# Transfer Learning from Deep Features for Remote Sensing and Poverty Mapping

**Michael Xie,   Neal Jean,   Marshall Burke,   David Lobell,** and **Stefano Ermon**

Department of Computer Science, Stanford University
{xie, nealjean, ermon}@cs.stanford.edu
Department of Earth System Science, Stanford University
{mburke,dlobell}@stanford.edu

## Abstract

The lack of reliable data in developing countries is a major obstacle to sustainable development, food security, and disaster relief. Poverty data, for example, is typically scarce, sparse in coverage, and labor-intensive to obtain. Remote sensing data such as high-resolution satellite imagery, on the other hand, is becoming increasingly available and inexpensive. Unfortunately, such data is highly unstructured and currently no techniques exist to automatically extract useful insights to inform policy decisions and help direct humanitarian efforts. We propose a novel machine learning approach to extract large-scale socioeconomic indicators from high-resolution satellite imagery. The main challenge is that training data is very scarce, making it difficult to apply modern techniques such as Convolutional Neural Networks (CNN). We therefore propose a transfer learning approach where nighttime light intensities are used as a data-rich proxy. We train a fully convolutional CNN model to predict nighttime lights from daytime imagery, simultaneously learning features that are useful for poverty prediction. The model learns filters identifying different terrains and man-made structures, including roads, buildings, and farmlands, without any supervision beyond nighttime lights. We demonstrate that these learned features are highly informative for poverty mapping, even approaching the predictive performance of survey data collected in the field.

## Introduction

New technologies fueling the Big Data revolution are creating unprecedented opportunities for designing, monitoring, and evaluating policy decisions and for directing humanitarian efforts   (Abelson, Varshney, and Sun 2014; Varshney et al. 2015). However, while rich countries are being flooded with data, developing countries are suffering from data drought. A new data divide is emerging, with huge differences in the quantity and quality of data available. For example, some countries have not taken a census in decades, and in the past five years an estimated 230 million births have gone unrecorded (Independent Expert Advisory Group Secretariat 2014). Even high-profile initiatives such as the Millennium Development Goals (MDGs) are affected (United Nations 2015). Progress based on poverty and infant mortality rate targets can be difficult to track. Often, poverty measures must be inferred from small-scale and expensive household surveys, effectively rendering many of the poorest people invisible.

Remote sensing, particularly satellite imagery, is perhaps the only cost-effective technology able to provide data at a global scale. Within ten years, commercial services are expected to provide sub-meter resolution images everywhere at a fraction of current costs (Murthy et al. 2014). This level of temporal and spatial resolution could provide a wealth of data towards sustainable development. Unfortunately, this raw data is also highly unstructured, making it difficult to extract actionable insights at scale.

In this paper, we propose a machine learning approach for extracting socioeconomic indicators from raw satellite imagery. In the past five years, deep learning approaches applied to large-scale datasets such as ImageNet have revolutionized the field of computer vision, leading to dramatic improvements in fundamental tasks such as object recognition (Russakovsky et al. 2014). However, the use of contemporary techniques for the analysis of remote sensing imagery is still largely unexplored. Modern approaches such as Convolutional Neural Networks (CNN) can, in principle, be directly applied to extract socioeconomic factors, but the primary challenge is a lack of training data. While such data is readily available in the United States and other developed nations, it is extremely scarce in Africa where these techniques would be most useful.

We overcome this lack of training data by using a sequence of transfer learning steps and a convolutional neural network model. The idea is to leverage available datasets such as ImageNet to extract features and high-level representations that are useful for the task of interest, i.e., extracting socioeconomic data for poverty mapping. Similar strategies have proven quite successful in the past. For example, image features from the `Overfeat` network trained on ImageNet for object classification achieved state-of-the-art results on tasks such as fine-grained recognition, image retrieval, and attribute detection (Razavian et al. 2014).

Pre-training on ImageNet is useful for learning low-level features such as edges. However, ImageNet consists only of object-centric images, while satellite imagery is captured from an aerial, bird's-eye view. We therefore employ a second transfer learning step, where nighttime light intensities are used as a proxy for economic activity. Specifically, we

start with a CNN model pre-trained for object classification on ImageNet and learn a modified network that predicts nighttime light intensities from daytime imagery. To address the trade-off between fixed image size and information loss from image scaling, we use a fully convolutional model that takes advantage of the full satellite image. We show that transfer learning is successful in learning features relevant not only for nighttime light prediction but also for poverty mapping. For instance, the model learns filters identifying man-made structures such as roads, urban areas, and fields without any supervision beyond nighttime lights, i.e., without any labeled examples of roads or urban areas (Figure 2). We demonstrate that these features are highly informative for poverty mapping and capable of approaching the predictive performance of survey data collected in the field.

## Problem Setup

We begin by reviewing transfer learning and convolutional neural networks, the building blocks of our approach.

### Transfer Learning

We formalize transfer learning as in (Pan and Yang 2010): A *domain* $\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathcal{X})$. Given a domain, a *task* $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ consists of a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$ which models $P(y|x)$ for $y \in \mathcal{Y}$ and $x \in \mathcal{X}$. Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, and a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, *transfer learning* aims to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{T}_T$ using the knowledge from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, $\mathcal{T}_S \neq \mathcal{T}_T$, or both. Transfer learning is particularly relevant when, given labeled *source domain data* $D_S$ and *target domain data* $D_T$, we find that $|D_T| \ll |D_S|$.

In our setting, we are interested in more than two related learning tasks. We generalize the formalism by representing the multiple source-target relationships as a *transfer learning graph*. First, we define a *transfer learning problem* $\mathcal{P} = (\mathcal{D}, \mathcal{T})$ as a domain-task pair. The *transfer learning graph* is then defined as follows: A transfer learning graph $G = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph where vertices $\mathcal{V} = \{\mathcal{P}_1, \cdots, \mathcal{P}_v\}$ are transfer learning problems and $\mathcal{E} = \{(\mathcal{P}_{i_1}, \mathcal{P}_{j_1}), \cdots, (\mathcal{P}_{i_e}, \mathcal{P}_{j_e})\}$ is an edge set. For each transfer learning problem $\mathcal{P}_i = (\mathcal{D}_i, \mathcal{T}_i) \in \mathcal{V}$, the aim is to improve the learning of the target predictive function $f_i(\cdot)$ in $\mathcal{T}_i$ using the knowledge in $\cup_{(j,i) \in \mathcal{E}} \mathcal{P}_j$.

### Convolutional Neural Networks

Deep learning approaches are based on automatically learning nested, hierarchical representations of data. Deep feedforward neural networks are the typical example of deep learning models. Convolutional Neural Networks (CNN) include convolutional operations over the input and are designed specifically for vision tasks. Convolutional filters are useful for encoding translation invariance, a key concept for discovering useful features in images (Bouvrie 2006).

A CNN is a general function approximator defined by a set of convolutional and fully connected layers ordered such that the output of one layer is the input of the next. For image data, the first layers of the network typically learn low-level features such as edges and corners, and further layers learn high-level features such as textures and objects (Zeiler and Fergus 2013). Taken as a whole, a CNN is a mapping from tensors to feature vectors, which become the input for a final classifier. A typical convolutional layer maps a tensor $x \in \mathbb{R}^{h \times w \times d}$ to $g_i \in \mathbb{R}^{\hat{h} \times \hat{w} \times \hat{d}}$ such that

$$g_i = p_i(f_i(W_i * x + b_i)),$$

where for the $i$-th convolutional layer, $W_i \in \mathbb{R}^{l \times l \times \hat{d}}$ is a tensor of $\hat{d}$ convolutional filter weights of size $l \times l$, $(*)$ is the 2-dimensional convolution operator over the last two dimensions of the inputs, $b_i$ is a bias term, $f_i$ is an elementwise nonlinearity function (e.g., a rectified linear unit or ReLU), and $p_i$ is a pooling function. The output dimensions $\hat{h}$ and $\hat{w}$ depend on the stride and zero-padding parameters of the layer, which control how the convolutional filters slide across the input. For the first convolutional layer, the input dimensions $h$, $w$, and $d$ can be interpreted as height, width, and number of color channels of an input image, respectively.

In addition to convolutional layers, most CNN models have fully connected layers in the final layers of the network. Fully connected layers map an unrolled version of the input $\hat{x} \in \mathbb{R}^{hwd}$, which is a one-dimensional vector of the elements of a tensor $x \in \mathbb{R}^{h \times w \times d}$, to an output $g_i \in \mathbb{R}^k$ such that

$$g_i = f_i(W_i \hat{x} + b_i),$$

where $W_i \in \mathbb{R}^{k \times hwd}$ is a weight matrix, $b_i$ is a bias term, and $f_i$ is typically a ReLU nonlinearity function. The fully connected layers encode the input examples as feature vectors, which are used as inputs to a final classifier. Since the fully connected layer looks at the entire input at once, these feature vectors "summarize" the input into a feature vector for classification. The model is trained end-to-end using minibatch gradient descent and backpropagation.

After training, the output of the final fully connected layer can be interpreted as an encoding of the input as a feature vector that facilitates classification. These features often represent complex compositions of the lower-level features extracted by the previous layers (e.g., edges and corners) and can range from grid patterns to animal faces (Zeiler and Fergus 2013; Le et al. 2012).

### Combining Transfer Learning and Deep Learning

The low-level and high-level features learned by a CNN on a source domain can often be transferred to augment learning in a different but related target domain. For target problems with abundant data, we can transfer low-level features, such as edges and corners, and learn new high-level features specific to the target problem. For target problems with limited amounts of data, learning new high-level features is difficult. However, if the source and target domain are sufficiently similar, the feature representation learned by the CNN on the source task can also be used for the target problem. Deep features extracted from CNNs trained on large annotated datasets of images have been used as generic features very

effectively for a wide range of vision tasks (Donahue et al. 2013; Oquab et al. 2014).

## Transfer Learning for Poverty Mapping

In our approach to poverty mapping using satellite imagery, we construct a linear chain transfer learning graph with $\mathcal{V} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$ and $\mathcal{E} = \{(\mathcal{P}_1, \mathcal{P}_2), (\mathcal{P}_2, \mathcal{P}_3)\}$. The first transfer learning problem $\mathcal{P}_1$ is object recognition on ImageNet (Russakovsky et al. 2014); the second problem $\mathcal{P}_2$ is predicting nighttime light intensity from daytime satellite imagery; the third problem $\mathcal{P}_3$ is predicting poverty from daytime satellite imagery. Recognizing the differences between ImageNet data and satellite imagery, we use the intermediate problem $\mathcal{P}_2$ to learn about the bird's-eye viewpoint of satellite imagery and extract features relevant to socioeconomic development.

### ImageNet to Nighttime Lights

ImageNet is an object classification image dataset of over 14 million images with 1000 class labels that, along with CNN models, have fueled major breakthroughs in many vision tasks (Russakovsky et al. 2014). CNN models trained on the ImageNet dataset are recognized as good generic feature extractors, with low-level and mid-level features such as edges and corners that are able to generalize to many new tasks (Donahue et al. 2013; Oquab et al. 2014). Our goal is to transfer knowledge from the ImageNet object recognition challenge ($\mathcal{P}_1$) to the target problem of predicting nighttime light intensity from daytime satellite imagery ($\mathcal{P}_2$).

In $\mathcal{P}_1$, we have an object classification problem with source domain data $D_1 = \{(x_{1i}, y_{1i})\}$ from ImageNet that consists of natural images $x_{1i} \in \mathcal{X}_1$ and object class labels. In $\mathcal{P}_2$, we have a nighttime light intensity prediction problem with target domain data $D_2 = \{(x_{2i}, y_{2i})\}$ that consists of daytime satellite images $x_{2i} \in \mathcal{X}_2$ and nighttime light intensity labels. Although satellite data is still in the space of image data, satellite imagery presents information from a bird's-eye view and at a much different scale than the object-centric ImageNet dataset ($P(\mathcal{X}_1) \neq P(\mathcal{X}_2)$). Previous work in domains with images fundamentally different from normal "human-eye view" images typically resort to curating a new, specific dataset such as Places205 (Zhou et al. 2014). In contrast, our transfer learning approach does not require human annotation and is much more scalable. Additionally, unsupervised approaches such as autoencoders may waste representational capacity on irrelevant features, while the nighttime light labels guide learning towards features relevant to wealth and economic development.

The National Oceanic and Atmospheric Administration (NOAA) provides annual nighttime images of the world with 30 arc-second resolution, or about 1 square kilometer (NOAA National Geophysical Data Center 2014). The light intensity values are averaged and denoised for each year to ensure that ephemeral light sources do not affect the data.

The nighttime light dataset $D_2$ is constructed as follows: The Demographic Health Survey (DHS) Program conducts nationally representative surveys in Africa that focus mainly on health outcomes (ICF International 2015). Predicting
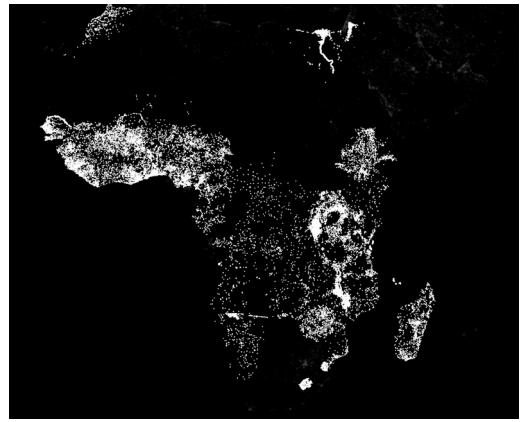


Figure 1: Locations (in white) of 330,000 sampled daytime images near DHS survey locations for the nighttime light intensity prediction problem.

health outcomes is beyond the scope of this paper; however, the DHS surveys offer the most comprehensive data available for Africa. Thus, we use DHS survey locations as guidelines for sampling training images (see Figure 1). Images in $D_2$ are daytime satellite images randomly sampled near DHS survey locations in Africa. Satellite images are downloaded using the Google Static Maps API, each with $400 \times 400$ pixels at zoom level 16, resulting in images similar in size to pixels in the NOAA nighttime lights data. The aggregate dataset $D_2$ consists of over 330,000 images, each labeled with an integer nighttime light intensity value ranging from 0 to $63^1$. We further subsample and bin the data using a Gaussian mixture model, as detailed in the companion technical report (Xie et al. 2015).

### Nighttime Lights to Poverty Estimation

The final and most important learning task $\mathcal{P}_3$ is that of predicting poverty from satellite imagery, for which we have very limited training data. Our goal is to transfer knowledge from $\mathcal{P}_2$, a data-rich problem, to $\mathcal{P}_3$.

The target domain data $D_3 = \{(x_{3i}, y_{3i})\}$ consists of satellite images $x_{3i} \in \mathcal{X}_3$ from the feature space of satellite images of Uganda and a **limited number** of poverty labels $y_{3i} \in \mathcal{Y}_3$, detailed below. The source data is $D_2$, the nighttime lights data. Here, the input feature space of images is similar in both the source and target domains, drawn from a similar distribution of images (satellite images) from related areas (Africa and Uganda), implying that $\mathcal{X}_2 = \mathcal{X}_3$, $P(\mathcal{X}_2) \approx P(\mathcal{X}_3)$. The source (lights) and target (poverty) tasks both have economic elements, but are quite different.

The poverty training data $D_3$ relies on the Living Standards Measurement Study (LSMS) survey conducted in Uganda by the Uganda Bureau of Statistics between 2011 and 2012 (Uganda Bureau of Statistics 2012). The LSMS

---

[1]Nighttime light intensities are from 2013, while the daytime satellite images are from 2015. We assume that the areas under study have not changed significantly in this two-year period, but this temporal mismatch is a potential source of error.

survey consists of data from 2,716 households in Uganda, which are grouped into 643 unique location groups. The average latitude and longitude location of the households within each group is given, with added noise of up to 5km in each direction. Individual household locations are withheld to preserve anonymity. In addition, each household has a binary poverty label based on expenditure data from the survey. We use the majority poverty classification of households in each group as the overall location group poverty label. For a given group, we sample approximately 100 1km×1km images tiling a 10km × 10km area centered at the average household location as input. This defines the probability distribution $P(\mathcal{X}_3)$ of the input images for the poverty classification problem $\mathcal{P}_3$.

## Predicting Nighttime Light Intensity

Our first goal is to transfer knowledge from the ImageNet object recognition task to the nighttime light intensity prediction problem. We start with a CNN model with parameters trained on ImageNet, then modify the network to adapt it to the new task (i.e., change the classifier on the last layer to reflect the new nighttime light prediction task). We train on the new task using SGD with momentum, using ImageNet parameters as initialization to achieve knowledge transfer.

We choose the VGG F model trained on ImageNet as the starting CNN model  (Chatfield et al. 2014). The VGG F model has 8 convolutional and fully connected layers. Like many other ImageNet models, the VGG F model accepts a fixed input image size of $224 \times 224$ pixels. Input images in $D_2$, however, are $400 \times 400$ pixels, corresponding to the resolution of the nighttime lights data.

We consider two ways of adapting the original VGG F network. The first approach is to keep the structure of the network (except for the final classifier) and crop the input to $224 \times 224$ pixels (**random cropping**). This is a reasonable approach, as the original model was trained by cropping $224 \times 224$ images from a larger $256 \times 256$ image (Chatfield et al. 2014). Ideally, we would evaluate the network at multiple crops of the $400 \times 400$ input and average the predictions to leverage the context of the entire input image. However, doing this explicitly with one forward pass for each crop would be too costly. Alternatively, if we allow the multiple crops of the image to overlap, we can use a convolution to compute scores for each crop simultaneously, gaining speed by reusing filter outputs at all layers. We therefore propose a fully convolutional architecture (**fully convolutional**).

### Fully Convolutional Model

Fully convolutional models have been used successfully for spatial analysis of arbitrary size inputs (Wolf and Platt 1994; Long, Shelhamer, and Darrell 2014). We construct the fully convolutional model by converting the fully connected layers of the VGG F network to convolutional layers. This allows the network to efficiently "slide" across a larger input image and make multiple evaluations of different parts of the image, incorporating all available contextual information.

Given an unrolled $h \times w \times d$-dimensional input $x \in \mathbb{R}^{hwd}$,

fully connected layers perform a matrix-vector product

$$\hat{x} = f(Wx + b)$$

where $W \in \mathbb{R}^{k \times hwd}$ is a weight matrix, $b$ is a bias term, $f$ is a nonlinearity function, and $\hat{x} \in \mathbb{R}^k$ is the output. In the fully connected layer, we take $k$ inner products with the unrolled $x$ vector. Thus, given a differently sized input, it is unclear how to evaluate the dot products.

We replace a fully connected layer by a convolutional layer with $k$ convolutional filters of size $h \times w$, the same size as the input. The filter weights are shared across all channels, which means that the convolutional layer actually uses fewer parameters than the fully connected layer. Since the filter size is matched with the input size, we can take an element-wise product and add, which is equivalent to an inner product. This results in a scalar output for each filter, creating an output $\hat{x} \in \mathbb{R}^{1 \times 1 \times k}$. Further fully connected layers are converted to convolutional layers with filter size $1 \times 1$, matching the new input $\hat{x} \in \mathbb{R}^{1 \times 1 \times k}$. Fully connected layers are usually the last layers of the network, while all previous layers are typically convolutional. After converting fully connected layers to convolutional layers, the entire network becomes convolutional, allowing the outputs of each layer to be reused as the convolution slides the network over a larger input. Instead of a scalar output, the new output is a 2-dimensional map of filter activations.

In our fully convolutional model, the $400 \times 400$ input produces an output of size $2 \times 2 \times 4096$, which represents the scores of four (overlapping) quadrants of the image for 4096 features. The regional scores are then averaged to obtain a 4096-dimensional feature vector that becomes the final input to the classifier predicting nighttime light intensity.

### Training and Performance Evaluation

Both CNN models are trained using minibatched gradient descent with momentum. Random mirroring is used for data augmentation, along with 50% dropout on convolutional layers replacing fully connected layers. The learning rate begins at 1e-6, a hundredth of the ending learning rate of the VGG model. All other hyperparameters are the same as in the VGG model as described in (Chatfield et al. 2014). The VGG model parameters are obtained from the Caffe Model Zoo, and all networks are trained with Caffe (Jia et al. 2014). The fully convolutional model is fine-tuned from the pre-trained parameters of the VGG F model, but it randomly initializes the convolutional layers that replace fully connected layers.

In the process of cropping, the random cropping model throws away over 68% of the input image when predicting the class scores, losing much of the spatial context. The random cropping model achieved a validation accuracy of 70.04% after 400,200 SGD iterations. In comparison, the fully convolutional model achieved 71.58% validation accuracy after only 223,500 iterations. Both models were trained in roughly three days. Despite reinitializing the final convolutional layers from scratch, the fully convolutional model exhibits faster learning and better performance. The final fully convolutional model achieves a validation accuracy of 71.71%, trained over 345,000 iterations.
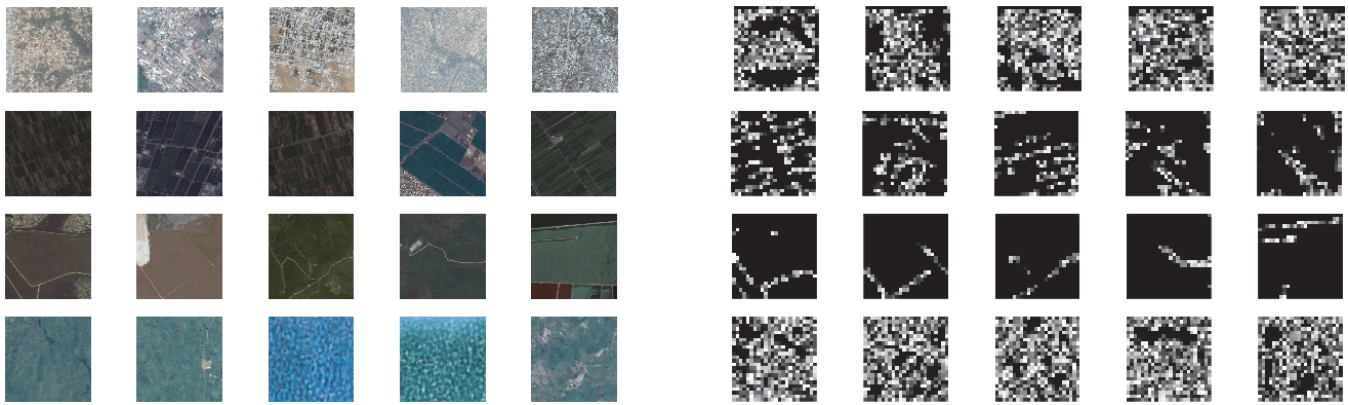
Figure 2: *Left:* Each row shows five maximally activating images for a different filter in the fifth convolutional layer of the CNN trained on the nighttime light intensity prediction problem. The first filter (first row) activates for urban areas. The second filter activates for farmland and grid-like patterns. The third filter activates for roads. The fourth filter activates for water, plains, and forests, terrains contributing similarly to nighttime light intensity. The only supervision used is nighttime light intensity, i.e., no labeled examples of roads or farmlands are provided. *Right:* Filter activations for the corresponding images on the left. Filters mostly activate on the relevant portions of the image. For example, in the third row, the strongest activations coincide with the road segments. Best seen in color. See the companion technical report for more visualizations (Xie et al. 2015). Images from Google Static Maps.

## Visualizing the Extracted Features

Nighttime lights are used as a data-rich proxy, so absolute performance on this task is not directly relevant for poverty mapping. The goal is to learn high-level features that are indicative of economic development and can be used for poverty mapping in the spirit of transfer learning.

We visualize the filters learned by the fully convolutional network by inspecting the 25 maximally activating images for each filter (Figure 2, left and the companion technical report for more visualizations (Xie et al. 2015)). Activation levels for filters in the middle of the network are obtained by passing the images forward through the filter, applying the ReLU nonlinearity, and then averaging the map of activation values. We find that many filters learn to identify semantically meaningful features such as urban areas, water, roads, barren land, forests, and farmland. Amazingly, **these features are learned without direct supervision**, in contrast to previous efforts to extract features from aerial imagery, which have relied heavily on large amounts of expert-labeled data, e.g., labeled examples of roads (Mnih and Hinton 2010; 2012). To confirm the semantics of the filters, we visualize their activations for the same set of images (Figure 2, right). These maps confirm our interpretation by identifying the image parts that are most responsible for activating the filter. For example, the filter in the third row mostly activates on road segments. These features are extremely useful socioeconomic indicators and suggest that transfer learning to the poverty task is possible.

## Poverty Estimation and Mapping

The first target task we consider is to predict whether the majority of households are above or below the poverty threshold for 643 groups of households in Uganda.

Given the limited amount of training data, we do not attempt to learn new feature representations for the target task. Instead, we directly use the feature representation learned by the CNN on the nighttime lights task ($\mathcal{P}_2$). Specifically, we evaluate the CNN model on new input images and feed the feature vector produced in the last layer as input to a logistic regression classifier, which is trained on the poverty task (**transfer model**). Approximately 100 images in a 10km $\times$ 10km area around the average household location of each group are used as input. We compare against the performance of a classifier with features from the VGG F model trained on ImageNet only (**ImageNet model**), i.e., without transfer learning from nighttime lights. In both the ImageNet model and the transfer model, the feature vectors are averaged over the input images for each group.

The Uganda LSMS survey also includes household-specific data. We extract the features that could feasibly be detected with remote sensing techniques, including roof material, number of rooms, house type, distances to various infrastructure points, urban or rural classification, annual temperature, and annual precipitation. These survey features are then averaged over each household group. The performance of the classifier trained with survey features (**survey model**) represents the gold standard for remote sensing techniques. We also compare with a classifier trained using the nighttime light intensities themselves as features (**lights model**). The nighttime light features consist of the average light intensity, summary statistics, and histogram-based features for each area. Finally, we compare with a classifier trained using a concatenation of ImageNet features and nighttime light features (**ImageNet + lights model**), an explicit way of combining information from both source problems.

All models are trained using a logistic regression classifier with L1 regularization using a nested 10-fold cross valida-
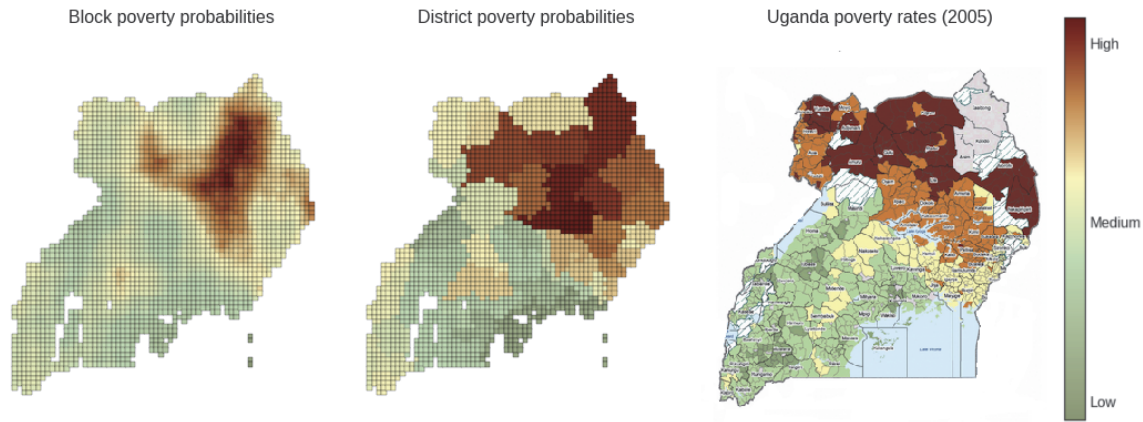
Figure 3: *Left:* Predicted poverty probabilities at a fine-grained 10km × 10km block level. *Middle:* Predicted poverty probabilities aggregated at the district-level. *Right:* 2005 survey results for comparison (World Resources Institute 2009).

|  | Survey | ImgNet | Lights | ImgNet +Lights | Transfer |
|---|---|---|---|---|---|
| Accuracy | 0.754 | 0.686 | 0.526 | 0.683 | **0.716** |
| F1 Score | 0.552 | 0.398 | 0.448 | 0.400 | **0.489** |
| Precision | 0.450 | 0.340 | 0.298 | 0.338 | **0.394** |
| Recall | 0.722 | 0.492 | 0.914 | 0.506 | 0.658 |
| AUC | 0.776 | 0.690 | 0.719 | 0.700 | **0.761** |

Table 1: Cross validation test performance for predicting aggregate-level poverty measures. Survey is trained on survey data collected in the field. All other models are based on satellite imagery. Our transfer learning approach outperforms all non-survey classifiers significantly in every measure except recall, and approaches the survey model.

tion (CV) scheme, where the inner CV is used to tune a new regularization parameter for each outer CV iteration. The regularization parameter is found by a two-stage approach: a coarse linearly spaced search is followed by a finer linearly spaced search around the best value found in the coarse search. The tuned regularization parameter is then validated on the test set of the outer CV loop, which remained unseen as the parameter was tuned. All performance metrics are averaged over the outer 10 folds and reported in Table 1.

Our transfer model significantly outperforms every model except the survey model in every measure except recall. Notably, the transfer model outperforms all combinations of features from the source problems, implying that transfer learning was successful in learning novel and useful features. Remarkably, our transfer model based on remotely sensed data approaches the performance of the survey model based on data expensively collected in the field. As a sanity check, we find that using simple traditional computer vision features such as HOG and color histograms only achieves slightly better performance than random guessing. This further affirms that the transfer learning features are nontrivial and contain information more complex than just edges and colors.

To understand the high recall of the lights model, we analyze the conditional probability of predicting "poverty" given that the average light intensity is zero: The lights model predicts "poverty" almost 100% of the time, though only 51% of groups with zero average intensity are actually below the poverty line. Furthermore, only 6% of groups with nonzero average light intensity are below the poverty line, explaining the high recall of the lights model. In contrast, the transfer model predicts "poverty" in 52% of groups where the average nighttime light intensity is 0, more accurately reflecting the actual probability. The transfer model features (visualized in Figure 2) clearly contain additional, meaningful information beyond what nighttime lights can provide. The fact that the transfer model outperforms the lights model indicates that transfer learning has succeeded.

## Mapping Poverty Distribution

Using our transfer model, we can scalably and inexpensively construct fine-grained poverty maps at the country or even continent level. We evaluate this capability by estimating a country-level poverty map for Uganda. We download over 370,000 satellite images covering Uganda and estimate poverty probabilities at 1km × 1km resolution with the transfer model. Areas where the model assigns a low probability of being impoverished are colored green, while areas assigned a high risk of poverty are colored red. A 10km × 10km resolution map is shown in Figure 3 (left), smoothed at a 0.5 degree radius for easy identification of dominant spatial patterns. Notably, poverty reduction in northern Uganda is lagging (Ministry of Finance 2014). Figure 3 (middle) shows poverty estimates aggregated at the district level. As a validity check, we qualitatively compare this map against the most recent map of poverty rates available (Figure 3, right), which is based on 2005 survey data (World Resources Institute 2009). This data is now a decade old, but it loosely corroborates the major patterns in our predicted distribution. Whereas current maps are coarse and outdated, our method offers much finer temporal and spatial resolution and an inexpensive way to evaluate poverty at a global scale.

## Conclusion

We introduce a new transfer learning approach for analyzing satellite imagery that leverages recent deep learning advances and multiple data-rich proxy tasks to learn high-level feature representations of satellite images. This knowledge is then transferred to data-poor tasks of interest in the spirit of transfer learning. We demonstrate an application of this idea in the context of poverty mapping and introduce a fully convolutional CNN model that, without explicit supervision, learns to identify complex features such as roads, urban areas, and various terrains. Using these features, we are able to approach the performance of data collected in the field for poverty estimation. Remarkably, our approach outperforms models based directly on the data-rich proxies used in our transfer learning pipeline. Our approach can easily be generalized to other remote sensing tasks and has great potential to help solve global sustainability challenges.

## Acknowledgements

## References

Abelson, B.; Varshney, K.; and Sun, J. 2014. Targeting direct cash transfers to the extremely poor. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1563–1572. ACM.

Bouvrie, J. 2006. Notes on convolutional neural networks.

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

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

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

Independent Expert Advisory Group Secretariat. 2014. A world that counts: Mobilising the data revolution for sustainable development. Technical report.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R. B.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. *CoRR* abs/1408.5093.

Le, Q. V.; Ranzato, M.; Monga, R.; Devin, M.; Chen, K.; Corrado, G. S.; Dean, J.; and Ng, A. Y. 2012. Building high-level features using large scale unsupervised learning. In *International Conference on Machine Learning*.

Long, J.; Shelhamer, E.; and Darrell, T. 2014. Fully convolutional networks for semantic segmentation. *CoRR* abs/1411.4038.

Ministry of Finance. 2014. Poverty status report 2014: Structural change and poverty reduction in Uganda.

Mnih, V., and Hinton, G. E. 2010. Learning to detect roads in high-resolution aerial images. In *Computer Vision–ECCV 2010*. Springer. 210–223.

Mnih, V., and Hinton, G. E. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 567–574.

Murthy, K.; Shearn, M.; Smiley, B. D.; Chau, A. H.; Levine, J.; and Robinson, D. 2014. Skysat-1: very high-resolution imagery from a small satellite. In *SPIE Remote Sensing*, 92411E–92411E. International Society for Optics and Photonics.

NOAA National Geophysical Data Center. 2014. F18 2013 nighttime lights composite.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, 1717–1724. Washington, DC, USA: IEEE Computer Society.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22(10):1345–1359.

Razavian, A. S.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR* abs/1403.6382.

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

Uganda Bureau of Statistics. 2012. Uganda national panel survey 2011/2012.

United Nations. 2015. The millennium development goals report 2015.

Varshney, K. R.; Chen, G. H.; Abelson, B.; Nowocin, K.; Sakhrani, V.; Xu, L.; and Spatocco, B. L. 2015. Targeting villages for rural development using satellite image analysis. *Big Data* 3(1):41–53.

Wolf, R., and Platt, J. C. 1994. Postal address block location using a convolutional locator network. In *Advances in Neural Information Processing Systems*, 745–752. Morgan Kaufmann Publishers.

World Resources Institute. 2009. Mapping a better future: How spatial analysis can benefit wetlands and reduce poverty in Uganda.

Xie, M.; Jean, N.; Burke, M.; Lobell, D.; and Ermon, S. 2015. Transfer learning from deep features for remote sensing and poverty mapping. *CoRR* abs/1510.00098.

Zeiler, M. D., and Fergus, R. 2013. Visualizing and understanding convolutional networks. *CoRR* abs/1311.2901.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using Places database. In *Advances in Neural Information Processing Systems*, 487–495.