Computer Assisted Methods in Engineering and Science
31(3): 351–387, 2024, doi: 10.24423/cames.2024.722

This article belongs to the *Special Issue on AI-Based Future Intelligent Networks and Communications Security* edited by Dr. S. Kumar, Dr. G. Mapp, Dr. A. Bansal and Dr. K. Cengiz

Predicting Wealth Score from Remote Sensing Satellite Images and Household Survey Data Using Deep Learning

Shashank SHEKHAR[1], Pratibha SINGH[2], Rashmi MISHRA[3], Sunil KUMAR[4]

[1] *Edunet Foundation, Gurugram, India*

[2] *Department of Computer Science & Engineering, Krishna Engineering College, Ghaziabad, India*

[3] *Delhi Technological University, India*

[4] *Institute for Communication Systems, University of Surrey, United Kingdom*

* *Corresponding Author e-mail: sk4sunilkumar@gmail.com*

The most exigent call of the United Nations' 17 sustainable goals is to end poverty everywhere by 2030. Unlike in the past, when poverty was measured based on data collected through ground-level surveys, the new technology adopted by many developing and developed countries is to estimate the poverty index using remote sensing satellite images with the help of machine learning techniques. Our approach demonstrates the prediction of cluster wealth score and establishes the relationship between wealth score obtained from Demographic and Health Survey (DHS) data and remote sensing satellite images of India by calculating Pearson's correlation coefficient (r^2). The implementation results have been analyzed in four phases. Phase 1 comprises four regression models (RMs): Ridge, RANSAC, Lasso, and k-nearest neighbor for feature extraction from a pre-trained convolutional neural network model using daylight & nightlight images. Here, the Lasso RM outperforms the others and is best suited for predicting the wealth score. Phase 2 categorizes daylight images with DHS data, where the Lasso RM efficiently generates the cluster wealth score. Phase 3 focuses on images of specific regions of Delhi, Tamil Nadu, Maharashtra and Telangana, using the Lasso RM, as it emerged as the best predictor of cluster wealth score in the previous two phases. Phase 4 compares the results attained through our proposed model with existing results.

Keywords: convolutional neural network, Demographic and Health Survey Data, Inception V3, transfer learning.

1. Introduction

The word poverty comes from the old Latin word *pauper*, which means *poor* in English. According to the World Bank, poverty can be regarded as the deprivation of well-being, including low income and the inability to acquire the basic goods and services essential for survival with dignity. The eradication of extreme poverty in all its forms is the major goal of the world's leading organizations, such as the United Nations (UN), the World Bank, etc. The mission of eliminating poverty is rooted in all the analytical, operational and united work in more than 145 client countries. Despite significant strides in reducing extreme poverty, the rate of reduced poverty remains high in low-income countries and those affected by conflict and political instability. Progress has been marked in reducing poverty, which is the first of the world's Sustainable Development Goals (SDGs) as mentioned by the UN [1].

As India is a developing nation, poverty is declining in the country with each passing day. According to Brookings reports, it is estimated that "India may become the third-largest economy by 2030 and poverty will fall below 10 percent by 2025". The reported data also showed that India will be out of the list of top 10 countries in terms of extreme poverty by 2030 [8]. The most common method used in India to estimate poverty is typically based on income or consumption levels, and if either parameter falls below the given minimum level, the person is considered to be in debt or living below the poverty line.

The government and its numerous bodies run several programs and implement different policies to alleviate poverty. Economists and intellectuals have different opinions regarding the estimation of poverty, which has led to divergent views and confusion in the country. In India, the earlier method used to estimate poverty involved collecting survey data on the ground, which was not always accurate, was a costly method of data collection, and consumed a lot of time [2]. The most novel approach used nowadays to estimate the poverty index is using techniques such as machine learning (ML) and deep learning (DL) combined with remote sensing satellite data. This method effectively overcomes the shortcomings of traditional data collection and is able to project poverty at a large scale in developing countries. The scheme has been successfully implemented and has yielded better outcomes in several African countries like Nigeria, Uganda, Tanzania and Rwanda.

In this paper, we have effectively used a pre-trained convolutional neural network (CNN) model on daylight satellite images for predicting the wealth score of different regions. We have also observed that there is a strong correlation between predictions obtained through daylight images with nightlight images and daylight images labelled using household survey data. The contributions of this paper are as follows:

1. Exploring the role of recent developments in ML for predicting the wealth score of regions in any country. This is prerequisite for utilizing limited available resources to reduce poverty in the most affected regions. It will help in attaining the most significant SDG of the UN, emphasizing the importance of addressing this problem and exploring its solutions.
2. Analyzing CNN's performance along with four different RMs: Ridge, Lasso (least absolute shrinkage and selection operator), RANSAC (random sample consensus) and KNN (k-nearest neighbor) for identifying the correlation of wealth score obtained from household survey data with high-resolution satellite images (daylight and nightlight) of various regions of India.
3. Learning robust features from daylight images categorized using nightlight images or survey data as available, using a pre-trained Inception V3 model. Transfer learning (TL) solves the problem of availability of insufficient labeled data and reduces the computational efforts of training a network from scratch.
4. As we know, collecting household data through ground-level surveys is very expensive and time-consuming, the use of TL has been encouraged by combining a CNN with different RMs to predict wealth score utilizing satellite images, thus reducing the role of survey data.
5. Forecasting wealth score will help government policy makers and NGOs to effectively allocate resources to regions with deficiencies and improve their economic conditions.

2. Literature survey

The scarcity of data concerning livelihood, social, health and environmental issues produces unreliable information regarding economic livelihoods in the developing world [3]. Due to the low number of surveys, developing countries have very little data to make significant decisions. In contrast, developed countries have a lot of data to measure poverty. In the list of SDGs released by the UN, poverty is the first of the 17 objectives to be eradicated by 2030 [8]. Traditionally, the major source for measuring poverty was collecting data from ground-level surveys such as household consumption and wealth surveys. However, these methods were inaccurate, tedious, and costly. The living standard measurement survey (LSMS) was initially used in the context of ground-level surveys. Household expenditure was measured using LSMS approach. The limitation of LSMS surveys was that they did not provide comprehensive information needed to accurately calculate the poverty in a particular region [4, 5].

The use of mobile data and digital footprint has also emerged as a method to calculate poverty. Frequent data on societal, economic, and livelihood conditions is required to track SDGs and target poverty-related information. New technologies such as ML and data science have now evolved as new measures to predict poverty. Researchers have begun to develop these techniques to calculate poverty [6]. One of the most highly structured, easily available, and inexpensive data sources is remote sensing data, such as satellite imagery. Using satellite imagery makes it easier to minimize the data gap between actual values and predicted values. For this purpose, a CNN is trained to estimate poverty levels using satellite imagery [7, 8]. With ML techniques, it is also possible to establish the relationship of predicting poverty over several years. These techniques learn visual features from satellite imagery which indicate the poverty levels in a region.

Several experiments also suggest that poverty prediction from different and multiple parameters gives more accurate results compared to using a single parameter [9]. Object identification, image segmentation and labelling of satellite images help to identify visual features to estimate poverty. The United States Government's Landsat 7 program was used to acquire publicly available satellite imagery. The model is pre-trained on one of the datasets, and further, this model can be trained to predict nightlight intensity corresponding to daylight satellite imagery. Daytime satellite imagery can be used to estimate local per capita outcomes [4]. Features learned from one model can be used to train other models. This technique is known as transfer learning (TL). A Google Geocoding API is employed to predict the location of a place's center as well as latitude and longitude. Multi-task learning involves learning multiple tasks simultaneously by analyzing the similarities and differences among the tasks [10, 11]. The suitable input image size for a model is 1920×1920 pixels. For predicting wealth, nightlight images serve as a good proxy. Socio-economic indicators can be best extracted with the help of raw satellite images. There is a need for more frequent data that can learn features from the TL approach to monitor the range of issues covered by 17 goals and 169 targets [12]. The TL approach not only predicts the nightlight intensity corresponding to daylight satellite images but also helps in mapping the poverty level.

The ultimate goal of TL approach is to transfer the knowledge from daylight satellite imagery to the target problem for predicting nightlight intensity. TL is highly scalable and does not require human annotation. However, nightlight intensity faces difficulties in distinguishing between densely populated, poor and sparsely populated areas, making it unsuitable for estimating per capita consumption [13]. The multitask model can be used for predicting poverty and at the same time evaluating the similarities and differences among the tasks. The multitask fully convolutional model comprises convolutional layer, ReLU activation layer, batch normalization layer, pooling layer, and a dropout layer [14].

The model is able to establish the relationship between specific graphic structures and growth factors. These features reflect the relationship between ground-truth values and predicted values. Pearson's correlation coefficient (PCC) calculation is used to assess model accuracy. Differences in the relationship between household data and remote sensed data are evident at the community level [8]. The prediction of household-level wealth in a particular area can be visualized with the help of a classification tree. To check whether the TL model improves with nightlights, 210 trials of 10-fold cross-validation were conducted. Another model to predict poverty is the Wikipedia embedding model, trained using the mean squared error loss function and Adam optimizer. The Wikipedia learning model outperforms using nightlights only [31]. The poor performance of the model is due to the limitations of the nightlight baseline. To forecast poverty predictions across years and determine their accuracy a recurrent neural network can be integrated on top of a CNN.

To revolutionize social science, remote sensing data presents new potential, especially with the night-time light dataset that provides night-time luminosity from 1993 to 2013 [17]. It evaluates the night-time light data against the rural electrification data from the 2011 Census of India. The results suggest that these night-time satellite images are accurate for estimating rural electrification. The accuracy is improved with better GIS maps, basic geoprocessing tools, and night-time luminosity aggregation [6]. The nighttime satellite images and census data were also used to correlate light intensity from images with state-wise poverty, population, GDP, and forest cover. The ARIMA model is used for forecasting the census data. The output is measured in terms of PCC with r^2 value of four states of India namely [19]:

1. Uttar Pradesh (0.68),
2. Himachal Pradesh (0.68),
3. Jharkhand (0.37) and
4. Gujarat (0.25).

The summary of literature survey is shown in Table 1. As seen in the literature survey of different countries, we conclude that wealth score of any area can be predicted with the help of satellite images such as daylight and nightlight images. There are several other methods also to predict the poverty of any area such as LSMS Surveys, Object Identification, UN Govt. Landsat's Program, Wikipedia Embedding Model, and Tracking of Mobile Call Records. Other technologies involved the use of ARIMA model. But the best among all that we have seen in the literature survey is the use of daylight and nightlight images that accurately predicts the wealth score or poverty of any area.

TABLE 1. Description of literature survey.

<table>
<tr><th>Title</th><th>Location</th><th>Datasets</th><th>Algorithm</th><th>Result (adjusted r^2)</th><th colspan="3">Parameter</th></tr>
<tr><td>Nischal <i>et al.</i> [6]</td><td>India</td><td>1) DMSP-OLS satellite images from the National Geophysical Data Center (NOAA),
2) State-wise GDP and population data for India from 2000–2012 from the Ministry of Statistics and Programme Implementation</td><td>Multivariate RM,
The ARIMA (Auto Regressive Integrated Moving Average) Model used for Future Census Forecasts</td><td>Uttar Pradesh (0.68),
Himachal Pradesh (0.68),
Jharkhand (0.37),
Gujarat (0.25)</td><td colspan="3">Coefficient of determination</td></tr>
<tr><td rowspan="2">Sheehan <i>et al.</i> [12]</td><td rowspan="2">Malawi,
Nigeria,
Tanzania,
Uganda,
Ghana</td><td rowspan="2">1) Demographic and Health Surveys (DHS) data,
2) Afrobarometer (Round 6): a survey across 36 countries in Africa, accessing various aspects of welfare, education and infrastructure quality,
3) Geolocated Wikipedia articles,
4) Visible Infrared Imaging Radiometer Suite (VIIRS) nightlight data</td><td rowspan="2">Nightlight-Only,
Wikipedia embedding Model,
Multi-modal Model</td><td>Location</td><td>Nightlight only</td><td>Wikipedia embedding</td><td>Multi-modal</td></tr>
<tr><td>Ghana
Malawi
Tanzania
Nigeria
Uganda</td><td>0.41
0.24
0.6
0.3
0.53</td><td>0.47
0.49
0.64
0.52
0.57</td><td>0.76
0.64
0.71
0.7
0.76</td></tr>
<tr><td>Pandey <i>et al.</i> [13]</td><td>India</td><td>1) 2011 census data (Uttar Pradesh Province),
2) Google Static Maps API</td><td>Multi-task fully CNN</td><td>0.82</td><td colspan="3">Coefficient of determination</td></tr>
<tr><td>Duan <i>et al.</i> [14]</td><td>New York City (USA)</td><td>1) Felony Dataset – contains several incidents, with the properties of coordinates, offense type and time,
2) 311 datasets</td><td>CNN, Support Vector Machine (SVM), Random Forest (RF), SpatioTemporal Crime Network (STCN)</td><td>SVM (0.74),
RF (0.82),
STCN (0.92),
SFCNN (0.82)</td><td colspan="3">F1 Score, AUC ROC</td></tr>
<tr><td>Ruchir [16]</td><td>Malawi,
Nigeria,
Tanzania,
Uganda</td><td>1) Living Standard Measurement Survey (LSMS) by the World Bank,
2) Landsat 7 imagery (US Govt.),
3) VIIRS Nightlight Data</td><td>ResNet,
Ridge RM</td><td>Malawi (0.55),
Nigeria (0.68),
Tanzania (0.57),
Uganda (0.69)</td><td colspan="3">Coefficient of determination</td></tr>
</table>

Table 1. [Cont.].

Title	Location	Datasets	Algorithm	Result (adjusted r^2)	Parameter
Eugenie *et al.* [2]	India	1) India Lights Project's API to download village-level data, 2) Nighttime Lights Dataset (by NOAA) Linear	Linear regression	0.63	Coefficient of determination
Neal *et al.* [3]	Rwanda	1) Demographic & Health Surveys (DHS) – the primary source for population and health statistics, 2) Expenditure data from the World Bank's LSMS surveys	CNN transfer learning	0.75	Coefficient of determination
Xie *et al.* [19]	Uganda	1) ImageNet, 2) DHS Survey	CNN transfer learning	71% accuracy	Cross-validation scheme test
Watmough *et al.* [20]	Sauri (western Kenya)	1) Remote sensing data/ satellite data, 2) Daylight images	Classification	0.62	Coefficient of determination
Perez *et al.* [21]	Uganda	1) Landsat 7 (African Continent), 2) DHS data (Africa)	Wasserstein generative adversarial networks (GANs)	0.57	Coefficient of determination
Piagessi *et al.* [22]	Santiago, Los Angeles, Chicago, Boston, Houston, Philadelphia	1) Household income data (Santiago, Chile), 2) Satellite data from DigitalGlobe web platform, 3) Satellite images (Google Static Maps API)	ResNet 50, VGGF, VGGF+Nightlights	Santiago (0.52) Los Angeles (0.56) Chicago (0.36) Boston (0.37) Houston (0.327) Philadelphia (0.46)	Coefficient of determination
Steele *et al.* [23]	Bangladesh	1) DHS data 2011, 2) National household Surveys, 3) Call detail record (Grameenphone)	Hierarchical Bayesian geo-statistical methods	0.78	Coefficient of determination
Catamuro *et al.* [24]	Kenya	Road quality measurement and corresponding satellite imagery	CNN, transfer learning, ResNet, AlexNet, VGG-11	0.79 0.35	Regression standard training-test split, Harder held-out regression problem

Table 1. [Cont.].

Title	Location	Datasets	Algorithm	Result (adjusted r^2)	Parameter		
Ajami *et al.* [9]	Bangalore	Very high resolution (VHR) satellite images and Set of geographic information system (GIS) layer	VGG	0.75	Coefficient of determination		
Guie *et al.* [15]	China	1) DMSP-OLS satellite images from National Geophysical Data Center (NGDC)	Gaussian process with radial basis function kernel (GPRBFK), partial least squares regression for generalized linear models (PLSRGLM)	GP (84.6%), PLS (82.8%), RF (83.3%)	Accuracy		
Subash *et al.* [17]	India	1) India Lights API, 2) Satellite Images from the Defense Meteorological Satellite Program (DMSP), 3) Per capita GDP	Artificial neural network, ML algorithms	Root Mean Square Error ML (nightlights), ML (per capita GDP)	RSME 7.38 8.29	r^2 0.59 0.56	 – –
Suraj *et al.* [25]	India	1) Census 2011, 2) Daylight Satellite Images (Google Static Maps API) (2017), 3) Nightlight data (Defense Meteorological Satellite Program's Operational Linescan System-DMSP-OLS)	VGG-CNN-S	0.65	Coefficient of determination		

3. Proposed method

The Phase 1 model, presented in Figs. 1 and 2, initiates with Demographic and Health Survey (DHS) data, i.e., household data (containing records for everyone) and geographical data (DHS data linked with health and infrastructure). By processing these two files, cluster and household data are generated, which contain wealth score, cluster wealth, latitude, and longitude for different locations. The raw survey data is then used to produce the more refined cluster and household data, containing wealth score, nightlight intensity, number of households, household ID, and cluster ID. The extracted cluster data is managed to store files that hold the values of each cluster size and the values of nightlight. Daylight images are downloaded from the signed URL of the HERE Maps API, sized at 400 × 400 pixels using generated image coordinates. Later, the size is reduced to 299 × 299 pixels, and the files generated contain the downloaded dataset locations. Similarly, from the DMSP-OLS, the nightlight images are downloaded at size 43 201 × 18 801 pixels.

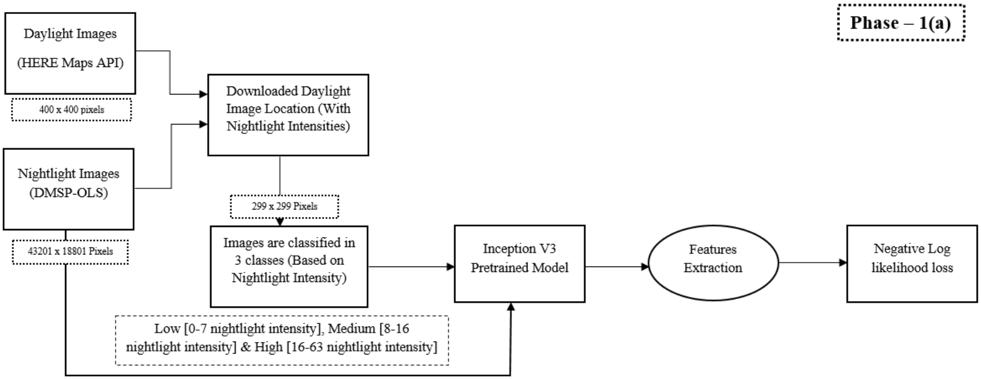

Phase – 1(a)
Daylight Images
(HERE Maps API)
400 x 400 pixels
Downloaded Daylight Image Location (With Nightlight Intensities)
Nightlight Images (DMSP-OLS)
43201 x 18801 Pixels
299 x 299 Pixels
Images are classified in 3 classes (Based on Nightlight Intensity)
Inception V3 Pretrained Model
Features Extraction
Negative Log likelihood loss
Low [0-7 nightlight intensity], Medium [8-16 nightlight intensity] & High [16-63 nightlight intensity]

Fig. 1. Phase 1(a) for the training model with TL.

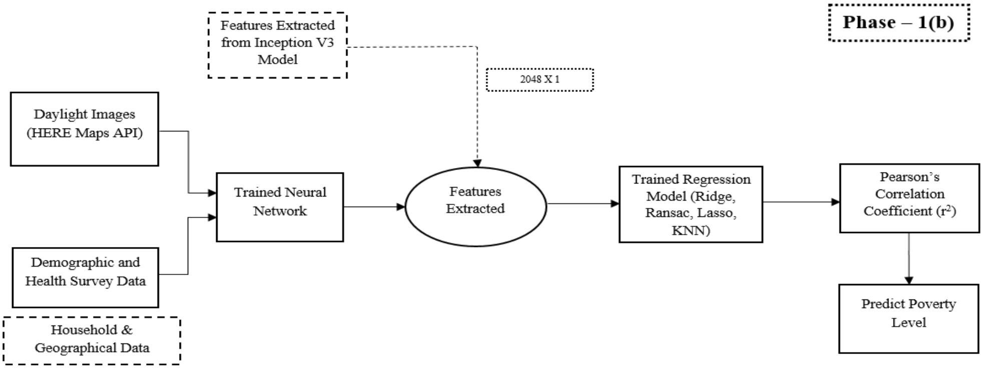

Phase – 1(b)
Features Extracted from Inception V3 Model
2048 X 1
Daylight Images (HERE Maps API)
Trained Neural Network
Features Extracted
Trained Regression Model (Ridge, Ransac, Lasso, KNN)
Pearson's Correlation Coefficient (r²)
Demographic and Health Survey Data
Household & Geographical Data
Predict Poverty Level

Fig. 2. Phase 1(b) for the regression model and predicting poverty.

The downloaded daylight images are then divided into three classes: low, medium, and high. With the help of the pre-trained Inception V3 model, we train our CNN that establishes the relationship between training, testing loss and accuracy. The features of satellite images are now extracted with the help of the trained model along with latitude and longitude values. Regression models such as Lasso, Ridge, RANSAC and KNN are trained using features generated from the trained CNN model and DHS data. We can use this DHS data to plot a single-panel regression value graph between model prediction and actual assets value for all four RMs. In the end, confusion matrices with normalization and without normalization are plotted for the trained model.

Similarly, in Phase 2 depicted in Fig. 3, the model initiates with daylight images only to predict poverty. These daylight images are classified and trained with the help of a pre-trained network, and different regression models are then applied to the daylight images only to predict poverty using PCC (r^2).

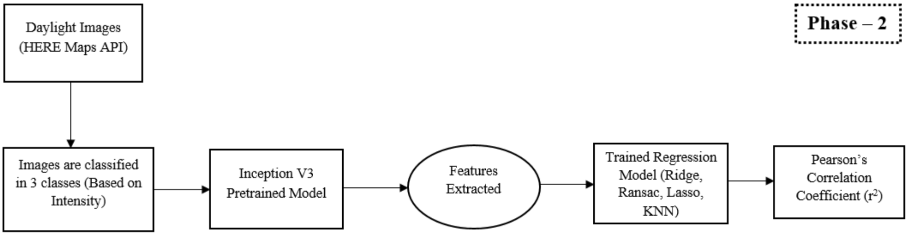

Daylight Images (HERE Maps API)
Phase – 2
Images are classified in 3 classes (Based on Intensity)
Inception V3 Pretrained Model
Features Extracted
Trained Regression Model (Ridge, Ransac, Lasso, KNN)
Pearson's Correlation Coefficient (r²)

FIG. 3. Phase 2 for the training, regression and prediction of poverty.

The Phase 3 model proposes a method for the prediction of poverty, where daylight images and nightlight images of Delhi/ NCR, Tamil Nadu, Maharashtra, and Telangana regions are considered. The locations of images are identified with the help of latitude and longitude values generated. These images are then divided into three categories: low, medium, and high, based on their nightlight intensity. These images are then processed through the Inception V3 pre-trained model for feature extraction. To predict poverty in all three regions, their images will be trained with the Lasso RM using PCC (r^2).

3.1. Regression models

3.1.1. The Ridge regression model (Ridge RM). It is a technique for handling multiple regression data that suffer from multicollinearity issues. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will give estimates that are more reliable. The cost function for ridge regression is shown in Eq. (1):

$$\sum_{i=1}^{M}(y_i-\widehat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j*x_{ij}\right)^2+\lambda\sum_{j=0}^{p}w_j^2, \tag{1}$$

where x – the predictor variable, y – the response variable, $\widehat{y}_i$ – the predicted value of y for the i-th instance, M – the number of instances in the dataset, λ – the regularization penalty, p – the number of features, and w – the number of coefficients.

In ridge regression, the cost function is transformed by penalty term that is equal to the sum of squared coefficients multiplied by a tuning parameter λ. It puts a limitation on the coefficients in such a way that a penalty is imposed on the coefficient function if it takes large values. With ridge regression, the value of coefficients is minimized and the complexity of the model is reduced. If the value of λ becomes 0, the cost function of ridge regression reduces to the linear model cost function [27].

3.1.2. The Lasso regression model (Lasso RM). Lasso RM is very similar to Ridge regression. While Ridge regression penalizes the sum of squared coefficients (L2 penalty), Lasso penalizes the absolute value sum (L1 penalty) for non-zero coefficients. The loss function for Lasso regression is shown in Eq. (2):

$$\sum_{i=1}^{M}(y_i-\widehat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\,*\,x_{ij}\right)^2+\sum_{j=0}^{p}|w_j|. \tag{2}$$

Equation (2) differs from Eq. (1) in that Lasso regression uses the magnitude of coefficients in place of the square of coefficients. This means that many coefficients may be disregarded. Lasso RM helps in feature selection as well as reducing overfitting [27].

3.1.3. The RANSAC regression model (RANSAC RM). It is a learning technique used to estimate model parameters for observed data through random sampling. If a dataset contains both inliers and outliers, RANSAC RM finds the optimal fitting result by using a voting scheme [28].

3.1.4. The KNN regression model (KNN RM). It is used for both classification and regression problems. To predict the value of new points of data, the concept of feature similarity is used. The new value of the data point is assigned based on the closest point in the training data [29].

3.1.5. Pearson's correlation coefficient (PCC). PCC is a statistical measure used to compute the linear correlation between two variables: X and Y.

The value for PCC ranges between 1 and -1, where $+1$ is total positive linear correlation, 0 is no linear correlation and -1 is total negative linear correlation. PCC is calculated as the covariance of the two variables X and Y divided by their standard deviations. The formula for PCC is given in Eq. (3)

$$\rho_{X,Y} = \frac{E\left[XY\right] - E\left[X\right]E\left[Y\right]}{\sqrt{E\left[X^2\right] - \left[E\left[X\right]\right]^2}\sqrt{E\left[Y^2\right] - \left[E\left[Y\right]\right]^2}}, \tag{3}$$

where $E[X]$ – the mean of X scores, $E[Y]$ – the mean of Y scores, and $E[XY]$ – the mean of the product of paired score.

3.2. Inception V3 architecture

Inception V3 is a CNN trained on an ImageNet dataset [26] to enhance image analysis and object detection. This ImageNet Dataset is comprised of more than 14 000 000 images. The size of each input image in the dataset is 299 × 299 pixels. Inception V3 model is a well-designed convolutional module that produces discriminative features and reduces the number of parameters. The inception module is composed of several convolutional layers, pooling layers, and fully connected layers. This model consists of a total of 32 convolutional layers of different sizes, 5 (3 avg and 2 max) pooling layers, and 5 fully connected layers with different numbers of neurons, including 2048, 2048 and 210. The base network of Inception V3 is formed with the combination of convolutional and pooling layers [18].

The output of the Inception V3 network consists of 210 classes. The layer-wise classification of the Inception V3 model is given above in Table 2.

Table 2. Inception V3 architecture [18].

Model	nCLs	Filter sizes	nPLs	Filter sizes	nFCs	No. of neurons
Inception V3	15	1 × 1				
	7	3 × 3				
	3	1 × 7	3 (avg)	3 × 3	3	2048, 2048, 210
	3	7 × 1	2 (max)	3 × 3		
	1	1 × 3				
	3	3 × 1				

3.3. Dataset description

3.3.1. Daylight images. The daylight satellite images used in the paper are downloaded from the HERE Maps API, where the dimension of each image is 400 × 400 pixels, and there are 60 000 images. The obtained latitudes and longitudes help to download the images from the HERE Maps API [10].

3.3.2. ***Nightlight intensities.*** The nightlight images are used to obtain the nightlight intensities. These nightlight images are taken from DMSP-OLS data source accessible from 1990s to the present. Each image is of size 43 201 × 16 801 pixels [4].

3.3.3. ***Cluster wealth score.*** The wealth index is calculated as an integral part of the inspection of goods. An inquiry is made into questions about the ownership of common items such as bicycles, televisions, and materials used for house construction. These parameters are normalized for each country and therefore we do not need further generalizations [5].

4. Implementation and results

We tested the same concept with a fixed learning rate (LR) value of 0.001 and 0.0001. We predicted the accuracy of the model on a pre-trained model (true and false) with these two LR values. Ridge RM was tested for predicting the accuracy and cluster value. With an LR value of 0.001 and pre-trained set to true, the best test accuracy achieved was 78.54%.

System configuration used for implementing the proposed approach is shown in Table 3.

Table 3. System configuration.

Python	3.8.1
Jupyter notebook	6.0.3
R	3.6.2
Processor	I7-9750H CPU (9th Gen)
RAM	16 GB DDR4
GPU	4GB GTX 1650

4.1. Phase 1: daylight + nightlight images

Table 4 presents the results of training the model with the help of daylight and nightlight Images. The model is trained with LR values set to 0.02, 0.42, and 0.89, and the momentum and epoch are kept at 0.9 and 20, respectively. The Inception V3 retrain model is used to train our model, and RMs are trained to predict the wealth score. As shown in Table 4, accuracy and loss analysis

Table 4. Accuracy and loss analysis for different values of learning rate.

Daylight + nightlight images						
Model	Accuracy and loss					
	Batch size = 64					
Pre-trained	Momentum/Epoch	Learning rate	Training accuracy	Training loss	Test accuracy	Test loss
True	0.9/20	0.02	99.1	0.35	80.1	1.7
		0.42	94.7	0.51	75.3	1.16
		0.89	88.7	1.16	76.9	1.11
	0.9/30	0.001	99.9	0.35	78.5	92.8
		0.0001	65.7	1.11	64.2	77.4
False	0.9/30	0.001	55.1	1.28	58.7	87.1
		0.0001	49.5	1.38	50.5	96.3

is done on different values of LR, i.e., 0.02, 0.42, and 0.89, revealing that the training test accuracies are higher when the LR value is low.

The value of test loss decreases with decreasing LR in some cases, However, this is not consistent across all LRs. We infer from the table that training and testing accuracy are higher when the value of LR is low. The results for LR values of 0.001 and 0.0001 for both pre-trained = true and pre-trained = false were also included [2].

LR is a hyper-parameter that controls the weights of the neural network with respect to the loss gradient. It defines how quickly the neural network updates the concept it has learned. Adaptive LR allows the training algorithm to monitor the performance of the model and automatically adjust the LR for the best performance. Neural networks with the help of adaptive LR can usually outperform those fixed values of LR.

4.1.1. Learning rate: 0.02. As shown in Fig. 4, the training loss gradually degraded from the 1st to the 7th iteration, then slightly curved down from the 6th to the 8th iteration before reaching a saturation state after the 9th iteration. The test loss shows frequent fluctuations until the 7th iteration, after which it starts to increase gradually and reaches saturation after the 13th iteration. On the other hand, the training accuracy gradually increased from the very first iteration until the 7th iteration and after the 7th iteration, it remained constant due to the fixed LR.

The test accuracy shows repeated fluctuations from different iterations and stabilizes from the 7th iteration without further deflection. Figure 5 presents the training of the model with the help of four different RMs: Ridge, RANSAC,

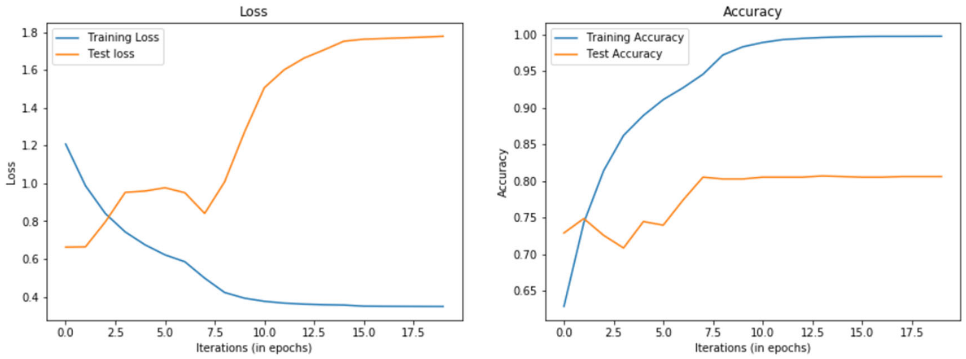
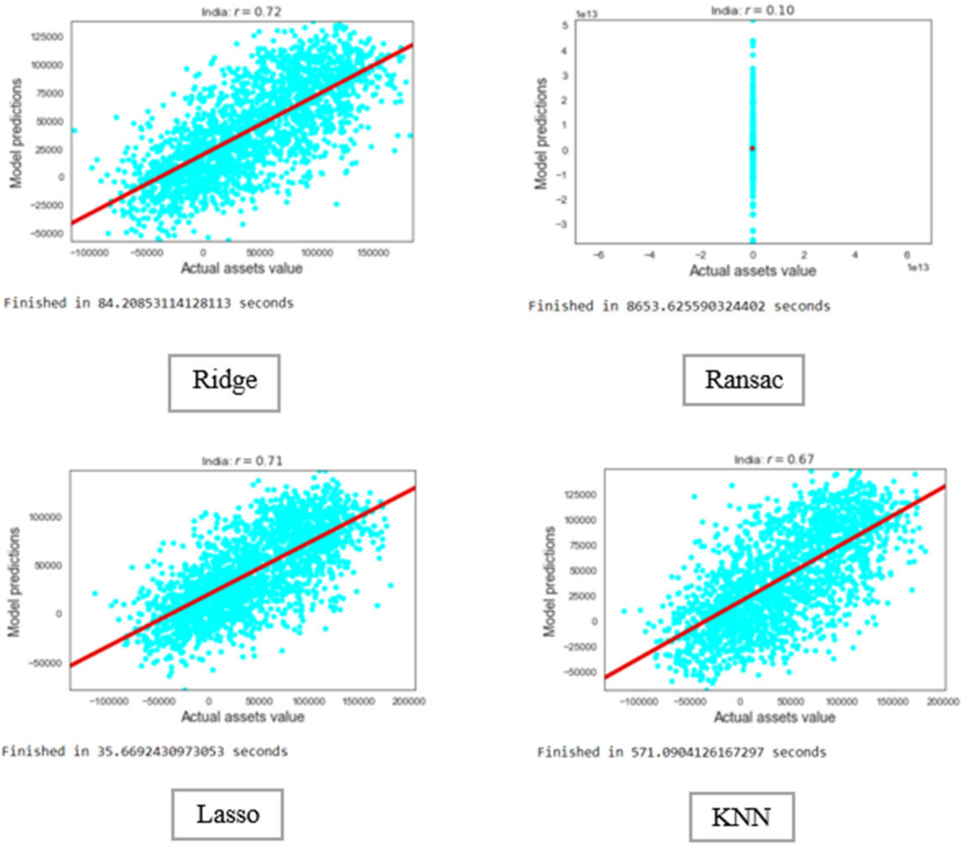


FIG. 4. Iteration vs. loss and accuracy graph.



```
Finished in 84.20853114128113 seconds
```

Ridge

```
Finished in 8653.625590324402 seconds
```

Ransac

```
Finished in 35.6692430973053 seconds
```

Lasso

```
Finished in 571.0904126167297 seconds
```

KNN

FIG. 5. Comparison of clusters with different RMs for predicting wealth score.

Lasso and KNN, where the Lasso RM requires significantly less time to train the model, achieving an r^2 value of 0.71.

For all three class labels (low, medium and high), confusion matrices with and without normalization are shown above in Figs. 6a and 6b.

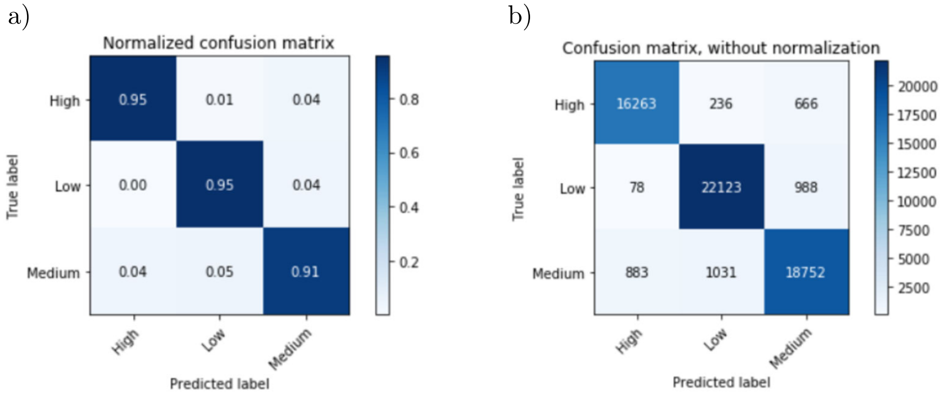


FIG. 6. Confusion matrices with standardization (a) and without standardization (b).

4.1.2. Learning rate: 0.42. As presented in Fig. 7, the training loss o starts from a very high value and then decreases after experiencing a sharp downward deflection at the 6th iteration, reaching saturation after the 11th iteration. Contrarily, the test loss starts with a value of 1 and without showing very much deflection it starts increasing very slowly after the 12th iteration. At the same time, the training accuracy starts from a very low value and exponentially increases from the very first iteration until the last iteration.

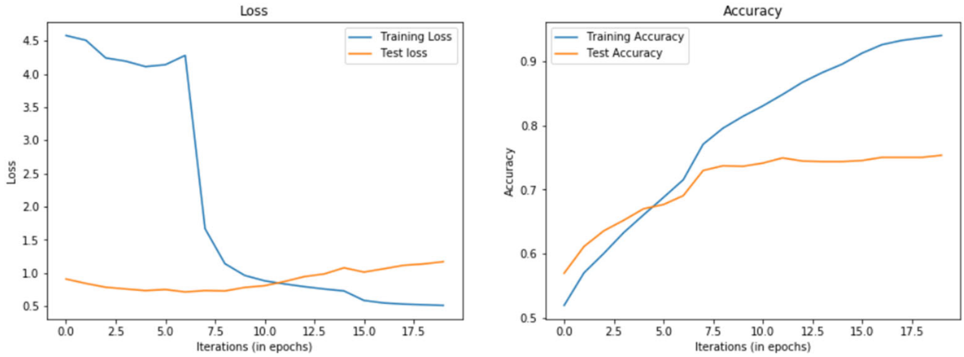
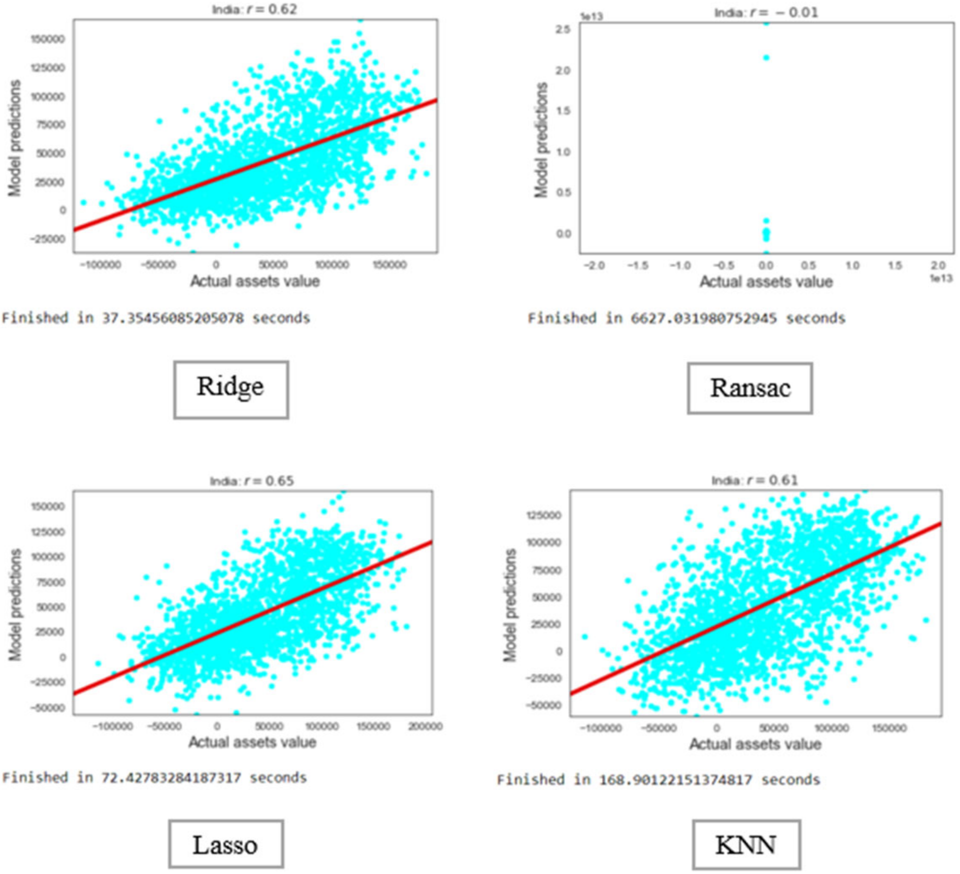


FIG. 7. Iteration vs. loss and accuracy graph.

Conversely, the test accuracy gradually increases at the beginning but shows some deflection due to the slowly decreasing LR and fixed momentum between the 5th and 10th iterations, after which the test accuracy saturates from the 11th iteration onward.

The results depicted below in Fig. 8 show the performance differences between various RM with the LR set at 0.42. The PCC (r^2) for Ridge, Lasso, RANSAC and KNN RMs is 0.62, 0.65, -0.01 and 0.61, respectively. A negative PCC here signifies that the straight-line model through the data explains $0.01^2 = 0.0001 = 0.99\%$ of the data variance, and the (–) signifies that the slope of the graph is negative.



```
Finished in 37.35456085205078 seconds
```

```
Finished in 6627.031980752945 seconds
```

```
Finished in 72.42783284187317 seconds
```

```
Finished in 168.90122151374817 seconds
```

FIG. 8. Comparison of clusters with different RMs for predicting wealth score.

Here, the Lasso RM outperforms the other RMs, achieving the highest accuracy of the model with an r^2 value of 0.65.

Confusion matrices for three output classes (low, medium, and high) are shown in Figs. 9a and 9b.

4.1.3. Learning rate: 0.89. As shown in Fig. 10, the training loss deflects sharply and frequently initially, followed by sharp decrease from the 6th iteration, and reaching a saturation state after the 8th iteration onward.

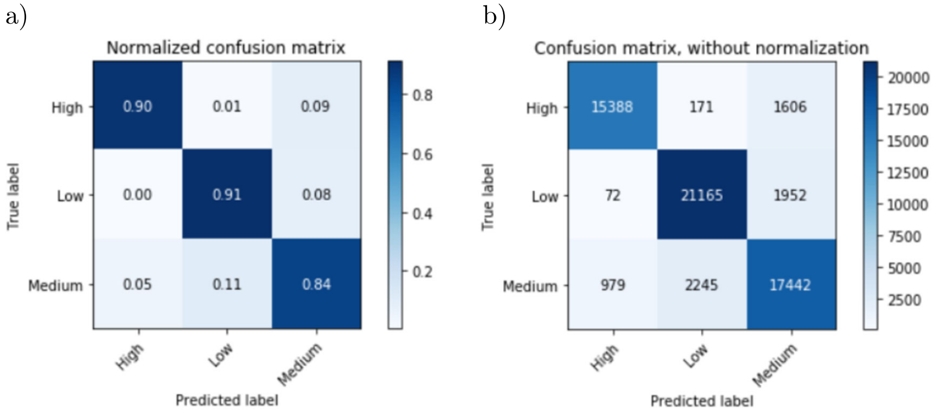


FIG. 9. Confusion matrices with standardization (a) and without standardization (b).

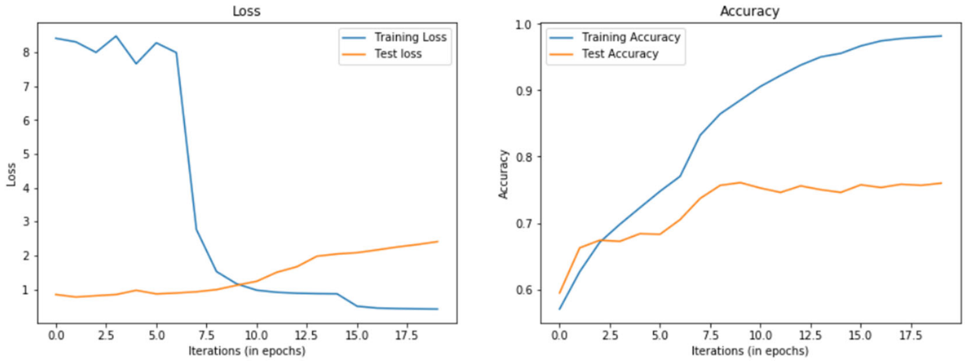
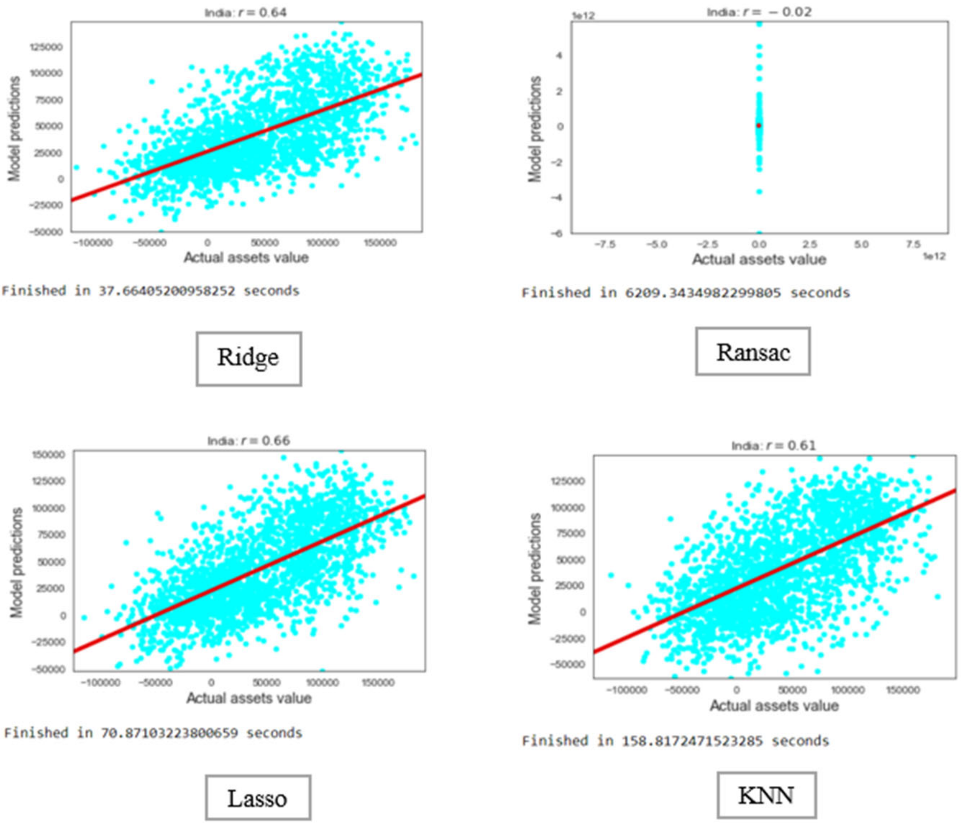


FIG. 10. Iteration vs. loss and accuracy graph.

This is due to the fast-paced training of the model with high LR of 0.89. Conversely, the test loss starts increasingly very smoothly from the very first iteration, and by the 13th iteration, it reaches a saturation state with some deflection. Additionally, the training accuracy exponentially increases from the very first iteration to the last iteration. However, the test accuracy shows repeated fluctuations with each iteration. It keeps on changing its values regardless of the fixed LR and momentum.

Figure 11 shows performance difference in four RMs with an LR value of 0.89. The PCC (r^2) is 0.64, 0.66, -0.02, and 0.61 for Ridge, Lasso, RANSAC, and KNN RMs, respectively. Among these, Lasso RM with LR 0.89 proved to be the best RM that predicted the best accuracy on the dataset provided. RANSAC RM again shows a negative correlation coefficient, indicating that an increase in the LR value leads to a decrease in the PCC value of the provided dataset.



```
Finished in 37.66405200958252 seconds
```

Ridge

```
Finished in 6209.3434982299805 seconds
```

Ransac

```
Finished in 70.87103223800659 seconds
```

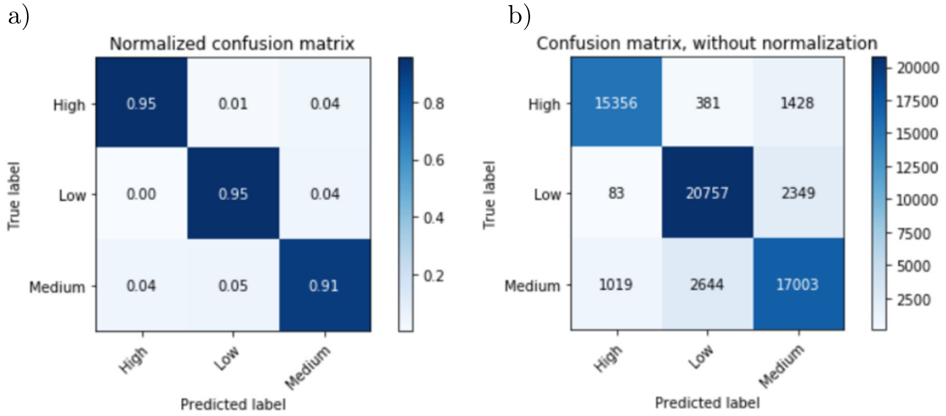
Lasso

```
Finished in 158.8172471523285 seconds
```

KNN

FIG. 11. Comparison of clusters with different RMs for predicting wealth score.

Confusion matrices with and without normalization are once again displayed for LR value of 0.89 and low, medium and high classes in Figs. 12a and 12b.


a)

b)

FIG. 12. Confusion matrices with standardization (a) and without standardization (b).

4.2. Phase 2: daylight images only

In this phase, the model is trained with the help of daylight images only. The locations of the images are identified with assistance of latitude and longitude obtained from the Household and Geographical Survey Data. Table 5 shows the analysis of accuracy and loss with different LR values. It is observed that the training accuracy and test accuracy are highest for lower values. It is also observed that train loss decreases when we decrease LR. The momentum and number of epochs taken here are set to 0.9 and 20, respectively. The results are obtained using the daylight images only.

TABLE 5. Accuracy and loss analysis with various values of LR.

Daylight + nightlight images						
Model	Accuracy and loss					
	Batch size = 64					
Pre-trained	Region	Momentum/Epoch	Learning rate	Training accuracy	Training loss	Test accuracy
True	Delhi/NCR	0.9/20	0.02	0.958	0.294	0.777
	Tamil Nadu	0.9/20	0.02	0.967	0.471	0.841
	Maharashtra	0.9/20	0.02	0.877	0.605	0.866

*4.2.1. **Learning rate: 0.02.*** Figure 13 shows the graph plotting the training and testing loss/accuracy vs. iteration. The training loss degrades from the very 1st iteration and saturates after the 10th iteration.

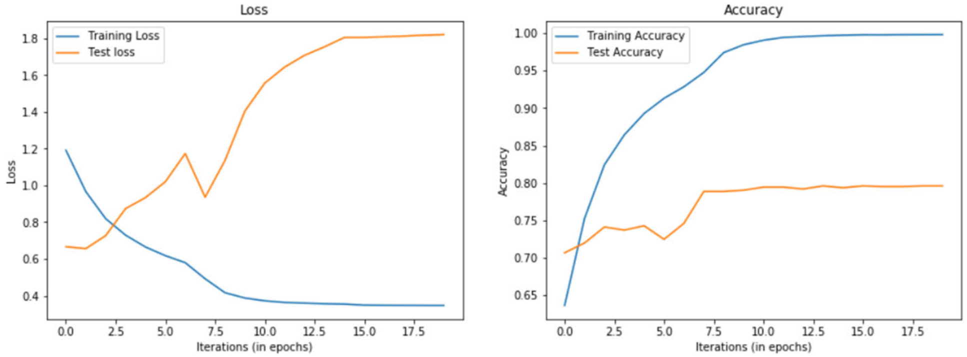
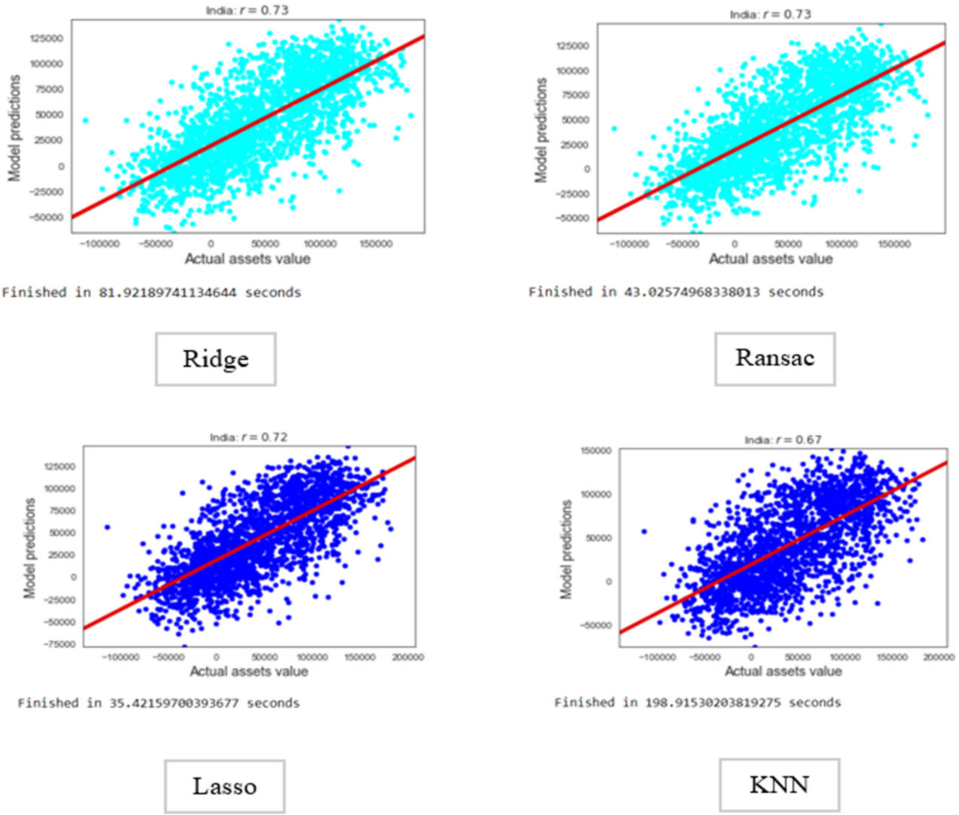


FIG. 13. Iteration vs. loss and accuracy graph.

The testing loss turns sharply at the 6th iteration and again at the 7th iteration. Next, it exponentially increases after the 7th iteration due to the LR decrement by a factor of 10. Meanwhile, the training accuracy exponentially increases from the very first iteration and saturates as LR decreases gradually. The test accuracy shows frequent deflections before the 7th iteration and then stabilizes after the 7th iteration.

Figure 14 shows the performance comparison of four different RMs-Ridge, Lasso, RANSAC and KNN, where the Lasso RM outperforms all the other RMs with a PCC value of 0.72, completing the process in 35.42 sec. This indicates that with LR of 0.02, Lasso RM gives a better output on daylight images.



```
Finished in 81.92189741134644 seconds
```

Ridge

```
Finished in 43.02574968338013 seconds
```

Ransac

```
Finished in 35.42159700393677 seconds
```

Lasso

```
Finished in 198.91530203819275 seconds
```

KNN

FIG. 14. Comparison of clusters with different RMs for predicting wealth score.

Confusion matrices with normalization and without normalization for LR = 0.02 are given in Figs. 15a and 15b.

4.2.2. ***Learning rate: 0.42.*** The training loss starts with a high value from the very 1st iteration and shows frequent deflections until the 6th iteration, then gradually decreases and saturates after the 8th iteration due to changes in LR. In contrast, the test loss starts with a very low value and maintains a single value throughout the iterations. Both training and testing accuracy exponen-

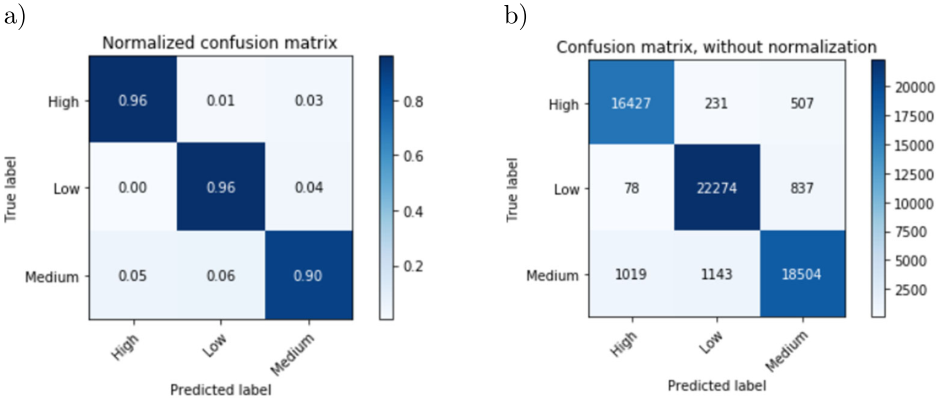


FIG. 15. Confusion matrices with standardization (a) and without standardization (b).

tially increase from the very 1st iteration and converge at the 7th iteration due to a single fixed value of LR, as shown in Fig. 16.

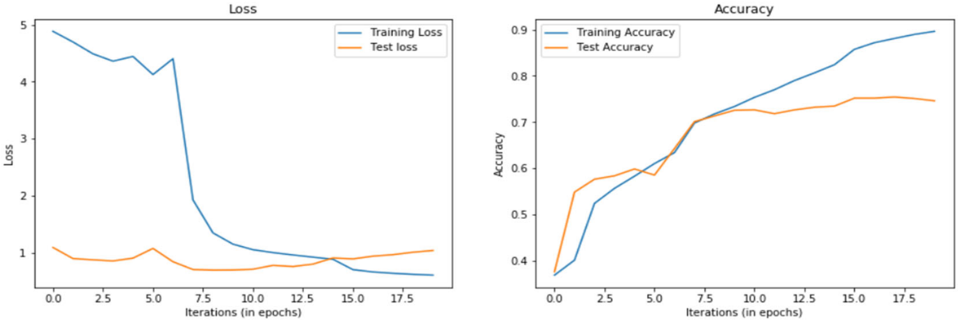
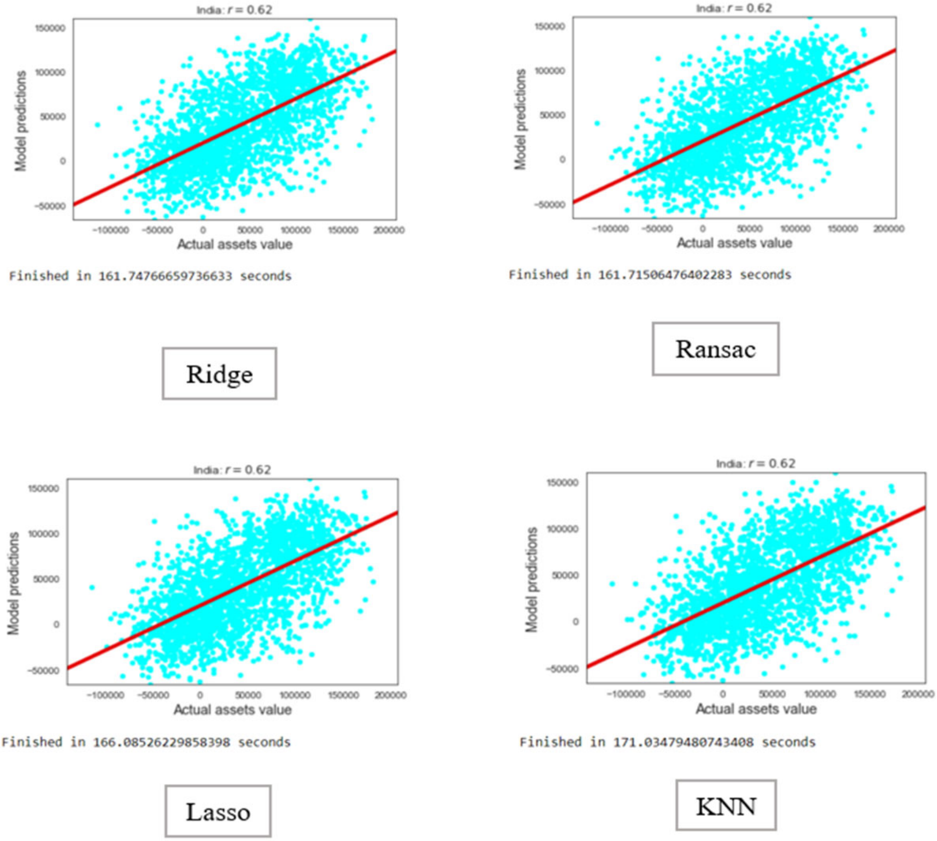


FIG. 16. Iteration vs. loss and accuracy graph.

As shown in Fig. 17, the four RMs are trained on daylight images with LR value = 0.42. Here, with this LR value, the RANSACRM outperforms all three models with an r^2 value of 0.62 and a total processing time of 161.7 sec, which is comparatively faster than the other three models (Ridge, Lasso and KNN).

Confusion matrices with normalization and without normalization for LR = 0.02 are given in Figs. 18a and 18b, respectively.

4.2.3. ***Learning rate: 0.89.*** The training loss, shown in Fig. 19, demonstrates a slight transformation at the 2nd iteration before exponentially falling after the 6th iteration and reaching saturation after the 8th iteration due to the higher LR value. Consecutively, the test loss remains constant throughout all iterations. Both training accuracy and testing accuracy start increasing simultaneously from the 1st iteration, and have the same value from the 3rd iteration



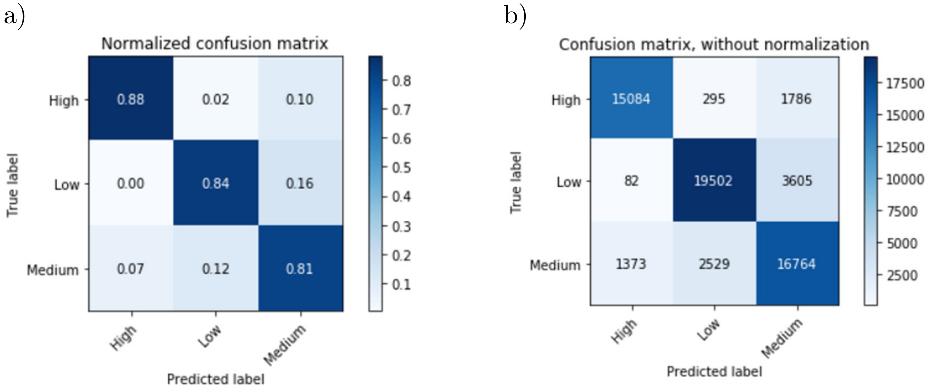
```
Finished in 161.74766659736633 seconds
```

```
Finished in 161.71506476402283 seconds
```

```
Finished in 166.08526229858398 seconds
```

```
Finished in 171.03479480743408 seconds
```

Fig. 17. Comparison of cluster with different RMs for predicting wealth score.



Normalized confusion matrix

True label \ Predicted label	High	Low	Medium
High	0.88	0.02	0.10
Low	0.00	0.84	0.16
Medium	0.07	0.12	0.81

Confusion matrix, without normalization

True label \ Predicted label	High	Low	Medium
High	15084	295	1786
Low	82	19502	3605
Medium	1373	2529	16764

Fig. 18. Confusion matrices with standardization (a) and without standardization (b).

to the 6th iteration, after which training accuracy increases rapidly while testing accuracy saturates from the 6th iteration.

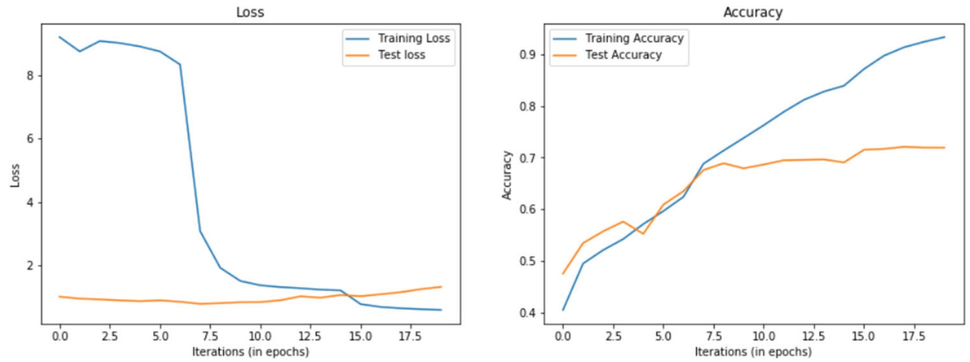
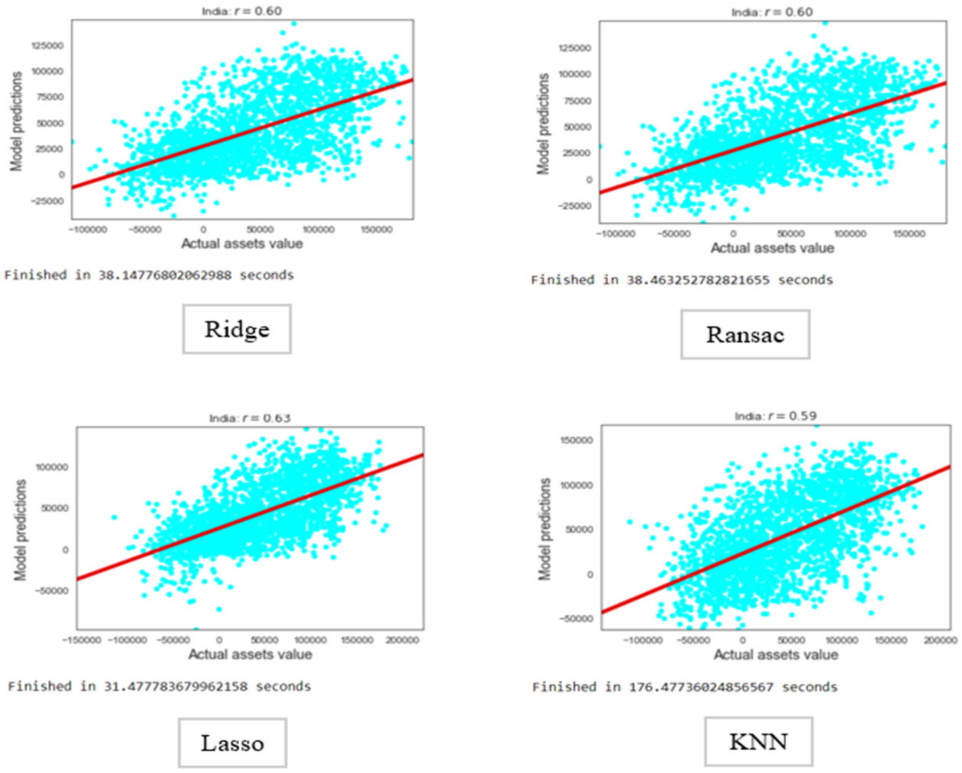


FIG. 19. Iteration vs. loss and accuracy graph.

The four regression models trained on daylight images are shown in Fig. 20, which clearly identifies that Lasso RM with an r^2 value $= 0.63$ is the best regression model for estimating PCC. Although the other TMs perform better at some points, Lasso RM predicts the coefficient with less processing time and fewer coefficient values.



```
Finished in 38.14776802062988 seconds
```

Ridge

```
Finished in 38.463252782821655 seconds
```

Ransac

```
Finished in 31.477783679962158 seconds
```

Lasso

```
Finished in 176.47736024856567 seconds
```

KNN

FIG. 20. Comparison of cluster with different RMs for predicting wealth score.

Confusion matrices with normalization and without normalization for LR = 0.02 are given in Figs. 21a and 21b, respectively.

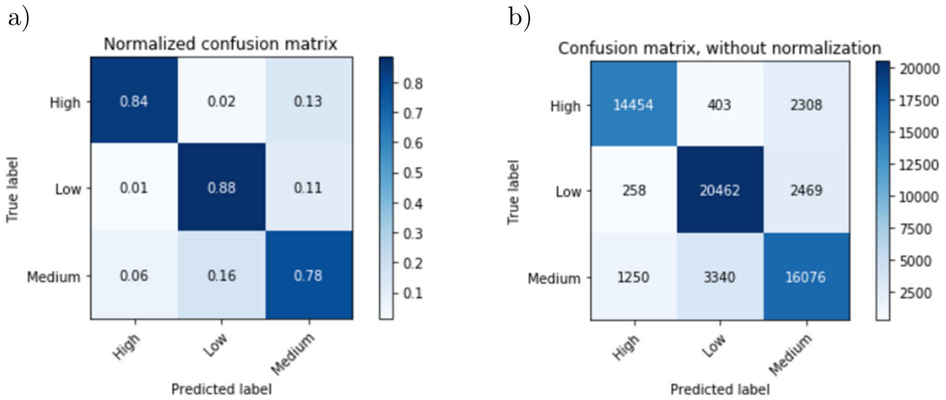


Fig. 21. Confusion matrices with standardization (a) and without standardization (b).

4.3. Phase 3

Table 6 shows the accuracy and loss analysis of training/testing conducted on daylight and nightlight images in the regions of Delhi/NCR, Tamil Nadu, and Maharashtra. The results are obtained using a single value of LR = 0.02, and the test accuracy for Delhi/NCR, Tamil Nadu, Maharashtra, and Telangana is 77.7, 84.1, 86.6 and 66.6, respectively.

Table 6. Accuracy and loss analysis for different regions.

Daylight + nightlight images							
Model	Accuracy and loss						
	Batch size = 64						
Pre-trained	Region	Momentum/ Epoch	Learning rate	Training accuracy	Training loss	Test accuracy	Test loss
True	Delhi/NCR	0.9/20	0.02	0.958	0.294	0.777	1.06
	Tamil Nadu	0.9/20	0.02	0.967	0.471	0.841	1.21
	Maharashtra	0.9/20	0.02	0.877	0.605	0.866	0.675
	Telangna	0.9/20	0.02	0.672	1.01	0.666	0.988

4.3.1. Delhi/NCR region (daylight + nightlight images). In this phase, we focus particularly on predicting poverty in the Delhi/NCR, Tamil Nadu, and Maharashtra regions of India. We filtered the images of these regions with the help of latitude and longitude values and trained the model for prediction with the help of the Lasso RM because it gave the best output with less time and less coefficient value when trained with both daylight + nightlight images, as well as with daylight images alone. Sample images of the Delhi/NCR region are presented in Figs. 22 and 23.

Fig. 22. Sample image 1 [10].

Fig. 23. Sample image 2 [10].

The model is trained with the help of the Inception V3 pre-trained model with LR = 0.02, and momentum and number of epochs are kept at 0.9 and 20, respectively. The test accuracy of the model was determined to be 77.7%. Features extracted from the model were passed on to the and a comparison was made between PCC and DHS data.

Figure 24 presents the graph of training/testing loss and accuracy against iterations, where the training loss decreases exponentially until the 7th iteration before the saturation due to the constant LR value and parameters involved in the pre-training model, By contrast, the test loss shows frequent deflections until the 8th iteration and saturates after the 9th iteration. Training accuracy increases exponentially and saturates after the 3rd iteration, while test accuracy sharply increases until the 2nd iteration and saturates after the 7th iteration.

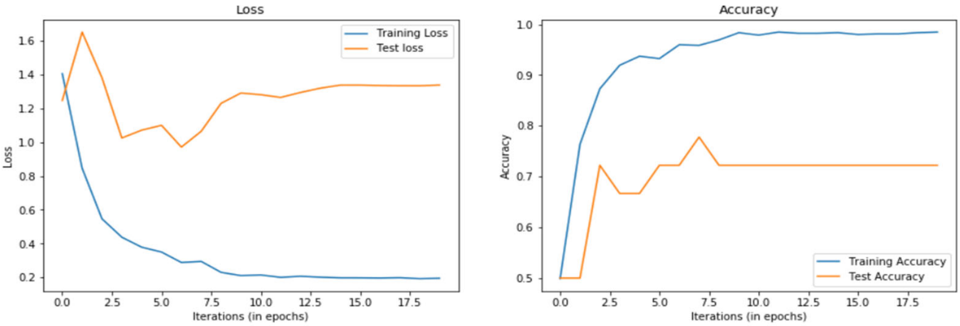
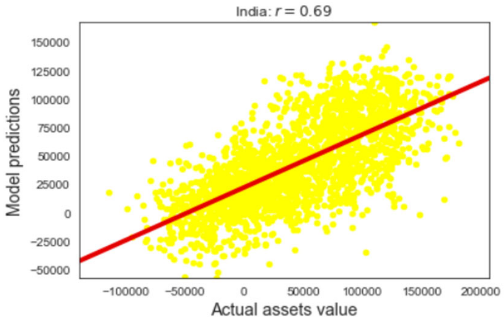


Fig. 24. Iteration vs. loss and accuracy graph.

The trained model is passed to the Lasso RM, which predicts PCC as 0.69 and takes 37.28 seconds. The prediction of the result is fairly accurate, considering that cluster locations of DHS have eruption fused, upholding the invisibility of the survey in Fig. 25.



```
Finished in 37.28893828392029 seconds
```

FIG. 25. Clusters for prediction of wealth score on daylight images.

Confusion matrices for the Delhi/NCR region (daylight + nightlight images) for low, medium and high classes are shown in Figs. 26a and 26b.

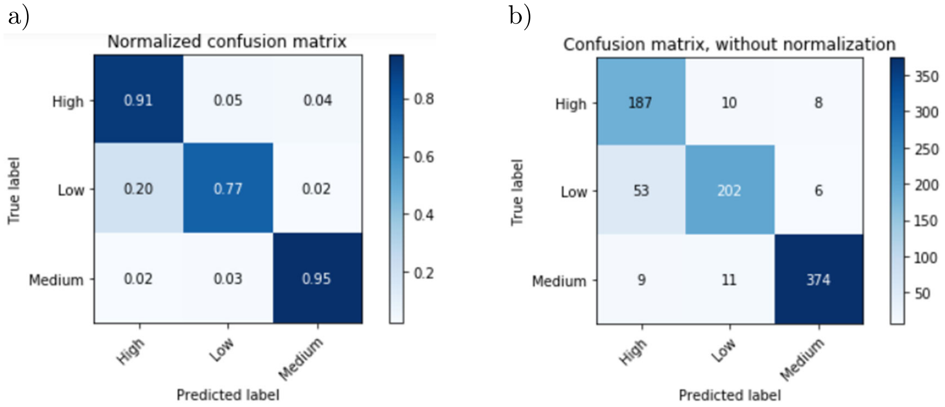


FIG. 26. Confusion matrices with standardization (a) and without standardization (b).

4.3.2. Tamil Nadu region (daylight + nightlight images). Sample images of the Tamil Nadu region are provided in Figs. 27 and 28.

As shown in Fig. 29, both test loss and training loss start with the same magnitude due to a fixed value of LR and momentum value, and the test loss shows frequent and sharp deflections until the 13th iteration before coming to a saturation state. The training loss continuously decreases with each iteration and saturates after the 12th iteration. The test accuracy with initial iteration shows very sharp deflection and then stabilizes during the iterations before finally saturating after the 13th iteration. The training accuracy gradually increases with each iteration.

FIG. 27. Sample image 1 [10].

FIG. 28. Sample image 2 [10].

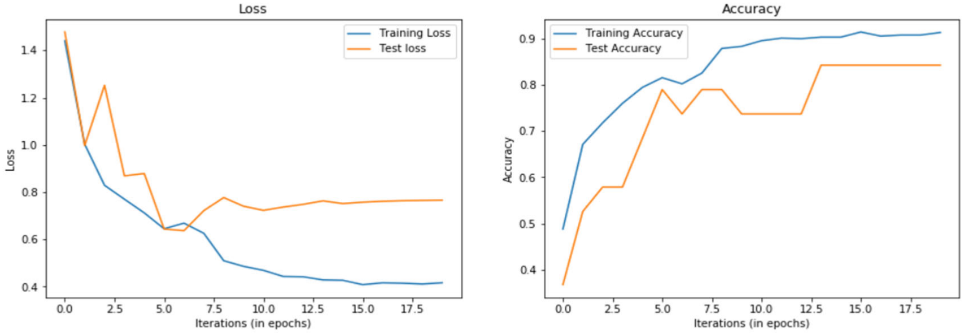
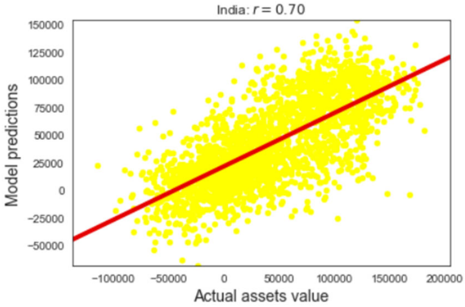


FIG. 29. Iteration vs. loss and accuracy graph.

The Lasso RM predicts PCC as 0.70 and takes 36.10 seconds. The prediction is quite accurate, considering that cluster locations of DHS have eruption fused maintaining the survey's invisibility, as shown in Fig. 30.



```
Finished in 36.107677936553955 seconds
```

FIG. 30. Clusters for prediction of wealth score on daylight images.

Confusion matrices for the Tamil Nadu region (daylight + nightlight images) for low, medium and high classes as shown in Figs. 31a and 31b.

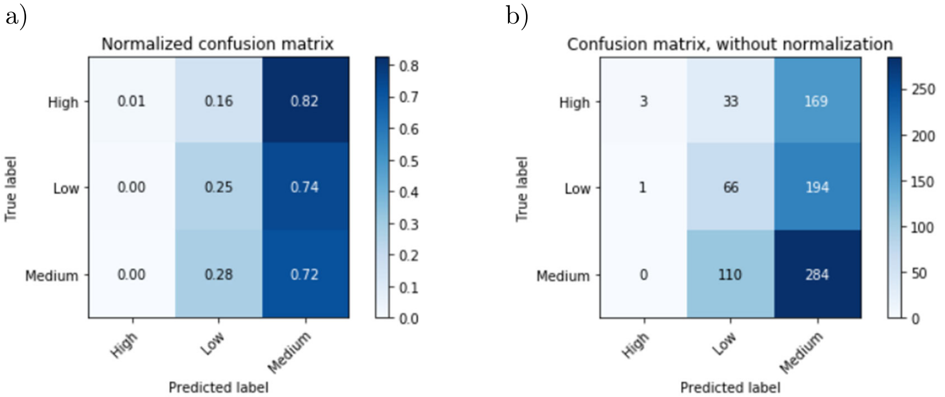


FIG. 31. Confusion matrices with standardization (a) and without standardization (b).

4.3.3. Maharashtra region (daylight + nightlight images). The third module deals with predicting the poverty index in the region of Maharashtra. Sample images of the Maharashtra region are given in Figs. 32 and 33.

FIG. 32. Sample image 1 [10].

FIG. 33. Sample image 2 [10].

As shown in Fig. 34, the test loss starts with a low value and exhibits sharp deflection from the 2nd to the 4th iteration before saturating after the 8th iteration. The training loss starts with a high magnitude and decreases gradually before coming to a saturation state after the 10th iteration. Test accuracy shows sharp variations before the 8th iteration and saturates after the 8th iteration. Training accuracy gradually increases with each iteration and shows no deflection after the 3rd iteration.

The model for the Maharashtra region is trained with Lasso regression, which predicts the value of r^2 as 0.69 and takes 76.10 seconds. The prediction is quite accurate considering that cluster locations of DHS have eruption fused that maintains the invisibility of the survey, as shown in Fig. 35.

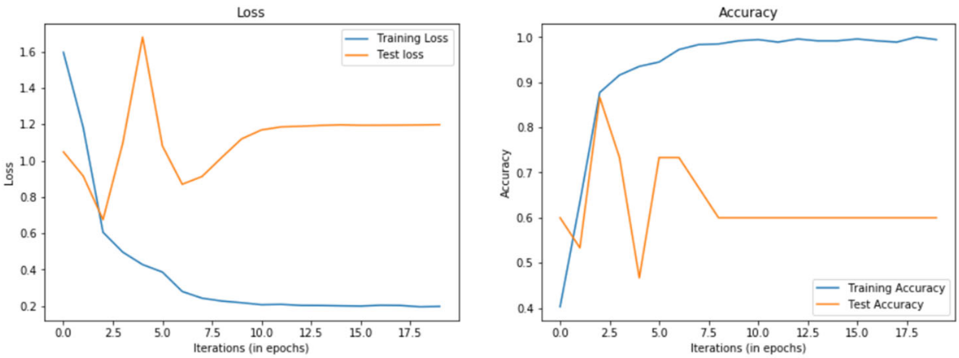
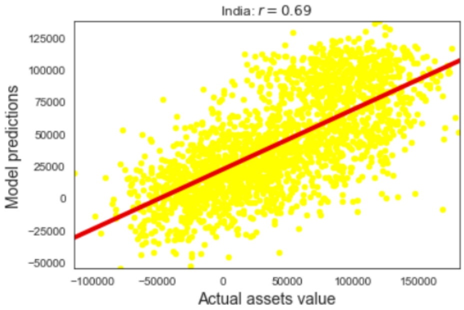


FIG. 34. The graph of iteration vs. loss and accuracy.



```
Finished in 76.10032796859741 seconds
```

FIG. 35. Clusters for prediction of wealth score on daylight images.

Confusion matrices for the Maharashtra region (daylight + nightlight images) for low, medium and high classes are shown in Figs. 36a and 36b.

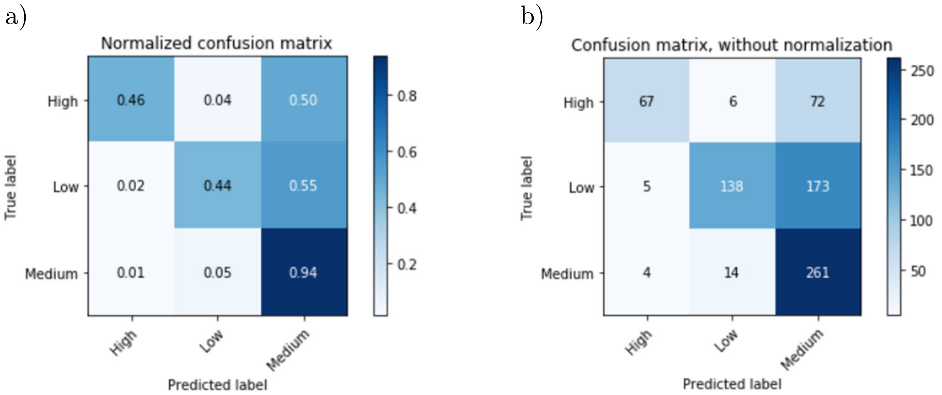


FIG. 36. Confusion matrices with standardization (a) and without standardization (b).

4.3.4. Telangana region (daylight + nightlight images). The sample images for the Telangana region are displayed in Figs. 37 and 38.

FIG. 37. Sample image 1 [10].

FIG. 38. Sample image 2 [10].

In Fig. 39, the training loss shows sharp deflections before the 5th iteration. After the 5th iteration, it starts increasing gradually and saturates after the 15th iteration. The test loss, on the other hand, stars with a very high value and degrades continuously until the 12th iteration before saturating after the 13th iteration. This is due to the very low value of LR and fixed value of momentum.

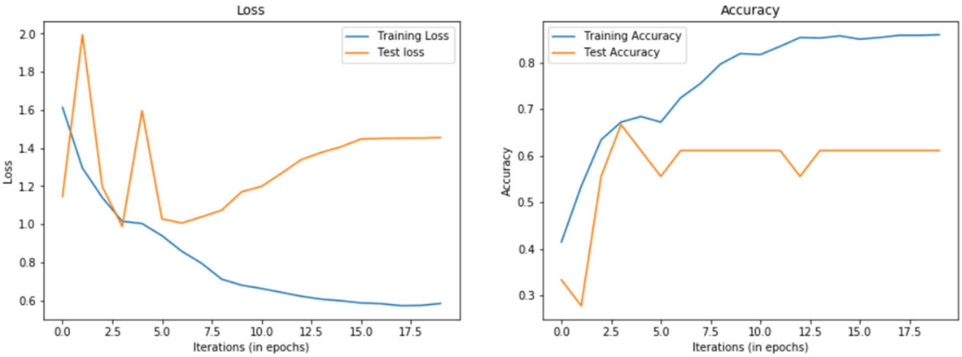
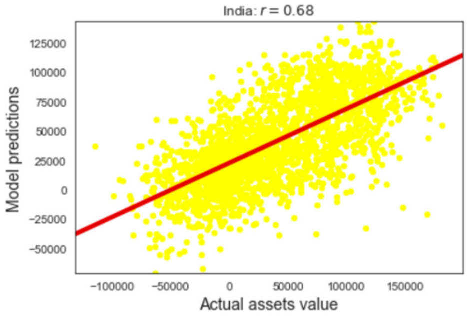


FIG. 39. Iteration vs. loss and accuracy graph.

Training accuracy gradually increases without significant variation in values, and saturates after the 13th iteration. Test accuracy, conversely, initially attains a very high value and decreases before the 5th, reaching a saturation status after the 6th iteration.

The model, trained with Lasso RM, yields an r^2 value of 0.68 and completes in 51.47 seconds. This suggests that the result obtained for the cluster score closely resembles the result obtained from the DHS survey, as shown in Fig. 40.



```
Finished in 51.4705491065979 seconds
```

FIG. 40. Clusters for prediction of wealth score on daylight images.

Confusion matrices for the Telangana Region (daylight + nightlight images) for low, medium and high classes are shown in Figs. 41a and 41b.

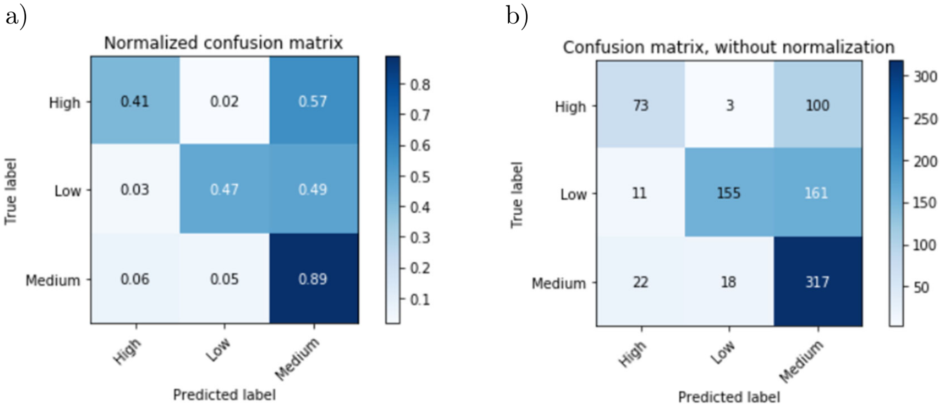


FIG. 41. Confusion matrices with standardization (a) and without standardization (b).

4.4. Phase 4: Comparison of results

As shown in Table 7, we have compared the results obtained from Phases 1 and 2 with respect to training and testing loss, accuracy and the coefficient of determination (R^2). Training accuracy remains the same in both phases and does not show much change in the values. The test loss also does not show much variance in the values. They remain close to 1 in both phases. We observed that the training loss decreases when using daylight images only. The coefficient of determination also improves when using daylight images only. The value of R^2 for daylight + nightlight images is 0.72 whereas for daylight images only, it is 0.73. This shows that using smaller data will improve the model's predictive performance.

Table 7. Comparison of results of different phases.

Comparison of results						
Momentum = 0.9, epoch = 20, batch size = 64						
Phases	Learning rate	Training accuracy	Training loss	Test accuracy	Test loss	R^2 (coefficient of determination)
1) Daylight + nightlight images	0.02	99.1	0.35	80.1	1.7	0.72
	0.42	94.7	0.51	75.3	1.16	
	0.89	88.7	1.16	76.9	1.11	
2) Daylight images only	0.02	99.8	0.34	79.6	1.8	0.73
	0.42	88.09	0.62	75.1	0.96	
	0.89	91.8	0.63	72.9	1.14	

The comparison of computation time is shown in Table 8, where the time taken by different RMs in Phases 1 and 2 is provided. We observed that the computation time for ridge RM does not show a significant difference for LR. However, a notable difference was observed in the case of the RANSAC RM, which required considerably more computation time when trained with both daylight and nightlight images compared to when trained with daylight images only required less time. Additionally, the computation time for the KNN RM shows a substantial difference when trained with LR of 0.02 and not much difference when trained with LR of 0.42 and 0.89. Notably, It was observed that Lasso RM takes shorter times compared to the other three RMs.

Table 8. Comparison of computation time.

Computation time [in seconds]					
Momentum = 0.9, epoch = 20, batch size = 64					
Phases	Learning rate	Ridge regression	Lasso regression	RANSAC regression	KNN regression
1) Daylight + nightlight images	0.02	84.2	35.6	8653.6	571
	0.42	37.3	72.4	6627	168.9
	0.89	37.6	70.8	6209.3	158.8
2) Daylight images only	0.02	81.9	35.4	43	198.9
	0.42	161.7	166	161.7	171
	0.89	38.1	31.4	38.4	176

As shown in Table 9, the results of our paper are compared with those of other papers in the same domain. The locations considered in other papers include India and several countries in Africa. The results obtained in our paper gave better results when compared with the results of several countries in Africa.

Table 9. Comparison of results of different research papers.

Parameter	Paper title	Location (value)
R^2 (coefficient of determination)	Multi-task DL for predicting poverty from satellite images [13]	India (0.82)
	Temporal poverty prediction in developing countries [16]	Africa: Malawi (0.55), Nigeria (0.68), Tanzania (0.57), Uganda (0.69)
	Combining satellite imagery and ML to predict poverty [3]	Rwanda (0.75)
	Socioecologically informed use of remote sensing data to predict the rural household poverty [20]	Sauri (Western Kenya) (0.62)
	Predicting city poverty using satellite imagery [22]	Chile: Santiago (0.52) USA: Los Angeles (0.56), Chicago (0.36), Boston (0.37), Houston (0.327), Philadelphia (0.46)
	Mapping poverty using mobile phones and satellite data [23]	Bangladesh (0.78)
	Identifying a slum degree of deprivation from VHR images using CNN [9]	India: Banglore (0.75)
	Predicting wealth score from remote sensing satellite images and household survey data using deep learning – current study	India (0.72) (daylight + nightlight) India (0.73) (daylight images) Delhi/NCR (0.69), Tamil Nadu (0.70), Maharashtra (0.69), Telangana (0.68)

Additionally, we provide a comprehensive comparison for both the entire area of India and four specific states – Delhi/NCR, Tamil Nadu, Maharashtra and Telangana.

5. Conclusion

We have exhaustively reviewed the literature to understand how ML techniques can help predict the socioeconomic parameter (poverty index) using remote sensing satellite images, household and geographical survey data. This review led us to analyze the use of satellite images and pre-trained models to predict poverty levels. We established a strong correlation between the results obtained from satellite images and DHS Data. In Phase 1, we considered both daylight and nightlight images. The poverty index was determined to be 0.72 using the Lasso RM. In Phase 2, we predicted the labels of the daylight images using survey data. Fine-tuning of the CNN improved the extraction of features, and the poverty index increased to 0.73. In both phases, Lasso RM outperformed

Ridge, RANSAC and KNN RMs. Considering the results obtained in Phases 1 and 2 using Lasso RM, we trained the model again in Phase 3 with images only from the Delhi/NCR, Tamil Nadu, Maharashtra and Telangana regions of India. The calculated wealth scores for these regions were 0.69, 0.70, 0.69, and 0.68, respectively. Phase 4 compared these results with existing data. Future prospects that can be considered include:

- Using the Google Maps API instead of the HERE Maps API to download the daylight images;
- Using DHS Data for 2019-20 for more refined results.

Acknowledgements

We would like to thank HERE Maps for providing the dataset for daylight satellite images and for their guidance on how to access these images for training of our model. We are also grateful to the DHS Program and the DMSP-OLS data source for granting us permission to use the DHS Survey data for 2015–2016 and nightlight images, respectively.

References

1. World Data Lab, World Poverty Clock, https://worlddata.io/portfolio/world-poverty-clock/.
2. D. Eugenie, R. Kennedy, J. Urpelainen, Satellite data for the social sciences: measuring rural electrification with night-time lights, *International Journal of Remote Sensing*, **39**(9): 2690–2701, 2018, doi: 10.1080/01431161.2017.1420936.
3. J. Neal, M. Burke, M. Xie, W.M. Davis, D.B. Lobell, S. Ermon, Combining satellite imagery and machine learning to predict poverty, *Science*, **353**(6301): 790–794, 2016, doi: 10.1126/science.aaf789.
4. NOAA, Nightlight Images, DMSP-OLS Dataset, https://ngdc.noaa.gov/eog/dmsp/.
5. The DHS Program, Available datasets, https://dhsprogram.com/data/available-data sets.cfm.
6. K.N. Nischal, R. Radhakrishnan, S. Mehta, S. Chandani, Correlating night-time satellite images with poverty and other census data of India and estimating future trends, [in:] *CODS'15: Proceedings of the 2nd ACM IKDD Conference on Data Sciences*, New York, NY, United States, pp. 75–79, Association for Computing Machinery (ACM), 2015, doi: 10.1145/2732587.2732597.
7. United Nations, A World That Counts: Mobilizing the Data Revolution for Sustainable Development – Report of the Secretary-General's Independent Expert Advisory Group on the Data Revolution for Sustainable Development, 2014, https://digitallibrary.un.org/record/3882725.
8. United Nations, Sustainable Development Goals: 17 Goals to Transform Our World, 2015, http://www.un.org/sustainabledevelopment/sustainable-development-goals/.

9. A. Ajami, M. Kuffer, C. Persello, K. Pfeffer, Identifying a slums' degree of deprivation from VHR images using convolutional neural networks, *Remote Sensing*, **11**(11): 1282, 2019, doi: 10.3390/rs11111282.

10. Daylight Images, Google HERE Maps API, www.here.com.

11. Living Standard Measurement Survey, The World Bank, www.worldbank.org/lsms.

12. E. Sheehan, C. Meng, M. Tan, B. Uzkent, N. Jean, M. Burke, D. Lobell, S. Ermon, Predicting economic development using geolocated Wikipedia articles, [in:] *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2698–2706, Association for Computing Machinery, New York, NY, United States, 2019, doi: 10.1145/3292500.3330784.

13. S. Pandey, T. Agarwal, N.C. Krishnan, Multi-task deep learning for predicting poverty from satellite images, [in:] *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32(1), pp. 7793–7798, New Orleans, Lousiana, USA, 2018, doi: 10.1609/aaai.v32i1.11416.

14. L. Duan, T. Hu, E. Cheng, J. Zhu, C. Gao, Deep convolutional neural networks for spatiotemporal crime prediction, [in:] *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, H.R. Arabnia [Ed.], pp. 61–67, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2017.

15. L. Guie, Z. Cai, X. Liu, J. Liu, S. Shiliang, A comparison of machine learning approaches for identifying high-poverty counties: Robust features of DMSP/OLS night-time light imagery, *International Journal of Remote Sensing*, **40**(15): 5716–5736, 2019, doi: 10.1080/01431161.2019.1580820.

16. R. Ruchir, Temporal poverty prediction in developing countries, *Stanford CS229: Machine Learning*, Stanford University, California, 2017, https://cs229.stanford.edu/proj2017/final-reports/5244050.pdf.

17. S.P. Subash, R.R. Kumar, K.S. Aditya, Satellite data and machine learning tools for predicting poverty in rural India, *Agricultural Economics Research Review*, **31**(2): 231–240, 2018, doi: 10.5958/0974-0279.2018.00040.X.

18. B. Raj, A simple guide to the versions of the inception network, *Towards Data Science*, May 29, 2018, https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202 [accessed: Aug. 20, 2020].

19. M. Xie, N. Jean, M. Burke, D. Lobell, S. Ermon, Transfer learning from deep features for remote sensing and poverty mapping, [in:] *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30(1), pp. 3929–3935, AAAI Press, Palo Alto, California USA, 2016, doi: 10.1609/aaai.v30i1.9906.

20. G.R. Watmough, C.L.J. Marcinko, C. Sullivan, K. Tschirhart, P.K. Mutuo, C.A. Palm, J.C. Svenning, Socioecologically informed use of remote sensing data to predict rural household poverty, [in:] *Proceedings of the National Academy of Sciences*, Vol. 116(4), pp. 1213–1218, 2019, doi: 10.1073/pnas.1812969116.

21. A. Perez, S. Ganguli, S. Ermon, G. Azzari, M. Burke, D. Lobell, Semi-supervised multitask learning on multispectral satellite images using Wasserstein generative adversarial networks (GANs) for predicting poverty, *arXiv*, 2019, arXiv: 1902.11110.

22. S. Piagessi *et al.*, Predicting City Poverty Using Satellite Imagery, [in:] *Proceedings of the IEEE/CFV Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 90–96, Long Beach, California, USA, 2019.

23. J.E. Steele *et al.*, Mapping poverty using mobile phone and satellite data, *Journal of the Royal Society Interface*, **14**(127): 20160690, 2017, doi: 10.1098/rsif.2016.0690.

24. G. Catamuro, A. Muhebwa, J. Taneja, Street smarts: measuring intercity road quality using deep learning on satellite imagery, [in:] *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies, Association for Computing Machinery*, pp. 145–154, New York, NY, United States, 2019, doi: 10.1145/3314344.3332493.

25. P.K. Suraj, A. Gupta, M. Sharma, S.B. Paul, S. Banerjee, On monitoring development indicators using high resolution satellite images, *arXiv*, 2017, arXiv: 1712.02282.

26. ImageNet, Database, http://www.image-net.org.

27. S. Bhattacharyya, Ridge and Lasso regression: L1 and L2 regularization, *Towards Data Science*, 2018, https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.

28. RANSAC Regression, Regression Models, 2018, https://scikit-learn.org/stable/modules/classes.html.

29. KNN Nearest Neighbors Regression, 2018, https://scikit-learn.org/stable/supervised_learning.html#supervised-learning.

30. Pearson's Correlation Coefficient, Correlation Coefficient, 2018, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

31. E. Sheehan, Z. Nabulsi, C. Meng, *Utilizing Latent Embeddings of Wikipedia Articles to Predict Poverty*, Preprint, Stanford University, 2018, https://cs229.stanford.edu/proj2018/report/134.pdf [accessed: Aug. 20, 2020].

Received July 16, 2022; accepted August 21, 2022;
published online June 24, 2024.