Understanding Development Indicators from Satellite Imagery

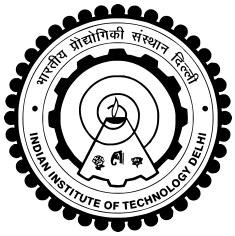
Potnuru Kishen Suraj Armaan Singh Bhullar

ee1130480@iitd.ac.in ee1130439@iitd.ac.in

2013EE10480 2013EE10439

Supervisor

Prof. Subhashis Banerjee



Department of Electrical Engineering

Indian Institute of Technology Delhi

Acknowledgments

We would like to express our thanks and gratitude for Prof. Subhashis Banerjee for providing his time and valuable guidance for the project. We thank Prof. Sourabh Paul for the insightful discussions we had especially about the economic aspects of the project. We would also like to thank Pitney Bowes for providing the GIS census data. Authors also thank the IIT Delhi HPC facility for computational resources.

Potnuru Kishen Suraj

Armaan Singh Bhullar

Contents

List of Figures

List of Tables

1 Introduction

Figure 1: Contrasting visual patterns in rich and poor regions. If Humans can do it, so can computer!

The primary aim of the project, when started was to develop a quick and reliable method to predict socio-economic indicators of a region. This is important because the current ground based surveying methods are costly and time-consuming. For instance, the Indian Census happens every 10 years and the most recent one cost about US $330 million[1].

The aim is to have real time estimates of various parameters like poverty, access to drinking water, income etc. For this task, satellite imagery is one of the important candidate as a source of primary data for a region. It is

[1]Census 2011 Wikipedia [2]

cheap, widely available in real time and accessible. Indian census data is a very good dataset for India containing diverse socio-economic indicators.

However, in the course of our project, especially as we explored the economic aspects, we also discovered some other interesting use cases for our project. These involve the use of the approach we have developed to gain insights into economic and social questions. One such application has been demonstrated in section 6.

2 Data Description

2.1 Overview

This problem of 'Predicting Development indicators from Satellite images' has several possible kinds of data sets. Here by a data set we mean a collection of key-value pairs i.e, a satellite image and a corresponding economic indicator to that image. For example consider a $1km^2$ image of a village, corresponding economic indicator could be like information about assets(kinds of houses, roads etc.). Key observation here is that this economic indicator should be Geo-referenced, otherwise it's not possible to map the satellite image to it's corresponding economic indicator. As this problem is being solved for Indian context, it makes it even more challenging and interesting since there is no geo-referenced economic data readily available for India in public domain for common usage. So,creating a proper dataset is one of the important challenges in the project. This data forms the backbone of the project and according to our knowledge this is the first such attempt to combine satellite imagery with geo-referenced economic data in Indian context. This problem was previously studied in the context of African nations in [11]. A combination of several datasets was used to create the required dataset to solve this problem. A key ingredient of the final dataset is the satellite imagery data. Possible economic indicators data includes census data, DHS(Demographic and Health Survey), nightlight imagery and LSMS(Living Standard Measurement Survey) .

2.2 Data Details

2.2.1 Indian Census Data

Recent Indian census was conducted in 2011 which is 15^{th} Indian Census[10]. This is one of our primary data-sets for the project. We processed the raw census data to construct economic indicators. We have used two census data-sets for this purpose :

- **House Listing** This data-set contains information about asset ownership and living conditions. This includes the sources of energy (electric/solar/oil etc.), sanitation (type of water source), assets (mobile phones, vehicles etc.). This data(at house level) is aggregated to create village level data by simple averaging techniques. Every row corresponds to a village or a ward(in cities) , columns correspond to facilities and the corresponding entries are the percentage of houses in the village with that particular facility. One important point here is that it does not contain any data pertaining to education and health facilities like literacy rates, hospitals etc.

  The data is available for individual states of India [1]. The level granularity of the data(at village level) makes it very useful. We aggregated this data-set to construct a set of 15 indicators described in the ((table).

- **District Census Handbook (DCHB)** DCHB contains information about village level infrastructure and other details. We have primarily used this as a source for education and health data. Important thing

to note here is that we could not find a suitable education data set at village level. So we used the education infrastructure as a proxy for education levels. The same approach was followed for constructing health indicator

Here, under infrastructure it contains the number of various types of educational institutes(primary/secondary/govt./private) and medical establishments(dispensary/hospitals etc.) Note that this data set is fundamentally different from Houselisting in that it is not directly derived from village residents. So while predicting this, we expected a deviation from the performance we achieved for the houselisting derived indicators.

Main drawback is that, this census data is not geo-tagged, so it can not be readily related to the satellite imagery. Basically, one can not get satellite images corresponding to the village entry in the census data. For this purpose we used 3^{rd} party data from Pitney Bowes.

2.2.2 Pitney Bowes Data

Survey of India has created village boundary maps for all census villages in India. This data is kindly granted to us Pitney Bowes. It has GIS maps for every village in India census 2011 data (see Figure 2). Basically this dataset contains boundaries of Indian villages which helps us to get the satellite images for these villages.

Presently we are granted this data for seven Indian states, under a NDA(Non Disclosure Agreement) .

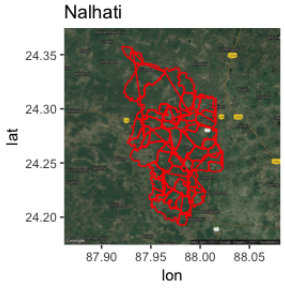


Figure 2: Boundaries of villages in Nalhati Tehsil of West Bengal, a sample from Pitney Bowes Data

2.2.3 Satellite Imagery

The two possible sources for satellite images are Google static maps and Bhuvan(Geo-platform of ISRO) . Only images from google static maps are used for this project. Google static maps are created by imagery that is collected over time and is mosaicked. For example, noise like presence of clouds is removed by processing the data overtime. Due to the ease of use and availability, google static maps are used. Appendix F details how images are downloaded. [2].

[2]Data is downloaded within standard usage limits, which allows a maximum of 25,000 images per day and an account is necessary

2.2.4 Night time satellite imagery

National Oceanic and Atmospheric Administration(NOAA) in collaboration with National Geophysical Data Centre(NGDC) collects night light data generated due to human habitation. Satellites under DMSP(Defence Meteorological Satellite Program) are equipped with Operational Linescan Systems(OLS) .These sensors have been used to collect the night light data. The products are 30 arc second grids, spanning -180 to 180 degrees longitude and -65 to 75 degrees latitude[13]. Each 30 arch second grid cell is mapped to discrete values from {0,1,2,..63}, where 63 corresponds to the highest light intensity.

We explored the use of night time light intensity as economic proxy for India (refer to [4] for details). In [4] Chen says that night light are not of significant value for developing countries.

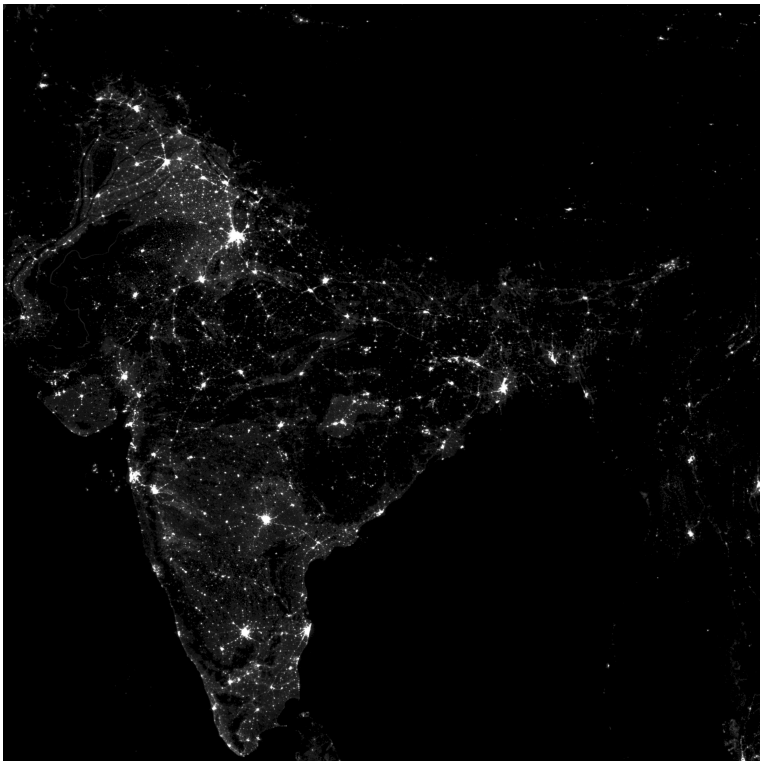


Figure 3: Nighttime image of India generated from NOAA data.

2.2.5 Global Administrative Areas

GADM is a spatial database of the locations of the world administrative areas(their boundaries) developed by Robert Hijmans at University of California [7]. Indian Boundaries at four levels(country,state,district and tehsil) are maintained by them,so the corresponding data looks like Figure4. This basically is a digital map of India,with boundaries at tehsil(sub-district) level being the lowest possible granularity. As this data does not have enough granularity ,it is not alone fit for generation of datasets for caffe model. But,this

can be used to limit our search space in the vast night time TIFF file.

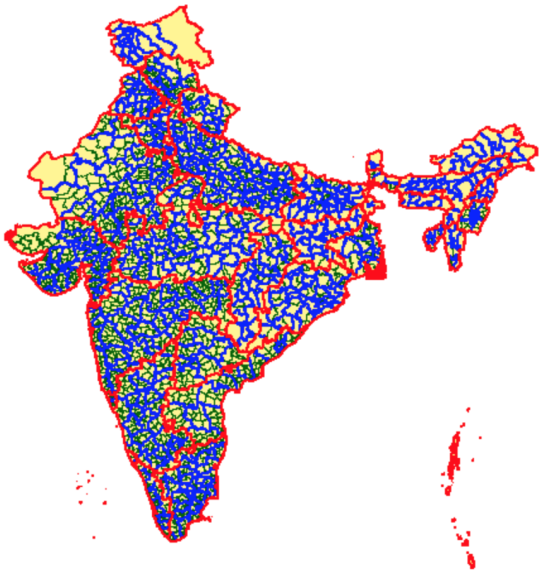


Figure 4: Visualization of Indian boundaries at tehsil level

2.3 Data set Creation

2.3.1 Economic and Social indicators

In this part we will describe the creation of Economic and Social indicators from the two census datasets described above (2.2.1). Table 1 shows the 15 constructed indicators and their meanings. For details please refer to the technical section.

Table 1: Description of Asset based indicators constructed from Houselisting dataset 2.2.1

Indicator	Description(% of houses in the village with the facility)
electronics	radio/transistor/tv/laptop
water-treated	tap-water from treated source/covered well/ tube-well
water-untreated	tap-water from untreated source/covered well
water-natural	drinking water from ponds/rivers/lakes
electric-like	electricity from grid/solar
oil-like	energy source as kerosene/other oil
has-phone	having land-line/mobile/both
transport-cycle	owning cycle
transport-motorized	owning motorcycle/scooter/car/jeep
no-assets	having no assets (cycle/phone etc.)
banking-services-availability	availability of banking services
cook-fuel-processed	cooking fuel as LPG/electric stove etc.
bathroom-within	bathroom within premises
rooms-under-3	number of rooms in dwelling less than 3
household-size-under-5	less than 5 family members

Table 2: Description of social/health/education based indicators constructed from DCHB dataset (2.2.1)

Indicator	Description
healthfacilities	normalized number of health facilities)
edufacilities	normalized number of education facilities
stpercent	percentage of ST population in village
scpercent	percentage of SC population in village

3 Deep CNN models for regression

Whole idea of the idea of the project is to know how much satellite images mean to multiple economic indicators and build models to predict economic indicators. Before attempting this big problem in total,we tried solving a simpler problem of predicting nightlight values from daytime satellite images. It encouraged us to move ahead to solve the problem for multiple census indicators.

3.1 Regression with Caffe

Caffe has a native support for classification problems, but for solving other problems one needs to define one's own data layers making it less direct. For supporting convolutional neural network architectures, caffe has several in-built layers like convolutional, fully connected and so on. There are no in-built data layers in caffe which support loading datasets having multiple

float target values. So we had to develop our data layers and as an added advantage we could do on the fly augmentation. It is very carefully developed to support prefetching of data and latency of this step is ensured to be less than forward-backward time of the CNN net. A version of this layer uses multi-threading to load the image data.

Previously models in model zoo were mostly used for solving classification problems. There is no generic simple model for regression and we don't know the typical hyper parameter setting to be used and the kind initialization which makes the problem much more challenging. We used VGG_CNN_S [3] by VGG team from BMVC 2014. It was trained on Imagenet challenge, and top performer among their other models. Our idea behind using this model is owing to the simplicity of the model. Hyper parameter setting in these was for classification paradigm. We had to experiment on the several hyperparameter settings to decide the right one.

The general learning policy used is learning rate 1e-6 (VGG model used 1e-3) weight decay 5e-3 (VGG model used 5e-4) momentum 0.8 (VGG model used 0.9)

The layer wise settings for these parameters is different in different models.

To monitor the performance we developed our own r2loss score layer.This was very helpful monitoring multi-output regression deep CNN models.

4 Nightlight training

In the first phase of the project, we used the data rich nightlight domain for training a preliminary model and then fine-tuned this model to predict census data, following the approach of [11].
The first step in this approach is to train a model to predict nightlights of a geographical region from it's daytime satellite image.

4.1 Introduction

Predicting Night light values is a much simpler problem due to the following reasons:

- Every cell in the Night light data spans a fixed latitude and longitude.
- For every cell there is a night light value from 0-63.

A data point in our data set is a satellite image taken at of the center of night light cell and target is the night light value (2.2.4).
Training on census data makes it difficult due to the fact the village boundaries are irregular and of varying size. We used VGG_CNN_S model ([3]) by VGG team from BMVC 2014. It was trained on ImageNet challenge, and was a top performer among their other models. We used the same layer configuration as the above network. We used the pre-trained model to initialize the weighs in our network.

4.2 Night light data creation

As we are solving this problem for Indian context, dataset is created in such a way that all the points are within India. Each 30arch second grid cell spans a latitude and a longitude of 0.833° in NOAA Night light data. One way of creating the dataset is to use a third party for a source latitude-longitude pairs of relevant cities(direct mapping). The major problem with this way of creating the dataset is that, the query lat-long will lie at a random place within a nighttime cell. Consequently, a satellite image collected at this point will span multiple nightlight cells on ground and thus would be distorted from the true image. The second problem with such a direct mapping is that NOAA data is extremely skewed. Most of the cells have a nighttime value of 0.

So, to overcome the first problem, the dataset is created in a reverse way, like an inverse mapping. The main problem with doing inverse mapping is that one has to iterate over the whole night time image, which is huge(10^9 points). Therefore to limit our search space we used from India boundary map from 2.2.5.

After collecting the data for the whole of India, we observe that this dataset is skewed towards low-intensity values (5). So we carefully selected the samples to reduce the skewness in the data(undersampling the lower night time values), resulting in a final dataset of size 2,19,000 datapoints.

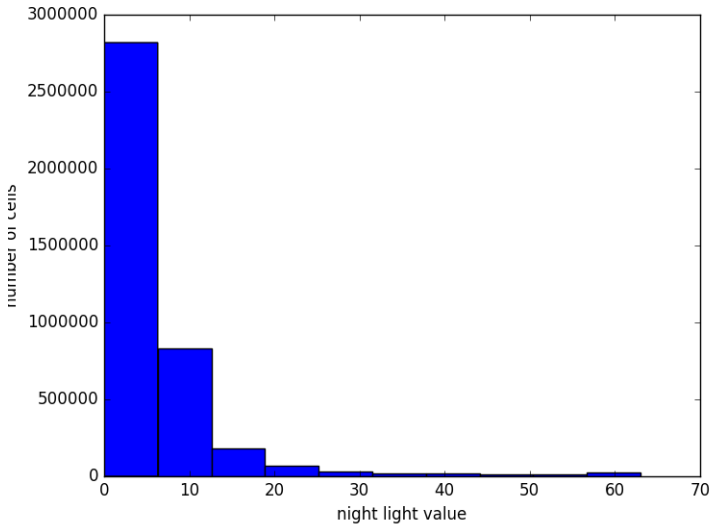


Figure 5: Distribution of Indian nightlight data.

It improved the standard deviation(of original, skewed data) from 8.50 to 19.65. This also decreased the skew(third order moment) from 3.62 to 0.4. Figure 6

Then for all the cells which are selected, lat-long of the center is calculated and used to download the images. The images are downloaded from google static maps API(F) at two different settings, zoom = 16 and image size 400x400 with area of $1\ km^2$ (area varies with latitude) zoom = 16 and image size 640x640 with area of $2km^2$

The training set has 190000 points and test set 20000 data points.

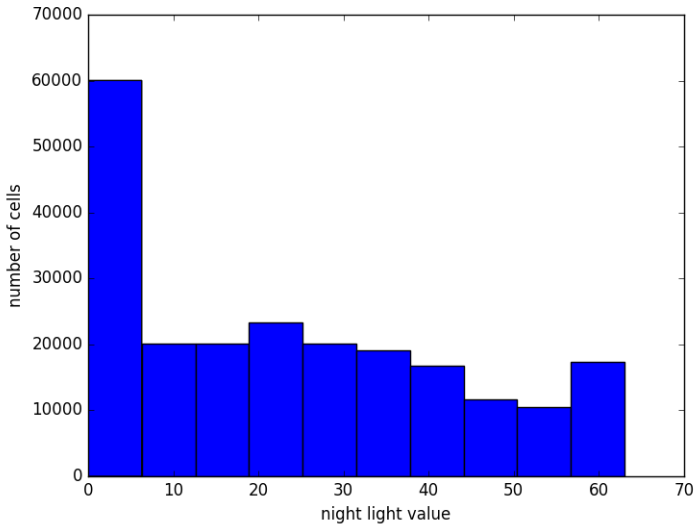


Figure 6: Final distribution of collected data.

4.3 Experiments

Input image size of the VGG_CNN_S [3] model is modified from $224x224$ to $400x400$. The last fully connected layers are trained with more regularization(weight decay) and a higher learning rate (this follows from Jeff Yosinski's paper on transfer learning, [19]).

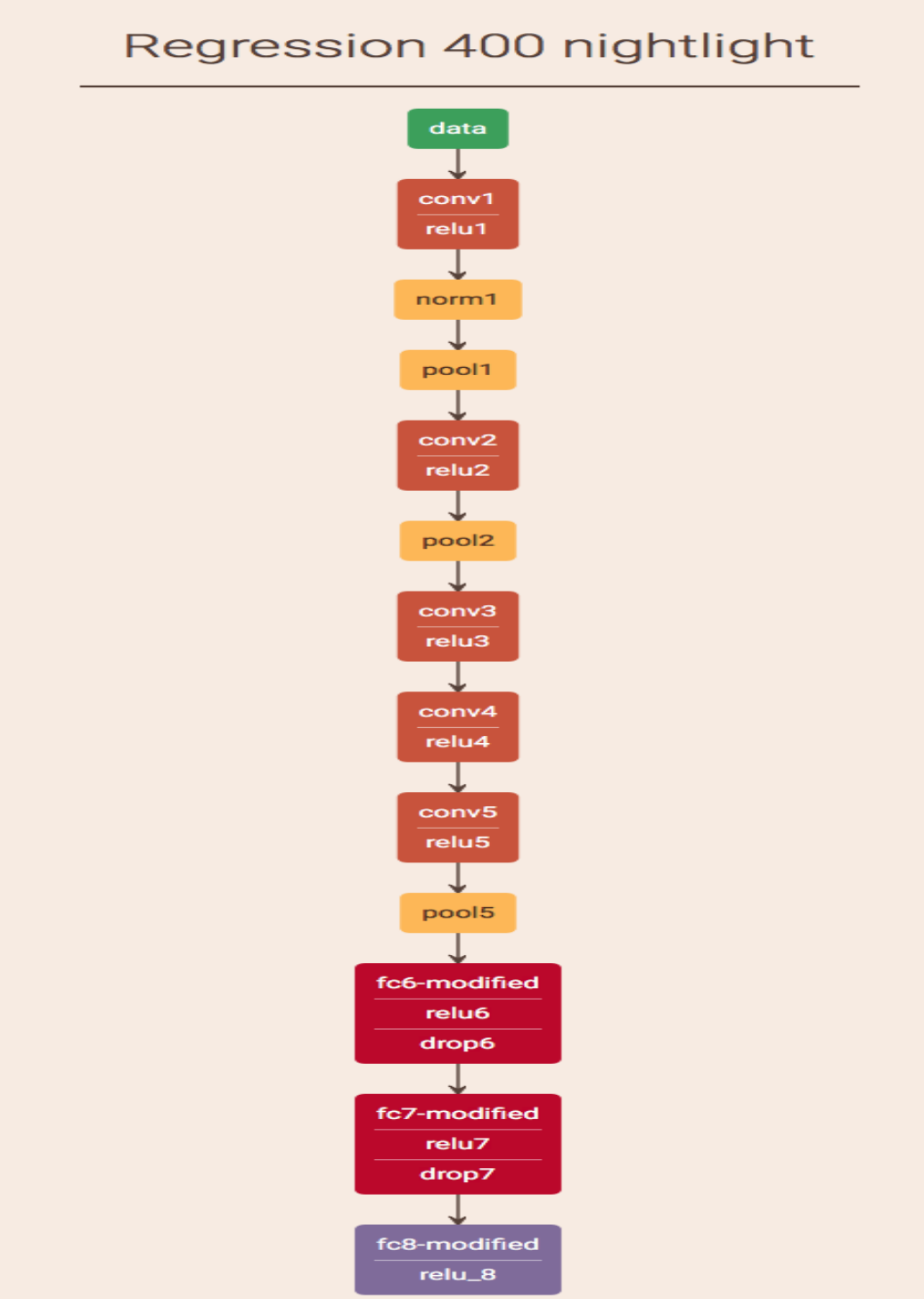


Figure 7: Night light model architecture

Evaluation metric used is R2 score which is also known as coefficient of determination. This can be intuitively understood as the amount of variance that the model can explain. Then we ran three different major experiments, using the same architecture.

1. As every cell spans a lat-long of 0.9 km^2 ,images collected at zoom=16 and size 400x400 will completely subsume night time cell. The regression model trained so resulted in a r2score of 0.66.

2. As discussed in 3.1, our custom data layer implementation helped in on the fly augmentation. Different kinds of augmentation techniques used are horizontal flip,vertical flip and image rotations. This further improved our r2score to 0.69.

3. Interestingly when we used day time satellite images at zoom level=16 and size 640x640 with a ground area of 2 km^2 as our input to the model, model seems to perform extremely well with a r2score 0.79. This dramatic improvement in the performance can be explained by the observation that night light at a particular cell is influenced by the surroundings cells, because light has a spread and that context is important for predicting the value at the current cell.

4.3.1 Conclusion

Finally the model has performed remarkably well, capturing 79% of the variance. This is a first of a kind result with no previous benchmarks. The performance is attributed to the proper creation of dataset, removing the

S.No	Models	R2score
1	$1km^2$ ground area	0.66
2	$1km^2$ ground area+augmentation	0.69
3	$2km^2$ ground area+augmentation	0.79

Table 3: A comparison of models trained with different configurations

inherent skew, making other right design choices. Also augmentation techniques increased the training sample while capturing rotational and translational invariance.

4.4 Transfer Learning

One important implication of training a model on night light data is that it could help in predicting other economic indicators by transfer learning. Transfer learning is based on the assumption that features learnt in night light predictions are general enough that could help predict all of the different indicators in a related task. Here in our experimentation we did not find observable gains doing transfer learning,infact it performed worse than direct training. The MSE of the model using transfer learning is 3208.49 compared to direct training with a MSE of 3092.4 using similar settings. This can be simply understood by the fact that number of villages we have are quite enough(2.13 lakh) to train a entire deep CNN. So,the idea of transfer from night lights model is not being pursued further.

Table 4: Performance of various models on census indicators

Indicator	tiling	9layer+9km^2	9layer+4km^2	8layer+9km^2+widefc
Mean Squared Error	6567.2	6184.8	7046	5819.4
rooms-under-3	0.655	0.419	0.408	0.456
household-size-under-5	0.518	0.509	0.518	0.541
water-treated	0.295	0.271	0.168	0.303
water-untreated	0.500	0.312	0.285	0.357
water-natural	0.435	0.286	0.211	0.332
electric-like	0.381	0.526	0.421	0.564
oil-like	0.381	0.532	0.426	0.571
electronics	0.406	0.327	0.257	0.347
has-phone	0.411	0.452	0.420	0.491
transport-cycle	0.435	0.346	0.347	0.400
transport-motorized	0.541	0.550	0.460	0.578
no-assets	0.381	0.341	0.345	0.380
banking-services-availability	0.421	0.371	0.383	0.406
cook-fuel-processed	0.401	0.520	0.463	0.545
bathroom-within	0.389	0.458	0.373	0.480

5 Regression into village census indicators

5.1 Introduction

In this section we are trying to predict village census indicators from day time satellite images of those villages. As our dataset had 213000 villages we could afford to train an entire CNN. Training the model for each of the 19 indicators individually would be a very cumbersome task. Doing Multi output regression into all of the indicators would be easier and fruitful. To solve this problem,a relevant dataset needs satellite images of villages and their corresponding census indicators. As Pitney Bowes Inc.(2.2.2) has Geo-referenced village boundaries along with census ids. Then corresponding village census indicators are tagged to them with the help of census ids. We show series models carefully improving the accuracy.

5.2 Initial Approach: Tiling

Census Indicators from House Listing data (2.2.1) is a qualitative indicator because it is averaged across all the households in the village.

5.2.1 Dataset Creation

The biggest problem to predict village census indicators is owing to the fact that all villages are of different shapes and sizes. So for this experiment we tried to do a tiling approach, where we tiled all the villages in bihar into $1km^2$ blocks,with the help of Pitney Bowes data. This resulted in 155000

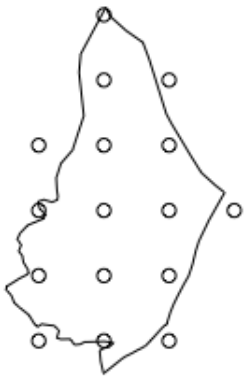


Figure 8: The generated centers of different tiles for a village

tiles for the whole of bihar. So there is an inherent assumption that every tile represents the village qualitatively. These tiles will form the input images to the model and corresponding census indicators as targets. Then test dataset had 20000 villages and rest are in train dataset.

5.2.2 Results

To do this regression we used euclidean loss as objection function with layer twise L2 regularization. So for every tile we tried to predict targets. This model is referred to as 'tiling' in the table(4). This model achieved a Mean Squared Error of 6490.

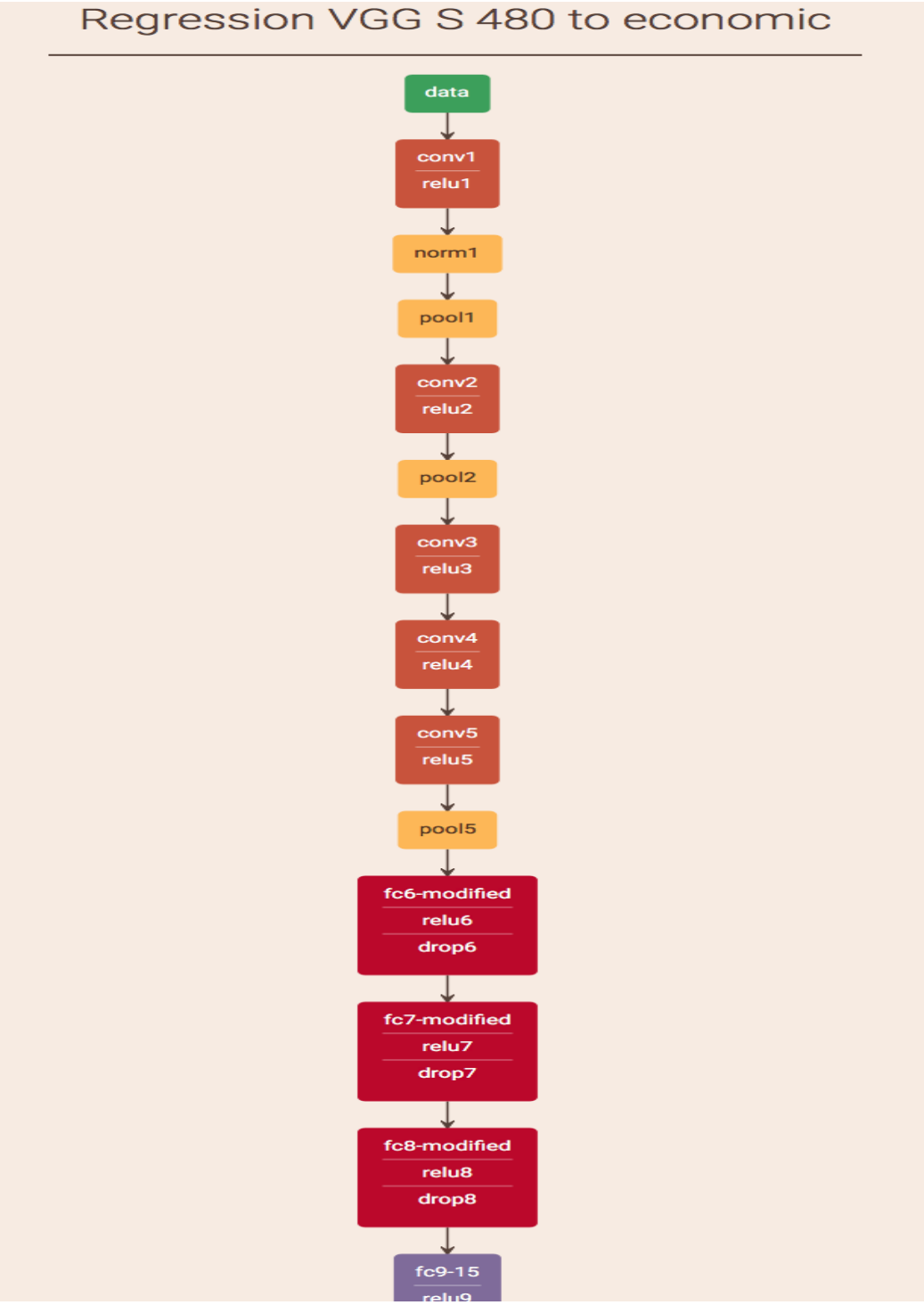


Figure 9: 9 layer model taking an input of 480x480 size of a village and regressing into 15 census indicators

5.3 Regression to Census Indicators

Here in this subsection, every village is represented using a single satellite image instead of tiled representation.

5.3.1 9 Layer model, 9 sqkm images

Instead of tiling the villages we thought of representing the villages in a simpler way, by choosing a $9km^2$ block at the centroid of the village. Now this single $9km^2$ image is used as input to the model and the corresponding the indicators as targets. An additional fully connected layer is added to the model, resulting in Mean Squared Error of 3092.4.

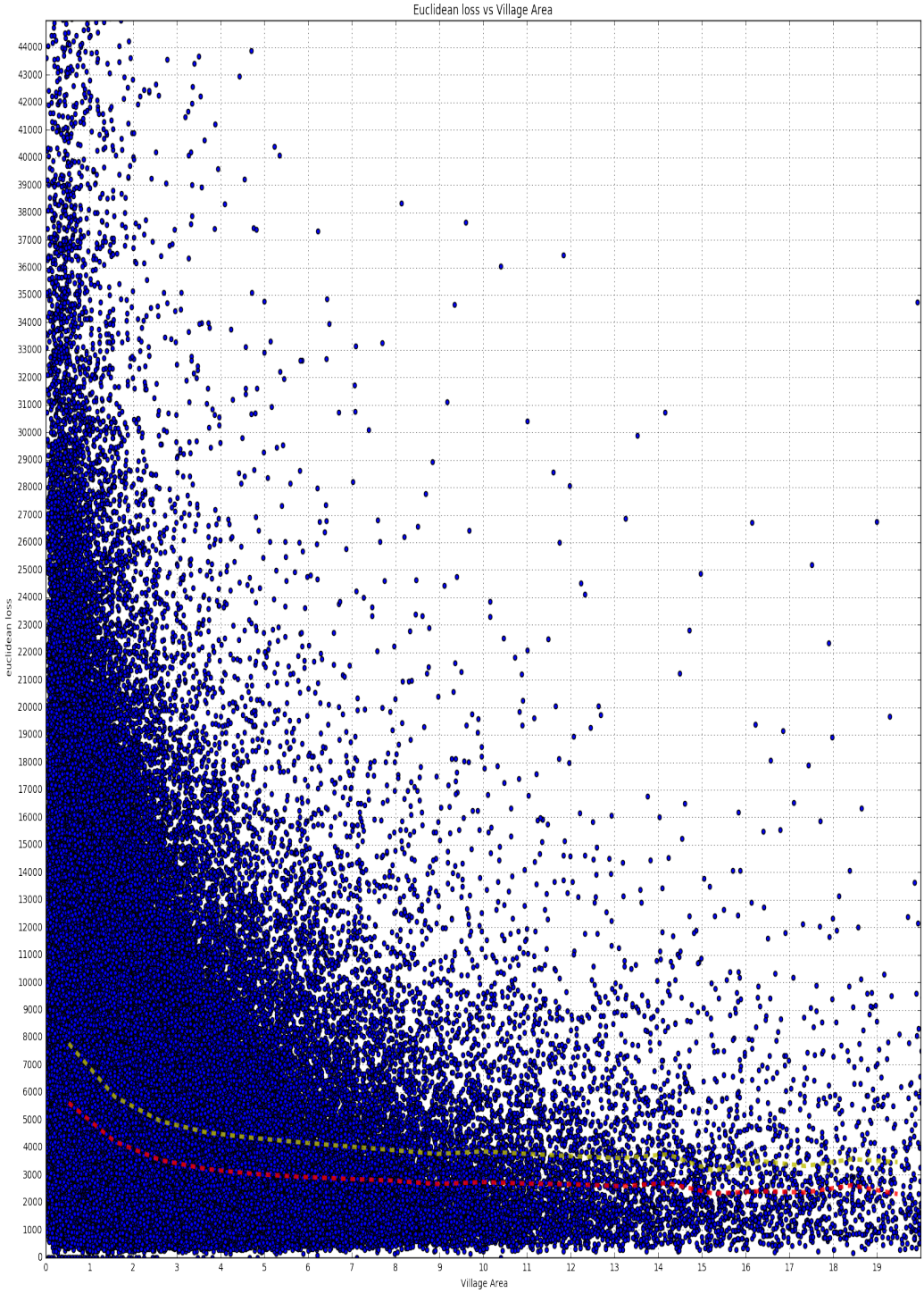


Figure 10: MSE versus village area. Yellow line in the plot is the bin(1 sqkm bin) wise mean and red line is the bin wise median.

5.3.2 Analysis

While analysing the model one the most interesting thing found is that when tried to plot the euclidean loss of the village(cost function) of model vs area (10).We observed that the villages with smaller area had much higher loss than the villages with high losses.

5.4 9layer model,4 sqkm villages

To overcome this problem we trained a model only with villages having ground area less than $2km^2$.This selection left us with 134000 villages. Every village is represented using a satellite image taken at the centroid of the boundary having $4km^2$ ground area. Observe that we have also reduced the ground area from $9km^2$ to $4km^2$ also in effect,expecting that this will reduce the influence of surrounding villages. Keep in mind that the boundaries from Pitney Bowes erratic and have 0.5km shift on average. So we couldn't use satellite images with $2km^2$ ground area. From the previous model the MSE only for these village is 7660 but with current approach it improved to 7060. This is no better than '9layer+$9km^2$' model. But the overall error didn't show any significant improvement. So it is a problem with the villages rather than with the previous design choice. This referred to as '9layer+$4km^2$' model in the table(4).

5.5 Final model

Finally ,after experimenting we selected a model with a totally different architecture. As choosing satellite images with lower ground area didn't improve our accuracy, we decided to represent every village with $9km^2$ satellite images. So, satellite images with $9km^2$ ground area are selected as input to the model.The convolutional layers are kept the same as the VGG_CNN_S([3]) This model had totally different architecture. Dropout ratio for fully connected layers reduced to 0.1. Width of fc7 layer is increased from 4096 to 9192. This is referred to as final model in table 4 The MSE is 5818.4 lowest of all the models.

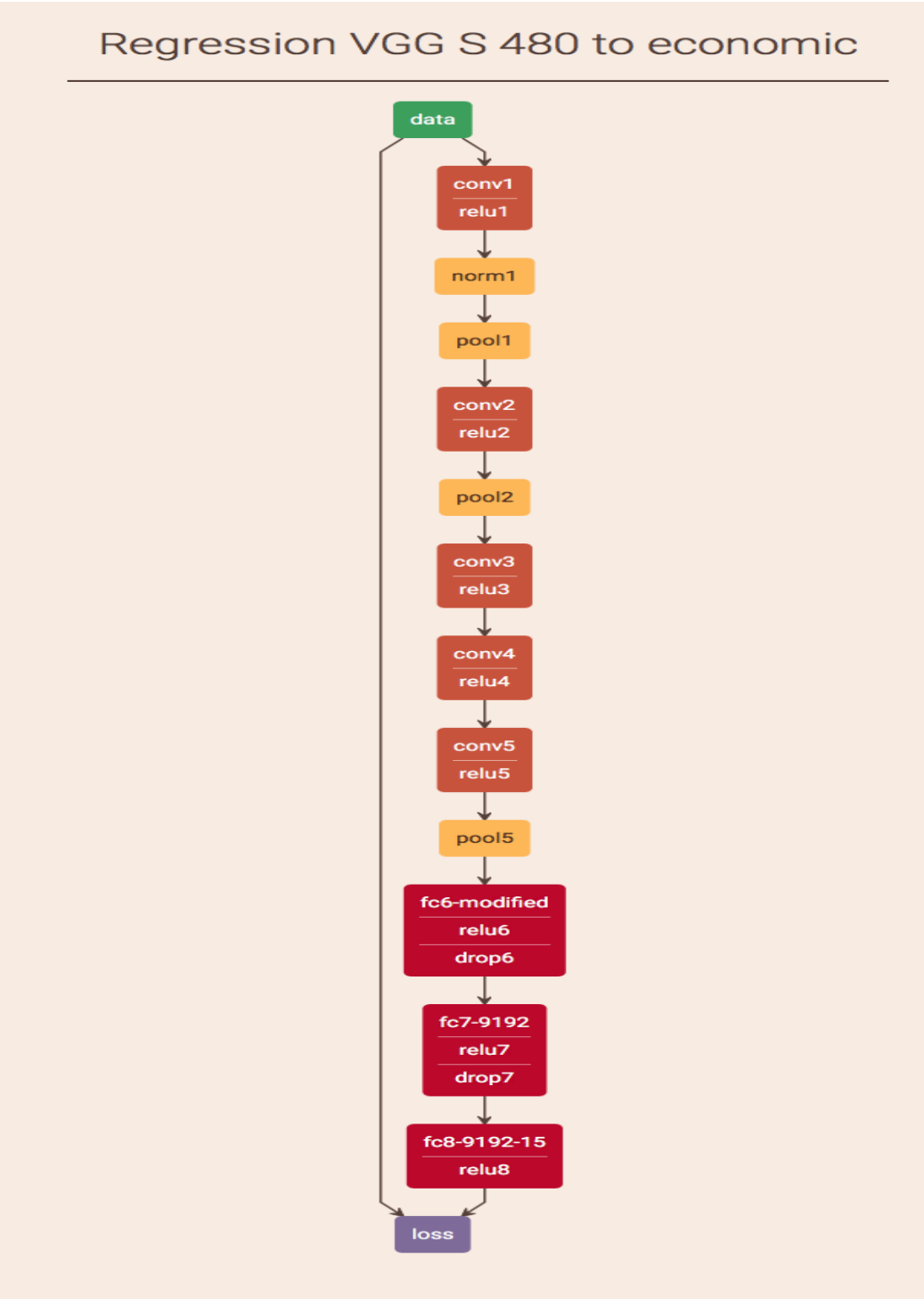


Figure 11: final model ,8 layer model

S.No	Indian state	MSE
1	Bihar	4618.91
2	Haryana	5094.47
3	Jharkhand	7864.03
4	Punjab	4433.09
5	Uttar Pradesh	6020.15
6	West Bengal	6929.89

Table 5: Model performance state wise showing the Mean squared error statewise

5.6 Statewise analysis

The state wise analysis shows the performance of model statewise(5). Loss is highest in Jharkhand.

5.7 Model performance on social indicators

We trained a separate model for predicting health, education and demographic (SC/ST percent) indicators . This is due to the fact that DCHB data is at village level and very different in nature to house listing data fundamentally(2). The results obtained are shown in table 6 .

5.7.1 Understanding the results

- The high performance on ST percent indicator implies strong geographical basis for ST populations. This is indeed expected because tribal

S.No	census indicators	R2score(test)
1	sc percent	0.187
2	st percent	0.659
3	health facilities	0.132
3	educational facilities	0.061

Table 6: Model performance on health/education and demographic (SC/ST percent) indicators. The model performs poorly on these except ST percent. The reasons are discussed below.

regions are generally correlated with proximity to forested area.

- Standard deviation for sc percent:22.43, st percent:23.87, health facilities:3.65,educational facilities: 2.92.

- Low R2 scores for health facilities and educational facilities is due to the fact that the standard deviation for them is very low. So modelling them as a classification problem for these indicators would make more sense.

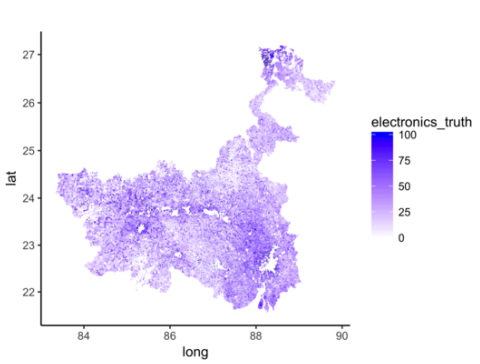
6 Applications

6.1 Observing geographical trends in economic indicators

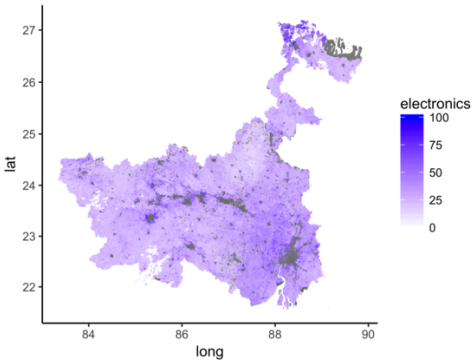
Census data, though statistically reliable, contains noise due to inaccurate collection(especially for remote regions). For instance, on analyzing the census(6.3), we see that a lot of villages have extreme variations (many of their indicators are 0 or 100).

Figure 12a shows the plot of electronics indicator obtained from Census for the states of West Bengal(WB) and Jharkhand. In comparison, 12b plots the predicted values of electronics for these states. As we can see, predicted values are more continuous and bring out the geographical trends.

Figure 12a shows the plot of electronics indicator obtained from Census for the states of West Bengal(WB) and Jharkhand. In comparison, 12b plots the predicted values of electronics for these states. As we can see, predicted values are more continuous and bring out the geographical trends.

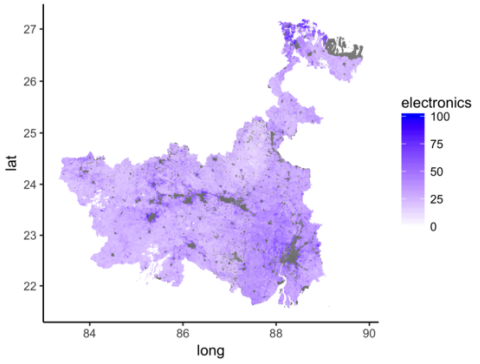


(a) Census electronics values for WB and Jharkhand

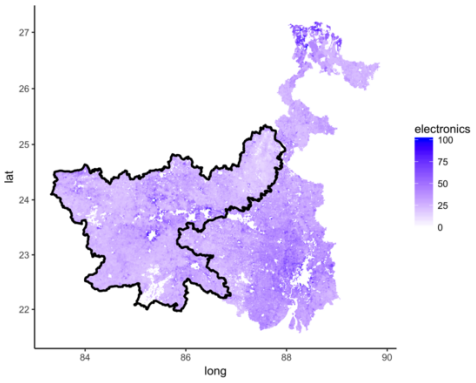


(b) Predicted electronics values for WB and Jharkhand

Figure 12: A comparison between census (12a) and model predictions (12b) for electronics indicator. Model predictions are more continuous as noise from census has been filtered out to show the underlying geographical trends. 12b shows a discontinuity, this discontinuity is actually along the state boundary between WB and Jharkhand.

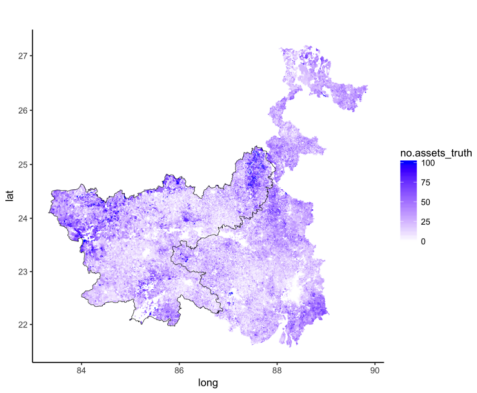


(a) Model predictions for electronics in WB-Jharkhand region

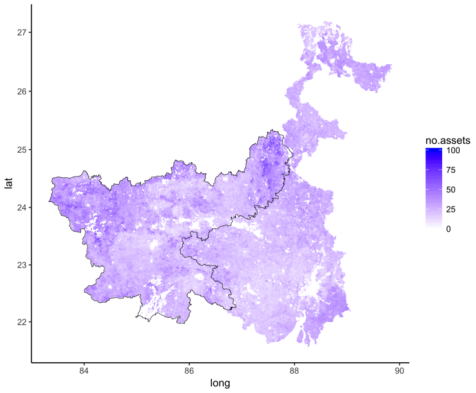


(b) Model predictions overlapped with Jharkhand boundary

Figure 16: Overlapping state boundary shows considerable discontinuity arising along the state boundaries. This merits a closer look by the economists.

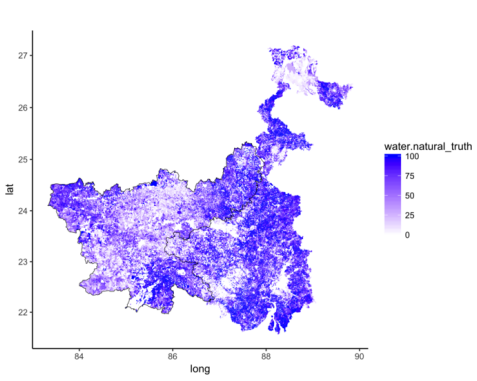


(a) Census no-assets values for WB and Jharkhand

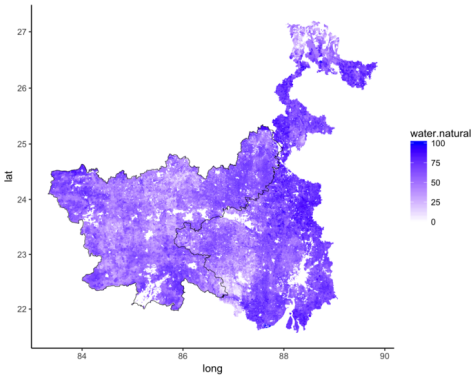


(b) Predicted no-assets values for WB and Jharkhand

Figure 13: A comparison of census and model preditions for no-assets (1) indicator

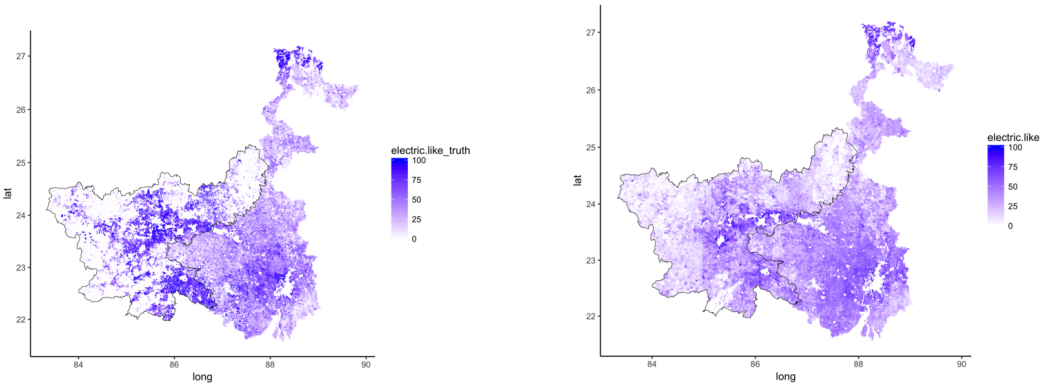


(a) Census water-natural values for WB and Jharkhand



(b) Predicted water-natural values for WB and Jharkhand

Figure 14: A comparison of census and model predictions for water-natural (1) indicator



(a) Census electric-like values for WB and Jharkhand

(b) Predicted electric-like values for WB and Jharkhand

Figure 15: A comparison of census and model predictions for electric-like (1) indicator

6.2 Why this discontinuity?

Figure 16 shows that this discontinuity is along the WB-Jharkhand boundary. This is an interesting result since there is geographical similarity between neighbouring border villages but a considerable gap in economic indicators. This points to a difference arising out of political divisions. Here the reasons can vary from lack of state resources in Jharkhand to the different policies followed by the state governments. However, such questions merit attention of economists.

6.3 The problem with Census data

This application also brings out a problem with census data. While understanding why our r2 scores were low, we discovered that census data has many outliers which our model detected (the points of high loss). These outliers had the characteristic that all their values were either 0 or 100, something which is unexpected from a ground based survey. For instance, some of these villages have 100% electricity based households but 0% households have phones (mobile or landline)!

7 Summary and Conclusion

In conclusion, we have done the following:

- Night light data is of high importance in economics as a ready proxy for economic indicators. Our model achieved a R2score of 0.79 3 for regression into night light values demonstrating the potential of day time satellite imagery as a viable candidate for regression into socio-economic indicators.

- Regression into different census indicators show that some of them can be better predicted than others from satellite images. For example, consider 'transport-motorized' which depends on the presence road and is easily observable from satellite images. So, consequently our model performed better on this indicator.

- A use case was also demonstrated in section 6. Here, it is observed that our model uncovers the underlying geographical trends in socio-economic indicators. This happens because our model incorporates geographical information while making predictions from satellite imagery and thus is able to assign meaningful values to these outlier points (section 6.3).

Appendices

A Transfer learning

We follow the analysis from Pan and Yang [14]. Let's set up the definitions: A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, $X = \{x_1, x_2, ...x_n\} \in \mathcal{X}$. So $\mathcal{D} = \{\mathcal{X}, P(X)\}$. Also given a domain $\mathcal{D}$, a task $\mathcal{T}$ consists of two components: a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$.

Now consider a source domain $\mathcal{D}_\mathcal{S}$ and a source task $\mathcal{T}_\mathcal{S}$ and a target domain $\mathcal{D}_\mathcal{T}$ and a target task $\mathcal{D}_\mathcal{T}$ with $|\mathcal{D}_\mathcal{S}| \ll |\mathcal{D}_\mathcal{T}|$. Then transfer learning involves using $\mathcal{D}_S$ and $\mathcal{T}_S$ to better estimate $f_T(\cdot)$ for $\mathcal{T}_T$.

In our case, the target domain $\mathcal{D}_\mathcal{T}$ and task $\mathcal{T}_\mathcal{T}$ is the set of images and the economic indicators for the region captured in a particular image. However this is a data poor domain implying the number of images with economic indicators are less than what would be needed to learn the objective function $f_T(\cdot)$. Thus we look at other data rich domains, namely the nighttime light intensity domain $\mathcal{D}_\mathcal{S}$ which consists of the nighttime light intensity values for a particular region. This satisfies the condition of $\mathcal{D}_\mathcal{S} \neq \mathcal{D}_\mathcal{T}$, $\mathcal{T}_\mathcal{S} \neq \mathcal{T}_\mathcal{T}$ [3]. These conditions along with $|\mathcal{D}_\mathcal{T}| \ll |\mathcal{D}_\mathcal{S}|$ imply that transfer learning can be applied here.

[3] refer to [14]

B Indian Postal Code Data

This data is collected by Geonames.org [6]. This data contains basically the postal code data along with latitude and longitude corresponding to that region. The main problem we have faced when we started training the model is that we did not know the lat-long coordinates of the regions which have habitation. The target labels are generated from the night time TIFF in section2.2.4. The total distinct data points in this dataset are nearly 60,000 (see Figure17).

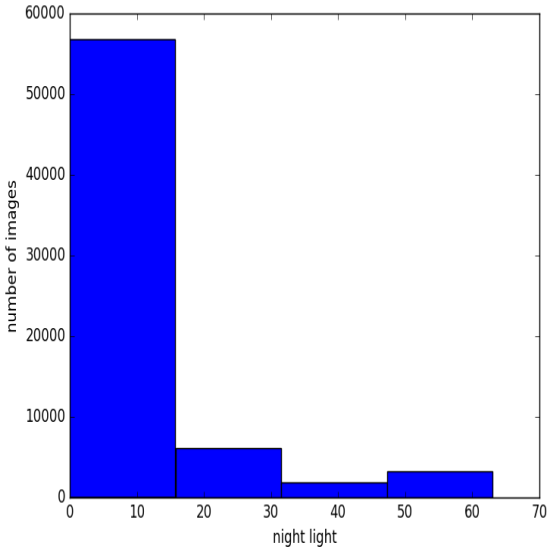


Figure 17: Distribution of nighttime labels of locations from Indian Postal Code data

C Caffe: A brief Introduction

We use Caffe for this task. Let's start with a brief intro to Caffe[12]. Caffe was developed at the Berkeley Vision and Learning Centre (BVLC) as a tool optimized for training deep learning models. It offers huge advantages in terms of speed and modularity of it's architecture. Another key consideration for us was the ease of using GPUs with Caffe. Now let's briefly look at the various components in Caffe.

- **Model definition** describes the architecture of your model. It consists of layers with each layer implements a specific function like Convolution, ReLU, max pooling, fully connected etc. According to it's authors, it contains any directed acyclic graph(DAG) of these layers. For our case, the model file contains a 19 layer CNN derived from the VGG 19 layer network(see Figure. Model file also specifies the location of train and test data.

- **Solver file** describes the configuration used to train your network. It contains various parameters like batch size, learning rate, solver type(Stochastic Gradient Descent, Adam, Nesterov etc.) , step size etc. Setting these parameters to appropriate values is very important and is often the result of careful experimentation. Depending on these values your training can either converge to a local optimum or converge to a sub-optimal value or even diverge.

- **PyCaffe** is a way to access your trained model. It allows you to access

the outputs and parameters of individual layers. You can also modify any specific layer's parameters(as we did, described below) .

Though Caffe is a powerful tool, it does have a drawback, it's lack of documentation makes it challenging to work with. We had to go through Github issues and cafe users group to figure out even some of the basic things.

D Training Nightlights model

Training a model requires us to specify the parameters. The following are some considerations while setting the parameters in model and solver file :-

- **batch size** refers to the number of images to be processed in a batch. A weight update during back-propagation is calculated based on loss evaluated on this batch. Memory considerations require batch size to be small however too small batch size causes weight updates to be erratic and thus make convergence slower.

- **iter** refers to the number of iterations, each iteration involves a forward pass and a backward pass (weight update) of the images in current batch. So total images processed in N iters=N$\times batchsize$. **Epoch** refers to one complete pass over training data.

- **learning rate (lr)** controls the size of weight updates. Too high initial lr causes training to diverge (as we experienced in some cases) while low initial lr causes learning to taper off and never reach optimum.

- **step size** and **gamma** control the lr evolution through the training process. lr is reduced every after step size number of iterations by a factor of gamma (< 1). This is needed because for a particular lr, the loss converges to a value and then plateaus, then after reducing lr further, loss again starts to reduce. Thus setting step sie and gamma properly is very important. Unfortunately there is no theory on how to set these values and it comes down to learning from observations and estimating. In our case we saw a dramatic improvement in accuracy with correct settings.

- **snapshot** describes how often should we save the trained model. If for some reason the training stops in between, we it can be restarted from the saved snapshot.

E Convolution Neural Networks : A brief Introduction

CNNs are biologically inspired variants of neural networks which utilize some special properties of images(or image like data) to vastly reduce the number of parameters and consequently the amount of computation necessary to achieve state of the art performance. Their sparse parameters allow us to stack one layer above another to build very deep architectures. Let's briefly discuss what makes them so powerful for image related tasks:

- **Spatial local correlation** implies the neurons need only be connected with a subset of neurons in the previous layer. This subset consists of spatially contiguous neurons. This is justifiable because most of the information in images is also found in contiguous regions. This reduces the number of parameters per neuron since the corresponding weight matrix W now only contains $F \times F$ parameters where F is the kernel size(size of the spatially contagious set) .

- **Weight sharing** allows each neuron in a layer to share the same weight matrix W. This is justifiable because information in images is translation invariant [4].

Combined together, the above two properties greatly reduce the number of parameters in a CNN and allow us to stack a large number of layers on

[4]Recently CNNs are also being used in NLP tasks[15]

top of each other, thus learning more high level features. Reader who wishes to understand ConvNets(as they are sometimes called) should refer to the excellent treatment from the book [8].

ConvNets can have varied architectures. For our task, we looked at some of the best performing ConvNets from the ILSVRC challenge and decided to work with VGG-19 architecture ([16]). We chose VGG primarily for the simplicity of the architecture. It includes convolution layers which neatly preserve the spatial dimensions of previous layers output. And even though the Google Inception([17]) architecture nearly matches VGG in terms of accuracy but is much faster [5].

[5]Performance for various models have been compared in [9]

F Downloading Google satellite Images

Each image can be downloaded from via a simple http request via google static maps API. The tens of thousands of images have a total size ranging from 10GB to 100GB[6]. Simple python script to download fifty thousand of images took us more than a day for completion. This speed is not sufficient,so a shell script was developed by carefully multi threading different download requests.Interestingly the same download took us few hours. It was observed that few of the download requests, took minutes. But some of them were quick, like in less than second. So,multi threading helped in creating a pipelined approach for download large number of images.

G Basic overview and handling the data

In this project we had to deal with different kinds of data in various formats. Several data sets encountered in the project have two concepts,

1. The fundamental complexity with Geo-located data is due to earth being an ellipsoid. WGS84 is a particular model of ellipsoid,which is a standard for use in GPS(Global Positioning System)[18]. For example,night light data described in section 2.2.4 has Geo-TIFF format,with WGS84 as coordinate system and datum. As Earth is an ellipsoid,so its data is stored generally in the form of a projection on a 2D plane.

[6]Download server of IIT Delhi was used for this purpose

2. GIS(Geographic Information System)[5] maps are essentially shapefiles,shapes of different regions are stored as 'SpatialPolygonsDataFrame'. Every region is made up polygon, lines and data. Many regions together in this format is crudely a 'SpatialPolygonsDataFrame'. Further these standard GIS maps are geo-located using WGS84 coordinate system and latlong as projection.

3. Format in which these data are presented, makes it necessary to use special tools. Firstly because these are projections on earth. Secondly due to the huge size of earth, it's corresponding datasets are also huge. And for example night light data(section 2.2.4) has $\sim$ billion entries.[7] So the necessary libraries are discussed below.

The libraries used are,

1. GDAL(Geospatial Data Abstraction Library) - GDAL is used by Open source GIS packages such as QGIS.Python and R have their own interface to GDAL tool.

2. sp(Spatial Points) - This library has packages to deal with spatial data.The spatial data structures implemented include points, lines, polygons and grids; each of them with or without attribute data.Only R hosts this library.

3. raster - This raster package is not for analysis of usual images.This

[7]From our attempts, we can say that numpy library of python was inefficient in handling data of this magnitude

raster corresponds to Spatial geographic data structure that divides the region(like a geotiff) into rectangles called cells or pixels. Here each cell can store one or more values(any kind of data). This package seperately exits in R and is a part of python's GDAL library.

4. rgeos(Spherical Trignometry) - To compute distances,presence of point in a region,etc., on the GIS maps,some of our calculations should consider that the model of the earth is spheroid(ellipsoid). This package implements functions for calculating area,distance,direction,centroid etc., for geodetic(geographic) coordinates.

Combinations of these libraries along with some standard libraries are necessary to handle some datasets. Libraries used for handling different datasets are,

- For Night Light Data described in section 2.2.4 libraries GDAL and raster are used.
- For GADM data, GDAL is used to handle its shape files.
- For Pitney Bowes Data described in section 2.2.2, libraries GDAL,sp and rgeos are used.
- Rest of the datasets(like census data and Indian Postal code data) are handled using dataframes.

References

[1] *Census houselisting data*. http://www.censusindia.gov.in/2011census/HLO/HL_PCA/Houselisting-housing-HLPCA.html. 2011.

[2] *Census India 2011 Wikipedia*. https://en.wikipedia.org/wiki/2011_Census_of_India. 2015.

[3] K. Chatfield et al. "Return of the Devil in the Details: Delving Deep into Convolutional Nets". In: *British Machine Vision Conference*. 2014. arXiv: 1405.3531 [cs].

[4] Xi Chen and William D. Nordhaus. "Using luminosity data as a proxy for economic statistics". In: *Proceedings of the National Academy of Sciences* 108.21 (2011), pp. 8589–8594. DOI: 10.1073/pnas.1017031108. eprint: http://www.pnas.org/content/108/21/8589.full.pdf. URL: http://www.pnas.org/content/108/21/8589.abstract.

[5] *Geographic information system*. https://en.wikipedia.org/wiki/Geographic_information_system. 2011.

[6] *GeoNames*. www.geonames.org. 2015.

[7] *Global Administrative Areas*. http://gadm.org/about. 2015.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep Learning". Book in preparation for MIT Press. 2016. URL: http://www.deeplearningbook.org.

[9] *ILSVRC models comparisons*. https://github.com/jcjohnson/cnn-benchmarks. 2015.

[10] *Indian Census Data*. http://censusindia.gov.in. 2011.

[11] Neal Jean et al. "Combining satellite imagery and machine learning to predict poverty". In: *Science* 353.6301 (2016), pp. 790–794. ISSN: 0036-8075. DOI: 10.1126/science.aaf7894. eprint: http://science.sciencemag.org/content/353/6301/790.full.pdf. URL: http://science.sciencemag.org/content/353/6301/790.

[12] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[13] *National Geophysical Data Center, Version 4 DMSP-OLS Nighttime Lights Time Series*. http://ngdc.noaa.gov/eog/archive.html. 2013.

[14] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Trans. on Knowl. and Data Eng.* 22.10 (Oct. 2010), pp. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191. URL: http://dx.doi.org/10.1109/TKDE.2009.191.

[15] Sebastian Sierra. "Convolutional Neural Networks for Text Classification". In: (2016).

[16] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[17] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015. URL: http://arxiv.org/abs/1409.4842.

[18] *World Geodetic System*. https://en.wikipedia.org/wiki/World_Geodetic_System. 2011.

[19] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *CoRR* abs/1411.1792 (2014). URL: http://arxiv.org/abs/1411.1792.