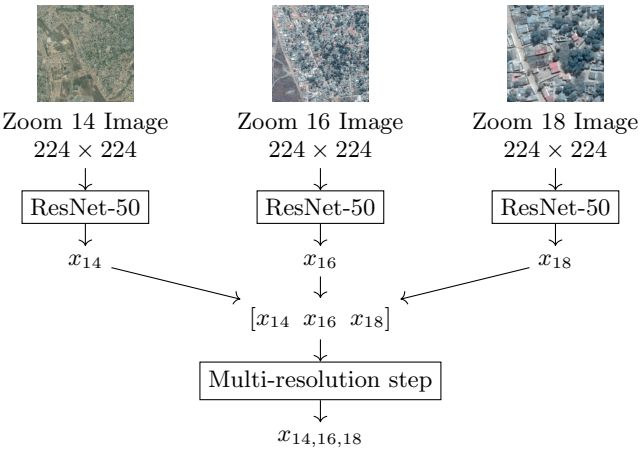

Figure 3: General multi-resolution model architecture which takes 3 input different resolution images per location and outputs a feature vector by concatenating the 2048-dimensional output of 3 single-resolution ResNet-50 models trained on the nighttime light transfer learning problem and using this concatenated vector as input to a multi-resolution step. In the End-to-end model, we take the multi-resolution step as the identity mapping and train the architecture jointly. In the Neural model, we take the multi-resolution step as a single hidden-layer neural network and train the multi-resolution step while keeping the rest of the architecture fixed.

On the 3-class nighttime light prediction problem, the Zoom14 (9.55 m/pixel) model achieved test accuracy of 0.7590 after 62,000 iterations, the Zoom16 (2.39 m/pixel) model achieved test accuracy of 0.7275 after 82,500 iterations, and the Zoom18 (0.60 m/pixel) model achieved test accuracy of 0.6878 after 54,500 iterations.

The End-to-end model is initialized with the final parameters of the trained single-resolution models and trained on the nighttime light prediction problem. All hyperparameters, including learning rate and momentum, are the same as single-resolution models. For the night light prediction problem, the End-to-end model achieved test accuracy of 0.7856 after 40,000 iterations.

We also use the nighttime light prediction problem as a transfer learning proxy for training an additional neural network which combines the single-resolution features. The image features extracted using the single-resolution models are concatenated and fed as input into a shallow neural network with one hidden layer, as described in 5.3. During the training stage, we use batch sizes of 1024. As before, we stop training when the moving average of the validation accuracy is minimized, which occurs after 72,250 iterations (123.3 epochs) with a final test accuracy of 0.8920 on the nighttime light prediction problem.

6.2 Model and feature generalization

We transfer knowledge from the nighttime light prediction

Country	VGG	Zoom14	Zoom16	Zoom18	End-to-end	Neural	Survey
Nigeria	0.6917	0.7440	0.7684	0.7579	0.7747	**0.7841**	0.7648
Tanzania	0.5746	0.7084	0.6984	0.6974	0.7468	**0.7615**	0.6866
Uganda	0.6614	0.7226	0.7579	**0.7610**	0.7447	0.7517	0.8252
Malawi	0.5470	0.6027	0.6221	0.6039	0.6252	**0.6380**	0.8218
Rwanda	0.7438	0.7701	0.7732	0.7722	0.7786	**0.7788**	0.5870

Table 1: r^2 for in-country models, which are trained and tested within the same country. The VGG models are trained following Xie et al. [17]. The Zoom14, Zoom16, Zoom18 models are single-resolution ResNet-50 models. The End-to-end and Neural models are multi-resolution models as described in Sections 5.2 and 5.3. The Survey model does not use satellite imagery, and is built using features from the DHS surveys.

Country	VGG	Zoom14	Zoom16	Zoom18	End-to-end	Neural	Survey
Nigeria	0.3968	0.4854	0.5623	0.5299	0.5702	**0.5799**	0.5686
Tanzania	0.4499	0.5001	0.5870	0.5870	0.6106	**0.6365**	0.6307
Uganda	0.5671	0.4298	0.5857	**0.6325**	0.6098	0.6218	0.7417
Malawi	0.4162	0.4093	0.4579	0.4374	0.4667	**0.5019**	0.5895
Rwanda	**0.6216**	0.5417	0.5799	0.5950	0.5971	0.6054	0.4860

Table 2: r^2 for out-of-country models, which are trained and tested in different countries.

problem to the asset index prediction problem by using the various trained CNN models as fixed feature extractors. The extracted image features are then used to build ridge regression models that predict asset indices. The metrics used to evaluate performance on the target poverty prediction problem are r^2 (where r is Pearson's correlation coefficient) and mean squared error (MSE). The linear correlation between the true asset indices and the predicted values is measured by the r^2 value, giving a measure of the ability of the image features to explain relative differences in asset wealth between clusters. The MSE measures how accurately the ridge regression model is actually predicting the outputs.

For each model, we evaluate prediction performance when trained and evaluated within the same country (in-country), and when trained and evaluated in different countries (out-of-country). We note a crucial difference between model generalization and feature generalization, which are captured by MSE and r^2, respectively. Model generalization refers to whether the regression coefficients are similar from training to test sets, such that the model prediction error (MSE) would be low. For example, if a model generalizes well in an out-of-country test, then the regression coefficients are similar across countries, suggesting that the same combination of features are important in many countries. Feature generalization refers to the ability of image features to capture variation in asset wealth via a linear dependence, which corresponds to high r^2. For example, if a set of features generalizes well in an out-of-country test, then these features allows for a good linear model of asset wealth to be fit across countries. However, if the true linear coefficients of these features are not necessarily the same in these different countries, then the predictive accuracy (MSE) across countries is not necessarily high. Even if the features give rise to regression models whose outputs are linearly correlated with assets in every country (high r^2), if these models have varying coefficients across countries, then they will perform poorly on predicting asset values (high MSE) in out-of-country tests. Therefore, it is possible for the features to generalize well but for the model to generalize poorly, exhibiting both high r^2 and high MSE.

6.3 Asset-based wealth prediction

We now compare the results of the single resolution models and the two multi-resolution models on asset-based wealth prediction. For in-country models, we employ a nested 10-fold cross validation scheme in which the inner cross validation loop chooses the regularization parameter by averaging the best regularization parameters found in the 10 inner folds and the outer cross validation loop evaluates a ridge regression model using the regularization parameter found in the inner loop. For out-of-country r^2 and MSE, we take the average of the four values obtained by applying the model trained on the four out-of-country tests to the target country (e.g., Tanzania $\rightarrow$ Nigeria, Uganda $\rightarrow$ Nigeria, Malawi $\rightarrow$ Nigeria, Rwanda $\rightarrow$ Nigeria). For all the regression models using satellite imagery, we use PCA to reduce the dimension of the input feature vectors to 100, removing the possibility that any differences in performance come from differences in feature dimension.

As a baseline, we also compare the results to regression models built using data from the same DHS surveys that the asset indices are drawn from. In this "Survey" model, we use other features of the cluster households to try to predict the average wealth index of the cluster. The features used for the survey model are: proportion of roofs that are metallic, average number of rooms per household, whether the cluster is urban or rural, average distance to the nearest population center of at least 100 thousand people, average elevation, and average temperature. The Survey model does not make use of satellite imagery.

To compare against the existing state-of-the-art, we also evaluate the performance of the VGG-based model described in Xie et al. [17]. The models are trained according to the transfer learning approach involving nighttime light prediction as the proxy task and are subject to the same in-country and out-of-country tests.

Tables 1 and 2 show r^2 values for in-country and out-of-country tests. Across the single-resolution models, r^2 values

Country	VGG	Zoom14	Zoom16	Zoom18	End-to-end	Neural	Survey
Nigeria	0.2408	0.1946	0.1778	0.1842	0.1714	**0.1652**	0.2398
Tanzania	0.3025	0.2066	0.2113	0.2193	0.1828	**0.1730**	0.1714
Uganda	0.2691	0.2204	**0.1918**	0.1926	0.2038	0.1932	0.1377
Malawi	0.3025	0.2658	0.2544	0.2650	**0.2453**	0.2466	0.1267
Rwanda	0.1789	0.1604	0.1663	0.1603	0.1566	**0.1556**	0.2987

Table 3: MSE for in-country models.

Country	VGG	Zoom14	Zoom16	Zoom18	End-to-end	Neural	Survey
Nigeria	0.6756	0.4662	0.6567	0.8829	0.5457	**0.4337**	0.9093
Tanzania	0.4138	0.4057	0.4243	0.4385	0.3933	**0.3253**	0.4787
Uganda	0.4356	0.4689	0.4412	0.5956	0.4090	**0.3687**	0.6755
Malawi	0.5432	0.4764	0.4853	0.4931	0.4476	**0.4272**	0.4682
Rwanda	0.3877	0.3970	**0.3531**	0.3756	0.3895	0.3866	0.6211

Table 4: MSE for out-of-country models.

tend to increase with resolution, especially in out-of-country models. This indicates that features extracted by higher resolution models generalize better than the lower resolution features, suggesting that fine-grained details (e.g., roof types, cars) are linearly correlated with asset wealth and capture more of the variation in wealth than low resolution features such as infrastructure and terrains. The End-to-end and Neural models have higher r^2 and lower MSE values than all single-resolution models and the VGG model in almost all countries. In particular, the Neural model exhibits the best performance in many cases, although the performance of the multi-resolution models are generally similar. Therefore, the features extracted by the multi-resolution models generalize the best both in-country and out-of-country. Additionally, out-of-country r^2 and MSE values for the multi-resolution models show better improvement than in-country r^2 values, suggesting that the multi-resolution step significantly improves the cross-border generalizability of the features. Finally, in comparison to prediction using features from the survey, we find that the multi-resolution models are competitive with the survey model for in-country tests and particularly strong in out-of-country tests. While survey data provides accurate measurements at a very sparse, local level, additional data such as census data is usually needed to extrapolate the survey data across a country. In contrast, our model allows for high-resolution coverage of countries even without survey data, as suggested by the strong out-of-country generalization.

Tables 3 and 4 show MSE for in-country and out-of-country cases. Across the single-resolution models, the MSE values tend to increase with increasing resolution, especially for out-of-country models. With the knowledge that higher resolution features generalize better across countries, this indicates that the ridge regression coefficients of higher zoom levels do not generalize well across countries, suggesting that high-resolution features are important for capturing variation, but having same values of high-resolution features may not correspond to equivalent levels of absolute wealth in different countries. On the other hand, this suggests that low-resolution models generalize well across countries. Since low-resolution images contain more infrastructure and terrain information, they can give a better absolute estimate of wealth without having enough variation to explain the intricate patterns of asset wealth at the cluster level.

Therefore, combining the two results, we conclude that higher resolution models tend to extract features which give rise to models with a high linear correspondence with asset wealth even across different countries, while lower resolution models give rise to models with coefficients that generalize better across countries, highlighting features that are consistently important. By taking advantage of the complementary information in the different resolutions, the multi-resolution models immediately improves in both r^2 and MSE. We further learn the relationship between the features from different resolutions through the neural network model, and performance improves in almost every test.

6.4 Visualizing the features

We can visualize the features identified by the different single-resolution models by examining the filter activations. We observe the output of each filter in layer 4f of the ResNet-50 model as described in He et al. [6] on a set of sample images. For each filter, we look for images with significant activation, meaning that the image contains features that the filter has learned to look for. By looking at the activation map, which is the output of the convolution after applying the nonlinearity, we can determine where the filter activates most strongly on the image.

Fig. 4 shows several examples of filter activation maps for the three single-resolution models. Low-resolution features capture geography and spatial context, such as urban area, rivers, and desert-like regions, while high-resolution features capture fine-grained details such as cars, buildings, and roads. High-resolution features achieved high r^2 in out-of-country tests, implying that fine-grained details such as cars and building features are highly correlated with assets. However, the high MSE of the high-resolution models indicates that, despite strong linear correlations, we cannot make accurate predictions with only those features. For example, suppose that one of the high-resolution features represents the number of cars found in an image. It is likely that a village with 10 cars in Malawi is wealthier than a village with 5 cars, and that this relative wealth relationship would also hold in Uganda. However, a village with 10 cars in Malawi may not have the same absolute level of wealth as a village with 10 cars in Uganda. High-resolution features

Figure 4: Activations of filters for single-resolution models. Each of the three columns corresponds to zoom level 14, 16, and 18, respectively. For each column, the image on the left is the satellite image, and the image on the right is the activation of the filter. Four representative filters out of 512 filters were chosen for each zoom level. Zoom 14 model captures geography/spatial contexts such as rivers, seas, desert-like cities, and other urban areas (top to bottom). Zoom 16 model captures infrastructures such as building clusters, farmlands, large buildings, and roads. Zoom 18 model captures fine-grained details such as cars, shadows (implying tall buildings), sparse buildings, and swimming pools.

may fail to capture more macroscopic characteristics that affect absolute wealth.

On the other hand, low-resolution features such as roads, urbanness, and infrastructure are not as linearly correlated with assets due to less variation between regions. However, they predict the absolute level of wealth better than high-resolution features, achieving better out-of-country MSEs. In other words, a region with more cars is wealthier than a region in the same country with less cars, but to determine the region's absolute wealth, and further a nation's level of wealth, spatial context and infrastructure are generally more important. The spatial context tells us the general level of wealth of a nation, while the fine-grained details provide us with information needed to compare between regions. Therefore, both the spatial context and fine-grained details are necessary to have an accurate depiction of the nation's wealth.

7. CONCLUSIONS

For the asset-based wealth prediction problem in Africa, we find that single-resolution models for three semantically different resolutions can each achieve reasonable performance, suggesting that each resolution contains useful information. We show two different ways of building models which incorporate image features from all three resolutions, the End-to-end model and the Neural model. These multi-resolution models provide an immediate performance improvement over any single-resolution model, including the state-of-the-art, in both in-country and out-of-country performance.

In analyzing the results, we find that high-resolution features and low-resolution features have different properties. High-resolution features show higher correlation with assets, but do not have high predictive power on their own because they provide little information about the absolute level of wealth of the given area. On the other hand, low-resolution features, despite having less correlation with assets due to lower variation, give stronger indications of the overall level of wealth. From this, it is clear that both low- and high-resolution images give useful information for asset wealth prediction. Further investigation into the information captured by each resolution and into how the low- and high-resolution features interact with each other could enhance the model's performance in the future.

Harnessing information from multiple resolutions is crucial for making full use of the satellite imagery data that is available. Our method of incorporating satellite images of multiple resolutions boosts predictive performance, which is critical for real-world deployment and can benefit a wide range of applications which utilize remote sensing data.

References

[1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

[2] Joshua Blumenstock, Gabriel Cadamuro, and Robert On. Predicting poverty and wealth from mobile phone metadata. *Science*, 350(6264):1073–1076, 2015.

[3] J. Bouvrie. Notes on convolutional neural networks. 2006.

[4] Shantayanan Devarajan. Africa's statistical tragedy. *Review of Income and Wealth*, 59(S1):S9–S15, 2013.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR abs/1310.1531*, 2013.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

[7] Lingzi Hong, Enrique Frias-Martinez, and Vanessa Frias-Martinez. Topic models to infer socio-economic maps. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] ICF International. Demographic and health surveys (various) [datasets], 2015.

[9] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[11] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[12] Neeti Pokhriyal, Wen Dong, and Venu Govindaraju. Virtual networks and poverty analysis in senegal. *arXiv preprint arXiv:1506.03401*, 2015.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition chellange. *International Journal of Computer Vision*, pages 1–42, 2014.

[14] David E Sahn and David Stifel. Exploring alternative measures of welfare in the absence of expenditure data. *Rev. Income Wealth*, 49(4):463–489, 1 December 2003.

[15] United Nations. A world that counts: Mobilising the data revolution for sustainable development. 2014.

[16] World Bank. Povcalnet online poverty analysis tool, http://iresearch.worldbank.org/povcalnet/, 2015.

[17] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.

[18] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.