Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Special Track on AI for Computational Sustainability and Human Well-being

Generating Interpretable Poverty Maps using Object Detection in Satellite Images

Kumar Ayush[1*], **Burak Uzkent**[1*], **Marshall Burke**[2], **David Lobell**[2] and **Stefano Ermon**[1]

[1]Department of Computer Science, Stanford University
[2]Department of Earth System Science, Stanford University
{kayush, buzkent}@cs.stanford.edu, mburke@stanford.edu, dlobell@stanford.edu,
ermon@cs.stanford.edu

Abstract

Accurate local-level poverty measurement is an essential task for governments and humanitarian organizations to track the progress towards improving livelihoods and distribute scarce resources. Recent computer vision advances in using satellite imagery to predict poverty have shown increasing accuracy, but they do not generate features that are interpretable to policymakers, inhibiting adoption by practitioners. Here we demonstrate an interpretable computational framework to accurately predict poverty at a local level by applying object detectors to high resolution (30cm) satellite images. Using the weighted counts of objects as features, we achieve 0.539 Pearson's r^2 in predicting village level poverty in Uganda, a 31% improvement over existing (and less interpretable) benchmarks. Feature importance and ablation analysis reveal intuitive relationships between object counts and poverty predictions. Our results suggest that interpretability does not have to come at the cost of performance, at least in this important domain.

1 Introduction

Accurate measurements of poverty and related human livelihood outcomes critically shape the decisions of governments and humanitarian organizations around the world, and the eradication of poverty remains the first of the United Nations Sustainable Development Goals [Assembly, 2015]. However, reliable local-level measurements of economic well-being are rare in many parts of the developing world. Such measurements are typically made with household surveys, which are expensive and time consuming to conduct across broad geographies, and as a result such surveys are conducted infrequently and on limited numbers of households. For example, Uganda (our study country) is one of the best-surveyed countries in Africa, but surveys occur at best every few years, and when they do occur often only survey a few hundred villages across the whole country (Fig. 1). Scaling up these ground-based surveys to cover more regions and more years would likely be prohibitively expensive for most countries in the developing world [Jerven, 2017]. The resulting lack of frequent, reliable local-level information on economic livelihoods hampers the ability of governments and other organizations to target assistance to those who need it and to understand whether such assistance is having its intended effect.

To tackle this data gap, an alternative strategy has been to try to use passively-collected data from non-traditional sources to shed light on local-level economic outcomes. Such work has shown promise in measuring certain indicators of economic livelihoods at local level. For instance, [Blumenstock *et al.*, 2015] show how features extracted from cell phone data can be used to predict asset wealth in Rwanda, and [Sheehan *et al.*, 2019] show how applying NLP techniques to Wikipedia articles can be used to predict asset wealth in multiple developing countries, and [Jean *et al.*, 2016] show how a transfer learning approach that uses coarse information from nighttime satellite images to extract features from daytime high-resolution imagery can also predict asset wealth variation across multiple African countries.

These existing approaches to using non-traditional data are promising, given that they are inexpensive and inherently scalable, but they face two main challenges that inhibit their broader adoption by policymakers. The first is the outcome being measured. While measures of asset ownership are thought to be relevant metrics for understanding longer-run household well-being [Filmer and Pritchett, 2001], official measurement of poverty requires data on consumption expenditure (i.e. the value of all goods consumed by a household over a given period), and existing methods have either not been used to predict consumption data or perform much more poorly when predicting consumption than when predicting other livelihood indicators such as asset wealth [Jean *et al.*, 2016]. Second, interpretability of model predictions is key for whether policymakers will adopt machine-learning based approaches to livelihoods measurement, and current approaches attempt to maximize predictive performance rather than interpretability. This tradeoff, central to many problems at the interface of machine learning and policy [Murdoch *et al.*, 2019], has yet to be navigated in the poverty domain.

Here we demonstrate an interpretable computational framework for predicting local-level consumption expenditure using object detection on high-resolution (30cm) daytime satellite imagery. We focus on Uganda, a country with

[*]Equal Contribution

existing high-quality ground data on consumption where performance benchmark are available. We first train a satellite imagery object detector on a publicly available, global scale object detection dataset, called xView [Lam *et al.*, 2018], which avoids location specific training and provides a more general object detection model. We then apply this detector to high resolution images taken over hundreds of villages across Uganda that were measured in an existing georeferenced household survey, and use extracted counts of detected objects as features in a final prediction of consumption expenditure. We show that not only does our approach substantially outperform previous performance benchmarks on the same task, it also yields features that are immediately and intuitively interpretable to the analyst or policy-maker.

2 Related Work

Poverty Prediction from Imagery. Multiple studies have sought to use various types of satellite imagery for local-level prediction of economic livelihoods. As already described, [Jean *et al.*, 2016] train a CNN to extract features in high-resolution daytime images using low-resolution nighttime images as labels, and then use the extracted features to predict asset wealth and consumption expenditure across five African countries. [Perez *et al.*, 2017] train a CNN to predict African asset wealth from lower-resolution (30m) multispectral satellite imagery, achieving similar performance to [Jean *et al.*, 2016]. These approaches provide accurate methods for predicting local-level asset wealth, but the CNN-extracted features used to make predictions are not easily interpretable, and performance is substantially lower when predicting consumption expenditure rather than asset wealth.

Two related papers use object detection approaches to predicting economic livelihoods from imagery. [Gebru *et al.*, 2017] show how information on the make and count of cars detected in Google Streetview imagery can be used to predict socioeconomic outcomes at local level in the US. This work is promising in a developed world context where streetview imagery is available, but challenging to employ in the developing world where such imagery is very rare, and where car ownership is uncommon. In work perhaps closest to ours, an unpublished paper by [Engstrom *et al.*, 2017] use detected objects and textural features from high-resolution imagery to predict consumption in Sri Lanka, but model performance is not validated out of sample and the object detection approach is not described.

3 Problem Setup

3.1 Poverty Estimation from Remote Sensing Data

The outcome of interest in this paper is consumption expenditure, which is the metric used to compute poverty statistics; a household or individual is said to be poor or in poverty if their measured consumption expenditure falls below a defined threshold (currently $1.90 per capita per day). Throughout the paper we use "poverty" as shorthand for "consumption expenditure", although we emphasize that the former is computed from the latter. While typical household surveys measure consumption expenditure at the household level, publicly available data typically only release geo-coordinate information at the "cluster" level – which is a village in rural areas and a neighborhood in urban areas. Efforts to predict poverty have thus focused on predicting at the cluster level (or more aggregated levels), and we do the same here. Let $\{(x_i, y_i, c_i)\}_{i=1}^{N}$ be a set of N villages surveyed, where $c_i = (c_i^{lat}, c_i^{long})$ is the latitude and longitude coordinates for cluster i, and $y_i \in \mathbb{R}$ is the corresponding average poverty index for a particular year.

For each cluster i, we can acquire high resolution satellite imagery corresponding to the survey year $x_i \in \mathcal{I} = \mathbb{R}^{W \times H \times B}$, a $W \times H$ image with B channels. Following [Jean *et al.*, 2016], our goal is to learn a regressor $f : \mathcal{I} \to \mathbb{R}$ to predict the poverty index y_i from x_i. Here our goal is to find a regressor that is both accurate and *interpretable*, where we use the latter to mean a model that provides insight to a policy community on why it makes the predictions it does in a given location.

3.2 Dataset

Socio-economic Data

The dataset comes from field Living Standards Measurement Study (LSMS) survey conducted in Uganda by the Uganda Bureau of Statistics between 2011 and 2012 [UBOS, 2012]. The LSMS survey we use here consists of data from 2,716 households in Uganda, which are grouped into unique locations called clusters. The latitude and longitude location, $c_i = (c_i^{lat}, c_i^{long})$, of a cluster $i = \{1, 2, \ldots, N\}$ is given, with noise of up to 5 km added in each direction by the surveyers to protect privacy. Individual household locations in each cluster i are also withheld to preserve anonymity. We use all $N = 320$ clusters in the survey to test the performance of our method in terms of predicting the average poverty index, y_i for a group i. For each c_i, the survey measures the poverty level by the per capital daily consumption in dollars. For simplicity, in this study, we name the per capital daily consumption in dollars as LSMS poverty score. We visualize the chosen locations on the map as well as their corresponding LSMS poverty scores in Fig. 1. From the figure, we can see that the surveyed locations are scattered near the border of states and high percentage of these locations have relatively low poverty scores.

Uganda Satellite Imagery

The satellite imagery, x_i corresponding to cluster c_i is represented by $K = 34 \times 34 = 1156$ images of $W = 1000 \times H = 1000$ pixels with $B = 3$ channels, arranged in a 34×34 square grid. This corresponds to a 10 km $\times$ 10 km spatial neighborhood centered at c_i. We consider a large neighborhood to deal with the noise in the cluster coordinates. The images come from DigitalGlobe satellites with three bands (RGB) and 30cm pixel resolution. Fig. 1 illustrates an example cluster from Uganda. Formally, we represent all the images corresponding to c_i as a sequence of K tiles as $x_i = \{x_i^j\}_{j=1}^{K}$.

4 Fine-grained Detection on Satellite Images

Contrary to existing methods for poverty mapping which perform end-to-end learning [Jean *et al.*, 2016; Sheehan *et al.*,

	Building	Fixed-Wing Aircraft	Passenger Vehicle	Truck	Railway Vehicle	Maritime Vessel	Engineering Vehicle	Helipad	Vehicle Lot	Construction Site
AP	0.40	0.59	0.42	0.27	0.39	0.24	0.17	0.0	0.012	0.0003
AR	0.62	0.65	0.76	0.56	0.49	0.47	0.37	0.0	0.06	0.006

Table 1: Class wise performance (average precision and recall) of YOLOv3 when trained using parent level classes (10 classes).

2019; Perez *et al.*, 2017], we use an intermediate object detection phase to first obtain interpretable features for subsequent poverty prediction. However, we do not have object annotations for satellite images from Uganda. Therefore, we perform transfer learning by training an object detector on a different but related source dataset $\mathcal{D}^s$.

4.1 Object Detection Dataset

We use xView [Lam *et al.*, 2018], as our source dataset. It is one of the largest and most diverse publicly available overhead imagery datasets for object detection. It covers over $1,400$ km^2 of the earth's surface, with 60 classes and approximately 1 million labeled objects. The satellite images are collected from DigitalGlobe satellites at 0.3 m GSD, aligning with the GSD of our target region satellite imagery $\{x_i\}_{i=1}^N$. Moreover, xView uses a tree-structured ontology of classes. The classes are organized hierarchically similar to [Deng *et al.*, 2009; Lai *et al.*, 2011] where children are more specific than their parents (e.g., *fixed-wing aircraft* as a parent of *small aircraft* and *cargo plane*). Overall, there are 60 child classes and 10 parent classes.

4.2 Training the Object Detector

Models. Since we work on very large tiles ($\sim$3000$\times$3000 pixels), we only consider single stage detectors. Considering the trade off between run-time performance and accuracy on small objects, YOLOv3 [Redmon and Farhadi, 2018] outperforms other single stage detectors [Fu *et al.*, 2017; Liu *et al.*, 2016] and performs almost on par with RetinaNet [Lin *et al.*, 2017b] but 3.8 $\times$ faster [Redmon and Farhadi, 2018] on small objects while running significantly faster than two-stage detectors [Lin *et al.*, 2017a; Shrivastava *et al.*, 2016]. Therefore, we use YOLOv3 object detector with a DarkNet53 [Redmon and Farhadi, 2018] backbone architecture.

Dataset Preparation. The xView dataset consists of 847 large images (roughly 3000×3000 px). YOLOv3 is usually used with an input image size of 416×416 px. Therefore, we randomly chip 416×416 px tiles from the xView images and discard tiles without any object of interest. This process results in 36996 such tiles of which we use 30736 tiles for training and 6260 tiles for testing.

Training and Evaluation. We use the standard per-class average precision, mean average precision (mAP), and per-class recall, mean average recall (mAR) metrics [Redmon and Farhadi, 2018; Lin *et al.*, 2017b] to evaluate our trained object detector. We fine-tune the weights of the YOLOv3 model, pre-trained on the ImageNet, using the training split of the xView dataset. Since xView has an ontology of parent and child level classes, we train two YOLOv3 object detectors using parent level and child level classes seperately.

After training the models, we validate their performance on the test set of xView. The detector trained using parent level classes (10 classes) achieves mAP of 0.248 and mAR of 0.42. On the other hand, the one trained on child classes achieves mAP of 0.082 and mAR of 0.163. Table 1 shows the class-wise performance of the parent-level object detector on the test set. For comparison, Lam *et al.* report 0.14 mAP, but they use a separate validation and test set in addition to the training set (which are not publicly available) so the models are not directly comparable. While not state of the art, our detector reliably identifies objects, especially at the parent level.

4.3 Object Detection on Uganda Satellite Images

As described in Section 3.2, each x_i is represented by a set of K images, $\{x_i^j\}_{j=1}^K$. Each 1000×1000 px tile (i.e. x_i^j) is further chipped into 9 416×416 px small tiles (with overlap of 124 px) and fed to YOLOv3.

Although the presence of objects across tile borders could decrease performance, this method is highly parallelizable and enables us to scale to very large regions. We perform object detection on $320 \times 1156 \times 9$ chips (more than 3 million images), which takes about a day and a half using 4 NVIDIA 1080Ti GPUs. In total, we detect 768404 objects. Each detection is denoted by a tuple (x_c, y_c, w, h, l, s), where x_c and y_c represent the center coordinates of the bounding box, w and h represent the width and height of the bounding box, l and s represent the object class label and class confidence score. In Section 5.1, we explain how we use these details to create interpretable features. Additionally, we experiment with object detections obtained at different confidence thresholds which we discuss in Section 6.1.

Transfer performance in Uganda. The absence of ground truth object annotations for our Uganda imagery $\{x_i^j\}_{j=1}^K$ prevents us from quantitatively measuring the detector's performance on Uganda satellite imagery. However, we manually annotated 10 images from the Uganda dataset together with the detected bounding boxes to measure the detector's performance on building and truck classes. We found that the detector achieves about 50%, and 45% AR for Building and Truck which is slightly lower than the AR scores for the same classes on the xView test set. We attribute this slight difference to the problem of domain shift and we plan to address this problem via domain adaptation in a future work. To qualitatively test the robustness of our xView-trained object detector, we also visualize its performance on two representative tiles in Fig. 2. The detection results prove the effectiveness of transferring the YOLOv3 model to DigitalGlobe imagery it has not been trained on.

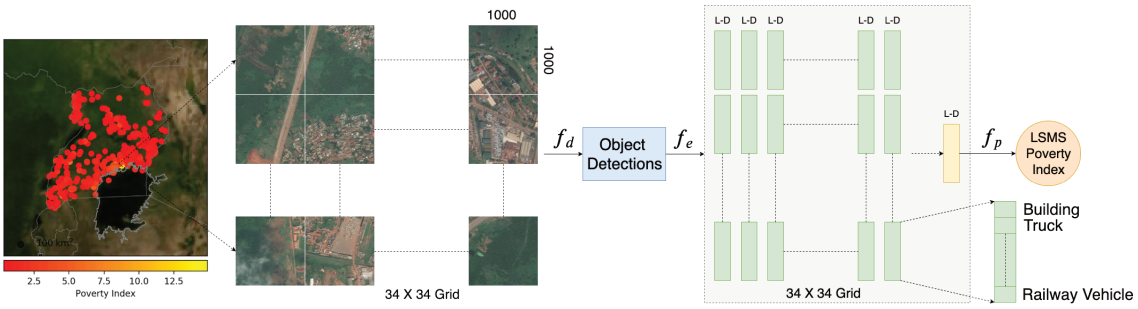


Figure 1: Pipeline of the proposed approach. For each cluster we acquire 1156 images, arranged in a 34×34 grid, where each image is an RGB image of 1000×1000 px. We run an object detector on its 10×10 km^2 neighborhood and obtain the object counts for each xView class as a L-dimensional categorical feature vector. This is done by running the detector on every single image in a cluster, resulting in a $34 \times 34 \times L$ dimensional feature vector. Finally, we perform summation across the first two dimensions and get the feature vector representing the cluster, with each dimension containing the object counts corresponding to an object class. Given the cluster level feature vector, we regress the LSMS poverty score.

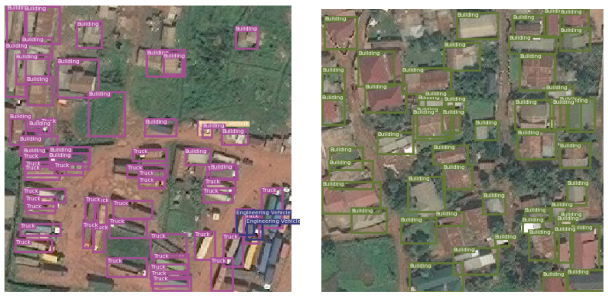


Figure 2: Sample detection results from Uganda. Zoom-in is recommended to visualize the bounding box classes.

5 Fine-level Poverty Mapping

5.1 Feature Extraction from Clusters

Our object detection pipeline outputs n_i^j object detections for each tile x_i^j of x_i. We use the n_i^j object detections to generate a L-dimensional vector, $\mathbf{v}_i^j \in \mathbb{R}^L$ (where L is the number of object labels/classes), by counting the number of detected objects in each class with each object weighted by its confidence score or size or their combination (details below). This process results in K L-dimensional vectors $\mathbf{v}_i = \{\mathbf{v}_i^j\}_{k=1}^K$. Finally, we aggregate these K vectors into a single L-dimensional categorical feature vector $\mathbf{m}_i$ by summing over tiles: $\mathbf{m}_i = \sum_{j=1}^K \mathbf{v}_i^j$. While many other options are possible, in this work we explore four types of features:

Counts. *Raw object counts corresponding to each class.* Here, each dimension represents an object class and contains the number of objects detected corresponding to that class.

Confidence×Counts. *Each detected object is weighted by its class confidence score.* The intuition is to reduce the contributions of less confident detections. Here each dimension corresponds to the sum of class confidence scores of the detected objects of that class.

Size×Counts. *Each detected object is weighted by its bounding box area.* We posit that weighting based on area coverage of an object class can be an important factor. For example, an area with 10 big buildings might have a different wealth level than an area with 10 small buildings. Each dimension in $\mathbf{m}_i$ contains the sum of areas of the bounding boxes of the detected objects of that class.

(Confidence, Size)×Counts. *Each detected object is weighted by its class confidence score and the area of its bounding box.* We concatenate the *Confidence* and *Size* based features to create a $2L$-dimensional vector.

5.2 Models, Training and Evaluation

Given the cluster level categorical feature vector, $\mathbf{m}_i$, we estimate its poverty index, y_i with a regression model. Since we value interpretability, we consider Gradient Boosting Decision Trees, Linear Regression, Ridge Regression, and Lasso Regression. As we regress directly on the LSMS poverty index, we quantify the performance of our model using the square of the Pearson correlation coefficient (Pearson's r^2). Pearson's r^2, provides a measure of how well observed outcomes are replicated by the model. This metric was chosen so that comparative analysis could be performed with previous literature [Jean *et al.*, 2016]. Pearson's r^2 is invariant under separate changes in scale between the two variables. This allows the metric to provide insight into the ability of the model to distinguish between poverty levels. This is relevant for many downstream poverty tasks, including the distribution of program aid under a fixed budget (where aid is disbursed to households starting with the poorest, until the budget is exhausted), or in the evaluation of anti-poverty programs, where outcomes are often measured in terms of percentage changes in the poverty metric. Due to small size of the dataset, we use a Leave-one-out cross validation (LOOCV) strategy. Since nearby clusters could have some geographic overlap, we remove clusters which are overlapping with the test cluster from the train split to avoid leaking information to the test point.

6 Experiments

6.1 Poverty Mapping Results

Quantitative Analysis. Table 2 shows the results of LSMS poverty prediction in Uganda. The detections are obtained

using a 0.6 confidence threshold (the effect of this hyperparameter is evaluated below). The best result of 0.539 Pearson's r^2 is obtained using GBDT trained on parent level *object counts* features (red color entry). A scatter plot of GBDT predictions v.s. ground truth is shown in Fig. 3. It can be seen that our GBDT model can explain a large fraction of the variance in terms of object counts automatically identified in high resolution satellite images. To the best of our knowledge, this is the first time this capability has been shown with a rigorous and reproducible out-of-sample evaluation (see however the related but unpublished paper by Engstrom *et al.*).

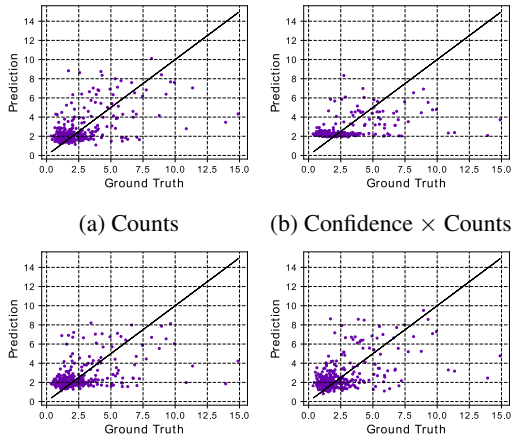
We observe that GBDT performs consistently better than other regression models across the four features we consider. As seen in Table 2, object detection based features deliver positive r^2 with a simple linear regression method which suggests that they have positive correlation with LSMS poverty scores. However, the main drawback of linear regression against GBDT is that it predicts negative values, which is not reasonable as poverty indices are non-negative. In general, the features are useful, but powerful regression models are still required to achieve better performance.

We also find that child-level object detections can perform better than the coarser ones (second and third best) in some cases. This is likely because although they convey more information, detection and classification is harder and less accurate at the finer level (see Section 4.2). Additionally, parent level features are more suited for interpretability, due to household level descriptions, which we show later.

— Best — Second Best — Third Best

Features/Method	GBDT		Linear		Lasso		Ridge	
	Parent	Child	Parent	Child	Parent	Child	Parent	Child
Counts	0.539	0.508	0.311	0.324	0.312	0.46	0.311	0.329
Confidence × Counts	0.466	0.485	0.305	0.398	0.305	0.461	0.305	0.409
Size × Counts	0.455	0.535	0.363	0.47	0.363	0.476	0.363	0.47
(Conf., Size) × Counts	0.495	0.516	0.411	0.369	0.418	0.343	0.411	0.476

Table 2: LSMS poverty score prediction results in Pearson's r^2 using parent level features (YOLOv3 trained on 10 classes) and child level features (YOLOv3 trained on 60 classes).



(a) Counts (b) Confidence × Counts

(c) Size × Counts (d) (Conf., Size) × Counts

Figure 3: Regression result of GBDT using parent level counts.

Comparison to Baselines and State-of-the-Art. We compare our method with two baselines and a state-of-the-art method: (a) **NL-CNN** where we regress the LSMS poverty scores using a 2-layer CNN with Nightlight Images (48×48 px) representing the clusters in Uganda as input, (b) **RGB-CNN** where we regress the LSMS poverty scores using ImageNet [Deng *et al.*, 2009] pretrained ResNet-18 [He *et al.*, 2016] model with central tile representing c_i as input, and (c) **Transfer Learning with Nightlights**, [Jean *et al.*, 2016] proposed a transfer learning approach where nighttime light intensities are used as a data-rich proxy.

Results are shown in Table 3. Our model substantially outperforms all three baselines, including published state-of-the-art results on the same task in [Jean *et al.*, 2016]. We similarly outperform the NL-CNN baseline, a simpler version of which (scalar nightlights) is often used for impact evaluation in policy work [Donaldson and Storeygard, 2016]. Finally, the performance of the RGB-CNN baseline reveals the limitation of directly regressing CNNs on daytime images, at least in our setting with small numbers of labels. As discussed below, these performance improvements do not come at the cost of interpretability – rather, our model predictions are much more interpretable than each of these three baselines.

Method	RGB-CNN	NL-CNN	[Jean *et al.*, 2016]	Ours
r^2	0.04	0.39	0.41	**0.54**

Table 3: Comparison with baseline and state-of-the-art methods.

Impact of Detector's Confidence Threshold. Finally, we analyze the effect of confidence threshold for object detector on the poverty prediction task in Fig. 4. We observe that when considering only *Counts* features, we get the best performance at 0.6 threshold. However, even for very small thresholds, we achieve around 0.3-0.5 Pearson's r^2 scores. We explore this finding in Fig. 4b, and observe that the *ratio of classes in terms of number of bounding boxes remain similar* across different thresholds. These results imply that the ratio of object counts is perhaps more useful than simply the counts themselves – an insight also consistent with the substantial performance boost from GBT over unregularized and regularized linear models in Table 1.

6.2 Interpretability

Existing approaches to poverty prediction using unstructured data from satellites or other sources have understandably

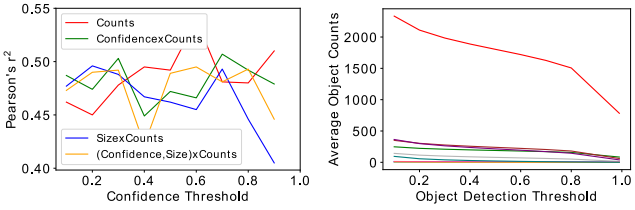


Figure 4: **Left:** Regression results (GBDT) using object detection features (parent level classes) at different confidence thresholds. **Right:** Average object counts across clusters for each parent class (see Table 1 for color coding) at difference confidence thresholds.

sought to maximize predictive performance [Jean *et al.*, 2016; Perez *et al.*, 2017; Sheehan *et al.*, 2019], but this has come at the cost of interpretability, as most of the extracted features used for prediction do not have obvious semantic meaning. While no quantitative data have been collected on the topic, our personal experience on multiple continents over many years is that the lack of interpretability of CNN-based poverty predictions can make policymakers understandably reluctant to trust these predictions and to use them in decision-making. Enhancing the interpretability of ML-based approaches more broadly is thought to be a key component of successful application in many policy domains [Doshi-Velez and Kim, 2017].

Relative to an end-to-end deep learning approach, our two-step approach provides categorical features that can be easily interpreted. We now explore whether these features also have an intuitive mapping to poverty outcomes in three analyses.

Explanations via SHAP. In this section, we explain the effect of individual features on poverty score predictions using SHAP (SHapley Additive exPlanations) [Lundberg and Lee, 2017]. SHAP is a game theoretic approach to explain the output of any machine learning model. We particularly use TreeSHAP [Lundberg *et al.*, 2018] which is a variant of SHAP for tree-based machine learning models. TreeSHAP significantly improves the interpretability of tree-based models through a) a polynomial time algorithm to compute optimal explanations based on game theory, b) explanations that directly measure local feature interaction effects, and c) tools for understanding global model structure based on combining many local explanations of each prediction.

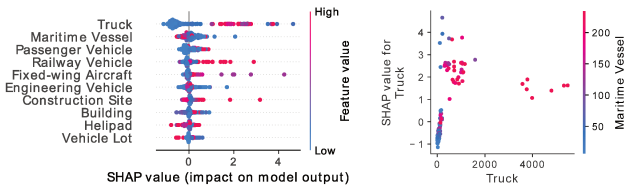


Figure 5: **Left**: Summary of the effects of all the features. **Right**: Dependence plot showing the effect of a single feature across the whole dataset. In both figures, the color represents the feature value (red is high, blue is low)

To get an overview of which features are most important for a model we plot the SHAP values of every feature for every sample. The plot in Figure 5 (left) sorts features by the sum of SHAP value magnitudes over all samples, and uses SHAP values to show the distribution of the impacts each feature has on the model output. The color represents the feature value (red high, blue low). We find that *Truck* tends to have a high impact on the model's output. Higher *#Trucks* pushes the output to a higher value and low *#Trucks* has a negative impact on the output, thereby lowering the predicted value.

To understand how the Truck feature effects the output of the model we plot the SHAP value of *Truck* feature vs. the value of the Truck feature for all the examples in the dataset. Since SHAP values represent a feature's responsibility for a change in the model output, the plot in Figure 5 (right) represents the change in predicted poverty score as *Truck* feature changes and also reveals the interaction between *Truck* feature and *Maritime Vessel* feature. We find that for small *#Trucks*, low *#Maritime Vessels* decreases the *Truck* SHAP value. This can be seen from the set of points that form a vertical line (towards bottom left) where the color changes from blue (low *#Maritime Vessels*) to red (high *#Maritime Vessels*) as Truck SHAP value increases.

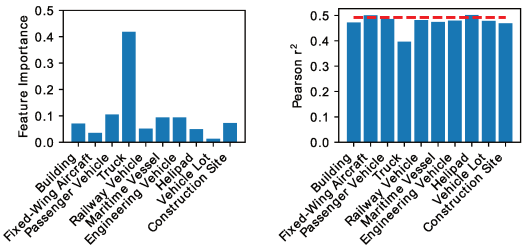


Figure 6: **Left**: Feature Importance of parent classes in the GBDT model. **Right**: Ablation analysis where the red line represents the GBDT's performance when including all the parent classes.

Feature Importance. We also plot the sum of SHAP value magnitudes over all samples for the various features (feature importance). Figure 6 (left) shows the importance of the 10 features (parent level features) in poverty prediction. *Truck* has the highest importance. It is followed by *Passenger Vehicle*, *Maritime Vessel*, and *Engg. Vehicle*.

Ablation Analysis. Finally, we run an ablation study by training the regression model using all the categorical features in the train set and at test time we eliminate a particular feature by collapsing it to zero. We perform this ablation study with the parent level features as it provides better interpretability. Consistent with the feature importance scores, in Figure 6 we find that when *Truck* feature is eliminated at test time, the Pearson's r^2 value is impacted most.

7 Conclusion

In this work, we attempt to predict consumption expenditure from high resolution satellite images. We propose an efficient, explainable, and transferable method that combines object detection and regression. This model achieves a Pearson's r^2 of 0.54 in predicting village level consumption expenditure in Uganda, even when the provided locations are affected by noise (for privacy reasons) and the overall number of labels is small ($\sim$300). The presence of trucks appears to be particularly useful for measuring local scale poverty in our setting. We also demonstrate that our features achieve positive results even with simple linear regression models. Our results offer a promising approach for generating interpretable poverty predictions for important livelihood outcomes, even in settings with limited training data.

Acknowledgements

This research was supported in part by Stanford's Data for Development Initiative and NSF grants #1651565 and #1522054.

References

[Assembly, 2015] General Assembly. Sustainable development goals. *SDGs Transform Our World*, 2030, 2015.

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

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Donaldson and Storeygard, 2016] Dave Donaldson and Adam Storeygard. The view from above: Applications of satellite data in economics. *Journal of Economic Perspectives*, 30(4):171–98, 2016.

[Doshi-Velez and Kim, 2017] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[Engstrom *et al.*, 2017] Ryan Engstrom, Jonathan Hersh, and David Newhouse. Poverty from space: Using high-resolution satellite imagery for estimating economic well-being, 2017.

[Filmer and Pritchett, 2001] Deon Filmer and Lant H Pritchett. Estimating wealth effects without expenditure data—or tears: an application to educational enrollments in states of india. *Demography*, 38(1):115–132, 2001.

[Fu *et al.*, 2017] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[Gebru *et al.*, 2017] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. *Proceedings of the National Academy of Sciences*, 114(50):13108–13113, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Jean *et al.*, 2016] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[Jerven, 2017] Morten Jerven. How much will a data revolution in development cost? In *Forum for Development Studies*, volume 44, pages 31–50. Taylor & Francis, 2017.

[Lai *et al.*, 2011] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011.

[Lam *et al.*, 2018] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[Lin *et al.*, 2017a] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[Lin *et al.*, 2017b] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[Lundberg *et al.*, 2018] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[Murdoch *et al.*, 2019] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[Perez *et al.*, 2017] Anthony Perez, Christopher Yeh, George Azzari, Marshall Burke, David Lobell, and Stefano Ermon. Poverty prediction with public landsat 7 satellite imagery and machine learning. *arXiv preprint arXiv:1711.03654*, 2017.

[Redmon and Farhadi, 2018] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[Sheehan *et al.*, 2019] Evan Sheehan, Chenlin Meng, Matthew Tan, Burak Uzkent, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Predicting economic development using geolocated wikipedia articles. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2698–2706, 2019.

[Shrivastava *et al.*, 2016] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.

[UBOS, 2012] Uganda Bureau of Statistics UBOS. Uganda national panel survey 2011/2012. *Uganda*, 2012.