Multi-Task Deep Learning for Predicting Poverty from Satellite Images

Shailesh M. Pandey, Tushar Agarwal, Narayanan C Krishnan
Department of Computer Science and Engineering
Indian Institute of Technology Ropar, India
{shailesh.pandey, tushar.agarwal, ckn}@iitrpr.ac.in

Abstract

Estimating economic and developmental parameters such as poverty levels of a region from satellite imagery is a challenging problem that has many applications. We propose a two step approach to predict poverty in a rural region from satellite imagery. First, we engineer a multi-task fully convolutional deep network for simultaneously predicting the material of roof, source of lighting and source of drinking water from satellite images. Second, we use the predicted developmental statistics to estimate poverty. Using full-size satellite imagery as input, and without pre-trained weights, our models are able to learn meaningful features including roads, water bodies and farm lands, and achieve a performance that is close to the optimum. In addition to speeding up the training process, the multi-task fully convolutional model is able to discern task specific and independent feature representations.

Introduction

Developing countries spend a significant amount of resources in planning and implementing policies and schemes for poverty alleviation. The primary sources of data, if and when used, for devising these schemes are ground level surveys, such as the decennial census, of socio-economic parameters. However, collecting extensive statistics is a significant exercise in manual effort and monetary resources resulting in infrequent sampling. Therefore, timely and accurate data are often not available at the time of formulating policies. This may lead to ineffective implementation and, at times, even wasteful expenditure. A timely, inexpensive and accurate source of data that is readily available should help in addressing some of these issues.

Satellite imagery is one such cost effective data-source that provides a wealth of information for learning developmental conditions of a region. The ever-increasing resolution of satellite imagery and relatively easy access to it in the public domain make it a potential resource. Figure 1 presents the satellite imagery for rural areas from different parts of India. In the first column, the image on the top shows a region having houses with concrete roofs (Figure 1 A1), whereas the image on the bottom shows a region with thatch-roofed houses (Figure 1 A2). The second column illustrates contrasting images of two regions classified as having 100% electricity (Figure 1 B1) and 0% electricity (Figure 1 B2) for lighting. The last column shows regions with tap water (Figure 1 C1) and a river or a canal (Figure 1 C2) as a major source of drinking water. There are distinct visual features that can be associated with the presence or lack of economic development. For instance, roads and streets are visible in satellite imagery and correlate with the level of economic development. Further, soil color, roof material, water bodies, farmland etc. are also visible and may provide useful information about the level of development in a region. The primary objective of this study is to automatically learn visual features in satellite images indicative of development and poverty and build models that can predict poverty in regions of India reliably.

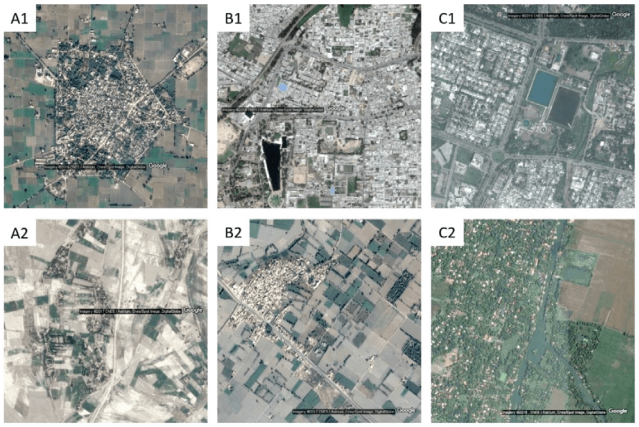

Figure 1: Regions with (A1) concrete roofs and (A2) thatch roofs, (B1) 100% electricity and (B2) 0% electricity for lighting, and (C1) 85.9% households with tap water and (C2) 99.1% with river/canal as drinking water source. Distinct visual features in satellite imagery can be associated with the presence or lack of economic development.

Our main contribution is a two step approach for poverty prediction. First, we engineer a multi-task fully convolutional model to predict the material of roof, source of lighting and source of drinking water from the satellite imagery of a village. Unlike income or poverty data, the values for these parameters are available at the village level in the In-

dian census of 2011 and hence provide a larger dataset for training. Moreover, these parameters correlate with visual features in satellite imagery, making them a good choice for our study. Second, we train a model to predict the income levels (a direct indicator of poverty) using the predicted developmental parameter outputs of the first model. The multi-task fully convolutional model enables us to study the relationships between the features learned for the prediction of different developmental parameters. The proposed architecture is flexible so as to learn shared and independent representations for the different tasks, and at the same time reduces the total computation time of training and prediction in comparison to single task models. The results presented in this paper clearly support the effectiveness of this approach. In addition, our experiments suggest that predicting poverty levels from multiple developmental parameters is more reliable than using a single parameter, an approach that is found in existing literature (Xie et al. 2015; Abelson, Varshney, and Sun 2014).

The source code used for our study, supplementary material and full-size versions of satellite images in this paper are available at following GitHub repository
https://github.com/agarwalt/satimage.

Related Work

Aerial and satellite imagery has been extensively studied in the context of image segmentation, labeling and object identification (Bruzzone and Demir 2014; Huang et al. 2015; Iovan, Boldo, and Cord 2008). Deep learning, applied effectively in diverse computer vision problems, has been successfully utilized for segmentation and classification tasks on satellite imagery such as division of terrain into classes like ground, water, vegetation and buildings (Längkvist et al. 2016; Paisitkriangkrai et al. 2016) and feature selection (Zou et al. 2015).

Recently, an interesting direction of investigation using satellite images has emerged, where satellite images are used to predict statistics such as poverty or income (Xie et al. 2015; Abelson, Varshney, and Sun 2014). These studies suggest the feasibility of predicting poverty related parameters from satellite imagery using nighttime light intensity (Xie et al. 2015) or roof type (Abelson, Varshney, and Sun 2014) as a "proxy". Xie et al. build on a convolutional neural network trained on the ImageNet dataset by adding additional layers to the model, and training the augmented model to predict the nighttime light intensity in a region using its daytime satellite imagery. The features learned by this model are then used in a new model to predict poverty levels. This transfer of knowledge overcomes the shortage of training data for directly predicting poverty levels from satellite imagery. However, nighttime light intensity values do not show significant variation over rural villages, and, as observed by Xie et al., have magnitudes close to zero for a large fraction of such villages. Developmental statistics collected on the ground are more detailed, show greater variation and more accurately represent the socio-economic situation of a region. Using more than one developmental statistic also makes our models more robust. Abelson et al. use the fraction of thatched roofs in a satellite image to estimate the poverty in a region. Template matching is utilized to detect roofs in 400×400 images. The model to predict the percentage of thatch-roofed households in an image is trained on crowdsourced labeled images. In contrast, our approach does not require any manual annotation of images. Nischal et al. (Nischal et al. 2015) correlate nighttime light intensity calculated from a single image of India with census data at the state level only. On the other hand, we estimate statistics at a significantly finer level of villages and sub-districts[1].

Dataset

The 2011 Census of India, data from which we utilize in this study, includes statistics about number of households, type of roof, source of lighting and drinking water, possession of assets, and more for all rural regions in India. In this study, we choose statistics related to the major source of drinking water, major source of lighting and the type of roof of households as indicators of economic progress of the most populous state of India, Uttar Pradesh. This state comprises $109,980$ villages and wards. Income statistics for rural regions at the sub-district level[2] are drawn from the publicly available Socio-Economic Caste Census of 2011.

We query the Google Geocoding API to obtain coordinates of the center of a village as well as the box-bounding latitudes and longitudes(geocodes) from its address in the census data. We then utilize the Google Static Maps API to extract images for the villages from the determined geocodes. We select a sufficiently high zoom level, maximizing the coverage of villages and the level of detail given the image-size constraints. The 1920×1920 sized images, at zoom level 16, fully cover 67.46% $(66,135)$ villages. Each image spans a ground surface area of approximately $19\ \mathrm{km}^2$. To the best of our knowledge, this is the first study to report deep learning experiments on images of size orders of magnitude larger than that of images in previous work (e.g. 400×400 in the study by Xie et al.). In order to remove images with imperceptible or without any visual features indicative of human settlement, we filter this dataset to $47,120$ villages by including only villages with at least 100 households. We use this dataset to train, tune and test our models.

Predicting Developmental Statistics

We divide our prediction task into two parts and train separate models for each part. Our first task consists of training a multi-task model to predict the material of roof, source of lighting and source of drinking water in a region. For our second task, we create a model to predict the household income level in a region using the material of roof, source of lighting and source of drinking water in the region as inputs.

[1] A sub-district is a set of villages.

[2] Each sub-district in Uttar Pradesh comprises, on average, 212 villages.

Multi-task Learning

Multi-task learning involves learning multiple tasks simultaneously while exploiting the similarities and differences among the tasks. A multi-task model can enable the learning of a better input representation for a particular task than a single task model since it can potentially take advantage of information from other related tasks. Constraining the input representation to be shared across tasks can also be seen as a form of regularization and can lead to features which produce lower generalization errors for the multiple tasks (Caruana 1998). This technique enables the transfer of knowledge among the tasks and in effect, increases the training data for each task. In this study, we use a multi-task model to predict (1) the roof type, (2) source of lighting and (3) the source of drinking water for rural villages.

Formally, let $(X^t, Y^t)_{t=1}^T$ be a set of T tasks. X^t are the training examples for task t and Y^t are the targets that have to be learned for the task. In the specific form of multi-task learning we employ in this study, all tasks share the same training examples, i.e., $X^1 = X^2 \ldots = X^T = X$. However, each task has a different target. We propose a multi-task fully convolutional deep learning model with the initial 3 convolutional layers shared across the tasks followed by $T = 3$ task-specific branches. Each task specific branch, in turn, has 8 convolutional layers. The output layer of each task specific branch produces a tuple of values, which can be compared to the true target. The cross-entropy loss function is applied on the task specific outputs and the errors propagated backwards into the task-specific branches. This architecture is illustrated in Figure 2A. Instead of creating task specific branches, one could potentially learn a model with a single layer that outputs the targets of all the tasks together. However, this will increase the number of parameters to be learned at the output layer and further assumes that all the outputs are related to each other in some manner. In addition, the multi-task model reduces the total computation time of training and prediction in comparison to three separately trained equivalent single task models (Figure 2B).

The input to the multi-task network is a 1920×1920 image of a region. We do not perform any enhancement operation (such as contrast-stretching) on the image before it is fed to the network. The output of the network is a tuple of tuples $O_i = (o_i^1, o_i^2, o_i^3)$, one each for the three tasks – roof-type, water-source and lighting-source – for each village i in the training dataset. Each sub-tuple o_i^t comprises values summing to one, each value representing a category and indicating the fraction of households in a village belonging to that category. For instance, the sub-tuple $(0.75, 0, 0, 0, 0, 0, 0, 0.25, 0)$ for the task roof type represents a region with 75% households with roofs made of grass/thatch/bamboo/wood/mud, and 25% households with concrete roofs. The sub-tuples for source of lighting and source of drinking water are similarly defined. The multi-task model, in summary, outputs 24 values (9 for roof type, 6 for source of lighting and 9 for source of drinking water) as three probability distributions, one distribution per task. The details about the categories for each task can be found on the project website.

We train the multi-task model for rural villages in Uttar Pradesh with at least 100 households. From the $47,120$ villages so obtained, we construct training, test and validation datasets by taking approximately 80%, 10% and 10% of the total number of villages. The division is approximate because instead of dividing villages randomly, we divide sub-districts. Hence, each sub-district lies entirely in either the training, test or validation set. We use the multi-task model's outputs for our poverty prediction task, and poverty/income statistics are available at the subdistrict level. By choosing a consistent division strategy for data for both our tasks, we are able to avoid any bias in the evaluation of the poverty-prediction model by using the data points in the test set for the multi-task model for evaluating the poverty-prediction model as well. Hence, no village from the training set for the multi-task model occurs in the validation set for the poverty-prediction model.

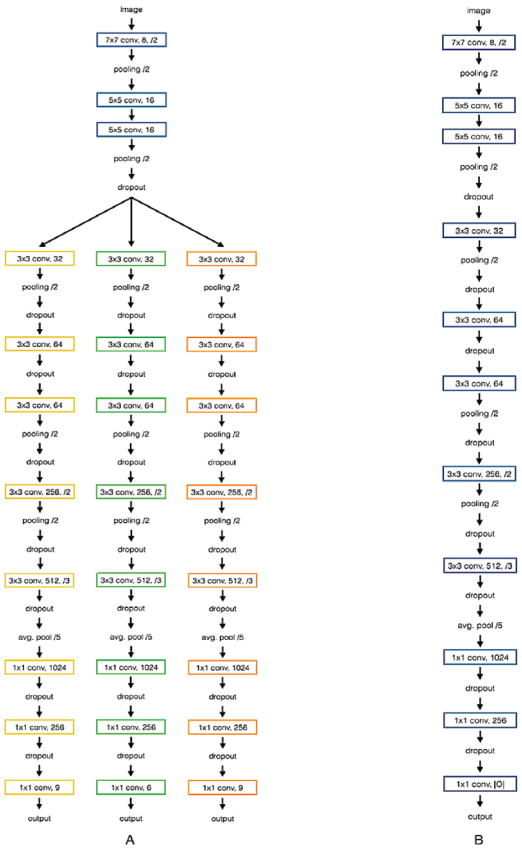

Figure 2: The architecture of (A) the multi-task model and (B) the single task models.

Model Architecture and Training

It is observed across a spectrum of computer vision tasks that lower layers of deep convolutional networks learn task-independent features such as edges, whereas features learned in layers close to the output layer are task-specific. The multi-task model enables the learning and use of common features together for reduction in training time by a factor of 3 over the total time for training three separate single-

task models, while allowing to distinguish between features learned in the task-specific branches of the model. Since the model is fully-convolutional, it requires fewer parameters than an equivalent fully-connected model and results in faster training. The shared part of the model consists of 3 convolutional "blocks". A convolutional block comprises a convolutional layer, a batch normalization layer, a ReLU activation layer, an optional pooling layer (window size 2×2 and stride of 2×2) and a dropout layer, in that order. For each convolutional layer, we set an L2 weight decay of 0.001 and a maxnorm constraint of 4 (Srivastava et al. 2014). The task specific branches include 8 convolutional layers which successively reduce the output size to equal the number of classes for a particular task (9 for roof type, 6 for source of lighting and 9 for source of drinking water). The dropout is set to 0.2 for the deepest shared block, 0.2 for the first five task-specific convolutional layers and 0.3 for the last three layers.

We use gradient descent with mini-batch and the Adadelta optimizer (Zeiler 2012) for training. The model is trained for $125,156$ steps (192 hours on an NVIDIA TITAN X GPU). The model is presented with 20 images at every step. The minimum validation loss occurs at the $95,268$th step and further training leads to overfitting. During forward propagation, the outputs at the branching points are replicated and passed on to each task specific branch. During backpropagation, errors from the task specific branches are averaged at the last shared layer before further propagation backwards. The average validation loss across the three tasks is used for early stopping. Additionally, we indirectly evaluate the quality of predictions of the multi-task model by utilizing the second model to predict poverty.

Visualizing the Learned Representations

We analyze the filter responses of the multi-task fully convolutional model to understand the representations learned by the model. In the multi-task model's first block, filters learn edges with different orientations. Figures 3 (1) and 3 (2) show differently oriented edges for different filters for a particular region. This is consistent with observations reported in the literature for computer vision tasks and thus provides additional validation for our training procedure. In the second and third blocks, more complex albeit generic features including roads, settlements and farmland are highlighted in the filter activations (Figures 3 (3) and 3 (4)). Interestingly, the "Google" watermark is not highlighted in the filter activations in the shared part of the multi-task model. In summary, task-independent features are learned in the shared part of the model.

On the other hand, we expect the filter activations for the task-specific branches to highlight objects of relevance to the respective tasks. Figure 4 illustrates the filter responses for the second convolutional layer in the task specific branches for each of the three tasks[3]. The filter responses for these layers are smaller than the input image by a factor of 16. For the branch corresponding to roof type (Figures 4 A1 and B1), only human settlements are highlighted. For instance, roads and canals are completely hidden in the filter responses. For the lighting source prediction task branch (Figures 4 A2 and B2), in addition to settlements, roads are highlighted prominently. This observation points to a correlation between presence of roads in a village and the source of lighting. It is important to note here that the color of an object in the satellite image is not as important as the kind of the object itself. Roads and settlements have different colors, and roads are not highlighted in the filter activations for roof type prediction. For the branch corresponding to the drinking water source prediction task (Figures 4 A3 and B3), settlements are still highlighted, although not as prominently as in the other two branches. Farmland – perhaps related to presence of tube-wells or hand-pumps – and roads – possibly related to presence of tap water – are visible. More importantly, the canal in top right corner in Figure 4 B is not visible in the activations for the first two branches but can be seen to leave an impression in the third branch's filter activation. In addition, for all three tasks, the clouds present in Figure 4 A have been completely ignored.

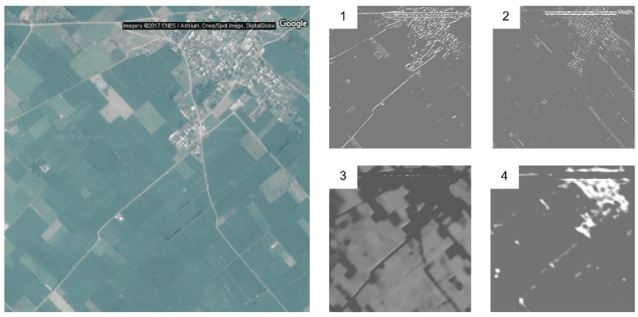

Figure 3: Filter activations for the multi-task model's shared layers for a region. Filters for activations in (1) and (2) are present in the first convolutional layer, and clearly show that edges of two different orientations have been learned. Filters for activations in (3) and (4) are present in the second and third convolutional layers respectively, and segment the image into its constituents such as human settlements and farmlands.

The model learns the correlation between specific visual features and developmental parameters without any external guidance such as annotation of specific objects in the training images. Further, the sharp filter responses (once the filter activations are reduced to their true size) indicate that the model has trained sufficiently well and is well regularized (Srivastava et al. 2014).

Predicting Poverty

For our second task, we create a simple four-layer fully-connected model which takes as inputs the outputs of the multi-task fully convolutional model and generates as output a distribution over three monthly household income levels: (1) income below ₹ 5,000[4], (2) income between ₹ 5,000

[3] Additional illustrations of filter responses have been documented on the project GitHub repository.

[4] ₹ is the symbol of the Indian Rupee. US $1 ≈ ₹ 64 (on 2017-09-09).

Figure 4: Filter activations for the multi-task model's task-specific layers. Images at the top show two different areas of the same region. Bottom images show filter activations for (1) roof type, (2) source of lighting and (3) source of drinking water. The filter activations for the task-specific branches highlight objects of relevance to the respective tasks.

- ₹ 10,000 and (3) income above ₹ 10,000. The output distribution represents the fraction of households in a sub-district with a particular level of income. Since income statistics are available at a much higher level (sub-district) than developmental statistics such as the material of roof, source of lighting and source of drinking water (available for each village in a sub-district), we choose to indirectly predict income and poverty through developmental statistics. Developmental statistics are also more directly visible in a satellite image than income statistics. Moreover, it is not feasible to download a reasonably detailed image of an entire sub-district (that could, on average, span 780km^2 for Uttar Pradesh) to train a model to directly predict income levels.

Since income statistics are available at the sub-district level, we need to aggregate the predictions generated by the multi-task fully convolutional model for all villages in each sub-district. We do this by calculating the average distribution of roof type, source of lighting and source of drinking water for all villages belonging to the same sub-district, weighted by the number of households in each village. We divide the total 312 sub-districts in the dataset into a training set (80%), validation set (10%) and test set (10%). These sets are the same as those used for the multi-task model.

Model Architecture and Training

The income estimation model is a four-layer fully-connected network that contains hidden layers with 8, 4 and 4 nodes and ReLU activation. The output layer contains 3 nodes, one for each of the three income levels defined in the income dataset. Softmax activation is used at the final layer. Further, each activation is preceded by a batch-normalization (Ioffe and Szegedy 2015) layer. Additionally, the input is channeled through another batch-normalization layer for standardization over batches before feeding into the model. The RMSProp optimizer is used for gradient descent with a batch size of 50. The model is trained for $1,000$ epochs with cross-entropy loss over the validation dataset as the early-stopping criterion. The model's hyperparameters are tuned based on performance on the validation set.

For comparison, we train a separate model on the Census of 2011 data for our chosen developmental statistics. Since this model is trained on statistics collected through a ground survey, this model represents the optimum for the poverty-prediction task. Hence, two poverty-prediction models are trained: (1) A model trained on values of developmental statistics from the official Census of 2011 data (model C.D., on census data) and (2) a model trained on the predictions of the multi-task model for the developmental statistics (model P.D., on predicted data) for the same regions as in model C.D.

Results

To compare model P.D. and the optimum model, model C.D., we calculate the correlation between the ground-truth values and predicted values for the three income categories (Figure 5). Predictions of model P.D. are observed to be positively correlated with the ground truth values. Also, model P.D. consistently performs close to the optimum model, model C.D., across all three income levels.

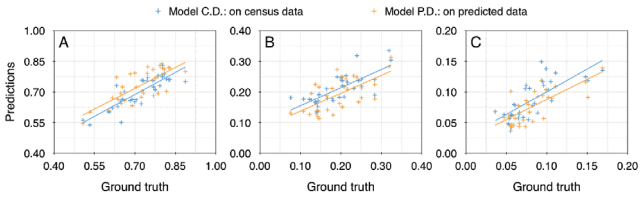

Figure 5: Correlation between the ground-truth values and predicted values of fraction of households with income (A) below ₹ 5,000, (B) from ₹ 5,000 to ₹ 10,000 and (C) more than ₹ 10,000. Predictions of model P.D. are positively correlated with the ground truth values. Also, model P.D. consistently performs close to the optimum model, model C.D., across all three income levels.

We also find the accuracy, precision and recall by setting a threshold on the fraction of households in a sub-district belonging to the lowest income category, income below ₹ 5,000 (Table 1). Let the threshold be $0 \leq t \leq 1$. Let $0 \leq p \leq 1$ be the fraction of households in a sub-district having income less than ₹ 5,000. If $p \geq t$, we classify the sub-district as "poor", and "not poor" otherwise. From survey data and models C.D. and P.D., we have the fraction of households in a sub-district with income below ₹ 5,000. We apply the threshold t to generate binary class labels ("poor" and "not poor") from survey data and outputs of models C.D. and P.D.. Accuracy, precision and recall (Table 1) for the models C.D. and P.D. are, therefore, calculated using the ground truth labels generated from survey data. We observe that model P.D. performs close to the optimum model, model C.D., and significantly better than the baseline (majority class prediction).

Table 1: A comparison of the performance of the poverty-prediction model trained on values of developmental statistics from the official Census of 2011 data (model C.D.), the poverty-prediction model trained on predictions of the multi-task model for the developmental statistics (model P.D.) and the baseline model (predict majority class).

Threshold	Model C.D.: on census data			Model P.D.: on predicted data			Baseline
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
$0.1 - 0.5$	**1.0**	1.0	1.0	**1.0**	1.0	1.0	**1.0**
0.6	**0.969**	1.0	0.967	**0.969**	0.967	1.0	0.937
0.7	**0.875**	0.895	0.895	0.75	0.789	0.789	0.594
0.8	0.781	1.0	0.125	**0.875**	0.7	0.875	0.75
$0.9, 1.0$	**1.0**	0.0	0.0	**1.0**	0.0	0.0	**1.0**

Using only satellite imagery as input, we are able to estimate income and, in turn, poverty, close to the true values collected on the ground by significant manual effort and monetary expense. In addition, model P.D.'s performance helps indirectly evaluate the multi-task fully convolutional model since better (worse) predictions of developmental statistics will improve (degrade) the performance of model P.D.. Additional experiments[5] show that the models utilizing all three developmental statistics (roof type, source of lighting and source of drinking water) perform better than models utilizing only one of the statistics. Therefore, using multiple developmental parameters improves the robustness and generalization performance of our models.

Summary

We propose a two-step approach for predicting poverty in rural regions of India from satellite imagery. First, we train a multi-task fully convolutional model to predict three developmental parameters – the main material of the roof, source of lighting and source of drinking water – from satellite imagery. We observe that meaningful features, such as roads, settlements, farm lands and water bodies are automatically learned by the multi-task fully convolutional model. Second, we train a model to predict the income levels (a direct indicator of poverty) using the predicted developmental parameter outputs of the first model. Using only satellite imagery as input, we are able to estimate income and poverty close to the true values collected on the ground by significant manual effort and monetary expense.

Acknowledgements

The authors are grateful to NVIDIA Corporation for donating the TITAN X GPUs used for this research.

References

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

Bruzzone, L., and Demir, B. 2014. A review of modern approaches to classification of remote sensing data. In *Land Use and Land Cover Mapping in Europe*. Springer. 127–143.

Caruana, R. 1998. Multitask learning. In *Learning to learn*. Springer. 95–133.

Google Maps Geocoding API, https://developers.google.com/maps/documentation/geocoding/, Accessed: 2017-09-09.

Google Static Maps API, https://developers.google.com/maps/documentation/static-maps/, Accessed: 2017-09-09.

Huang, X.; Xie, C.; Fang, X.; and Zhang, L. 2015. Combining pixel-and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8(5):2097–2110.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Iovan, C.; Boldo, D.; and Cord, M. 2008. Detection, characterization, and modeling vegetation in urban areas from high-resolution aerial imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 1(3):206–213.

Längkvist, M.; Kiselev, A.; Alirezaie, M.; and Loutfi, A. 2016. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing* 8(4):329.

Nischal, K.; Radhakrishnan, R.; Mehta, S.; and Chandani, S. 2015. Correlating night-time satellite images with poverty and other census data of india and estimating future trends. In *Second ACM IKDD Conference on Data Sciences*, 75–79.

Paisitkriangkrai, S.; Sherrah, J.; Janney, P.; and van den Hengel, A. 2016. Semantic labeling of aerial and satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9(7):2868–2881.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

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

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zou, Q.; Ni, L.; Zhang, T.; and Wang, Q. 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters* 12(11):2321–2325.

[5]Refer to the GitHub repository