# Predicting Child Mortality Rate from Satellite Imagery Using CNNs

**Raaisa Moktader**
Department of Computer Science
rmoktad@stanford.edu

**Shayana Venukanthan**
Department of Computer Science
shayanav@stanford.edu

**Tim Wu**
Department of Computer Science
timwu0@stanford.edu

## Abstract

Progress towards the United Nations' Sustainable Development Goals (SDGs) has been hindered by a lack of data on key environmental and socioeconomic indicators, but recent advances in machine learning have made it possible to utilize abundant and frequently-updated data from satellites to provide insights. We propose a Convolutional Neural Network (CNN) to predict the child mortality rate using satellite imagery. We use the SustainBench dataset, which pulls from the Demographic and Health Surveys (DHS) from 1996-2019. Starting with a ResNet architecture and applying transfer learning, we are able to outperform the baseline model proposed by SustainBench, demonstrating the feasibilty of using deep learning frameworks to estimate child mortality rates from satellite data. All code is publicly available at this GitHub repository.

## 1 Introduction

Progress towards the United Nations' Sustainable Development Goals (SDGs) [1] has been hindered by a lack of data on key environmental and socioeconomic indicators, but recent advances in machine learning have made it possible to utilize abundant, frequently-updated, and globally available data from satellites to provide insights into progress toward SDGs. However, these approaches thus far have largely been evaluated on different datasets or used inconsistent evaluation metrics, making it hard to understand performance.

We propose a model for predicting indicators of the progress towards the United Nations' Sustainable Development Goals (SDGs) through satellite imagery. Our focus for this project will be to predict the child mortality rate for a region given the corresponding satellite imagery, as there is evidence that child mortality is "connected to environmental factors such as housing quality, slum-like conditions, and neighborhood levels of vegetation" [5]. With this insight, we hope to facilitate gauging progress towards SDGs especially in remote, less accessible locations.

The input to our algorithm is a 255 x 255 x 3px satellite image. We then use a CNN to output a predicted value for the number of deaths per 1,000 children, as our child mortality rate predictions are grouped into buckets as a classification problem.

**JOINT PROJECT DETAILS:** This project was completed as a joint project for both CS 230 and CS231N. We received permission from both teaching teams to use the same code base and results for both projects. All code and model architectures were developed for both classes.

CS231N: Deep Learning for Computer Vision, Spring 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

## 2 Related work

Our task applies CNNs to a regression problem which takes as input satellite imagery and must output a numerical value using the SustainBench dataset [11]. As such, it is important that we mention prior work on SustainBench, image classification with CNNs, satellite imagery classification, and final regression layers.

**SustainBench** The SustainBench paper[11] introduces SustainBench, a collection of 15 benchmark tasks across 7 SDGs, including child mortality rate, and includes publicly released datasets for 11 of the 15 tasks. The baseline for the child mortality rate task is a $k$-Nearest-Neighbors (KNN) model that inputs the average pixel value for the the nightlights band. We adopt this baseline but acknowledge its great weakness as it relies solely on a single average pixel value from the least expressive band (see last band in Table 1), which fails to adequately represent an 8x255x255 satellite image.

**Novel CNNs** One approach used for image classification problems such as ours is to hand-construct a CNN architecture. Sun, et. al. [8] found that a VGG-inspired simple CNN greatly outperformed the more complex pretrained state-of-the-art CNNs on multi-label classification of Amazon satellite imagery. Inspired by their success, we constructed our own simple CNN for our second baseline.

**State-of-the-Art CNNs** The most common approach to image classification problems is to leverage existing state-of-the-art CNN architectures, such as VGGNet [7] or the more-complex ResNet [4]. While both of these models are for single label image classification, they're easily generalizable to other tasks by unfreezing layers and making minor modifications.

In their work classifying snow using multispectral satellite imagery, Xia et. al.[9] apply the multidimensional deep residual network (M-ResNet). Sun. et. al. [8] use pretrained VGGNet, Inception, and ResNet to classify rainforest satellite imagery. We apply transfer learning with a pretrained ResNet [4] modified for our regression problem like Xie et. al.'s [10] design of a final regression layer.

## 3 Dataset and Features

We use the SustainBench dataset [11], which contains benchmark datasets for several SDG-related tasks from the Demographic and Health Surveys (DHS) from 1996 to 2019 for 56 different countries, including child mortality. These satellite images contain eight bands. The first seven bands of the satellite image are surface reflectance values from the Landsat 5/7/8 satellites and have the following order: blue, green, red, shortwave infrared 1, shortwave infrared 2, thermal, and near infrared. The last band in the satellite image is the nightlights band, from either the DMSP or VIIRS satellite. To pre-process our data, we investigated the distribution of our data using Google Colab.

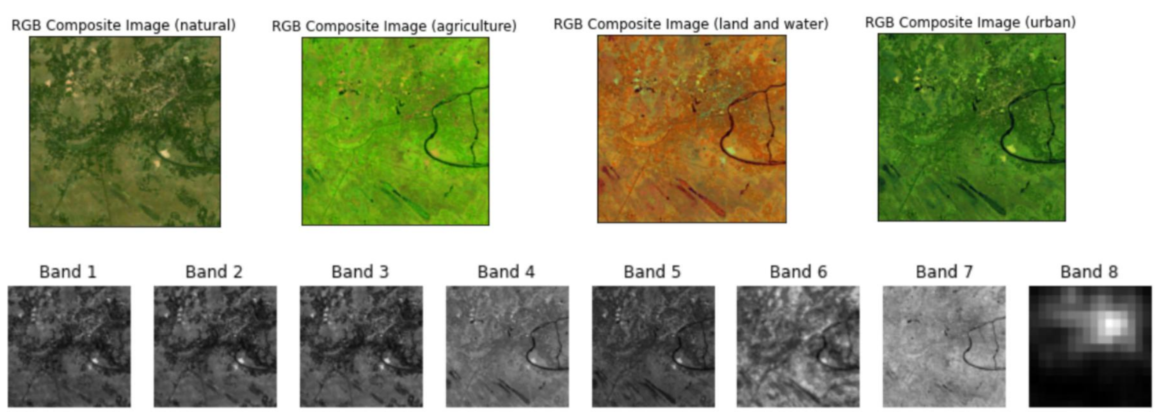



**Table 1.** RGB composites with different band combinations and grayscale images of the eight individual bands of the LandSat 5/7/8 for a sample image.

Some entries in the raw SustainBench dataset contained `NaNs` for the child mortality rate, so we discarded these data points since our data set is quite large, and these images would impede the performance of our model without accurate child mortality rates labeled. The following analysis of our data refers to the dataset where data points with `NaNs` for child mortality are cleaned out (105,582 total data points).

Our data contains almost double the satellite images from regions labeled as rural than urban, which fits our purpose since our goal is to help determine child mortality rates at less accessible sites, which often tend to be more rural. The labels for child mortality in our data ranges from $5.0$ to $166.0$, with a mean of $18.335$ and standard deviation of $12.160$. All labels are whole numbers stored as floats for the $105,582$ total data points used.

Following the example of SustainBench [11], we use a uniform train/validation/test data split by country. Delineating by country ensures that there is no overlap between any of the splits (i.e., a model trained on our train split will not have "seen" any part of any image from the test split). See appendix for the specific countries in each split.

| | Train | Validation | Test |
|---|---|---|---|
| # of Countries | 30 | 13 | 13 |
| child mortality rate | 69,052 (65%) | 17,062(16%) | 19,468 (18%) |

We also normalized our data before running it through our model. We employed the default mean and standard deviation values calculated from the ImageNet dataset [3], based on millions of images from this database, so that gradient descent converges faster.

# 4 Methods

In order to develop a more accurate child mortality prediction algorithm than those used in previous works, we employ Convolutional Neural Networks (CNNs) to learn the input-output mappings between the child mortality rate at a site and the values of the various LandSat 5/7/8 bands captured by the satellite. For this project, we use transfer learning with residual neural networks (ResNet), which uses residual blocks and skip connections to allow for larger, more sophisticated models to be built without running into the issues of over-fitting or vanishing gradients. We modify ResNet to our task by adding a Fully Connected (FC) Layer to output 167 scores and take the weighted average for our final prediction. Our implementation was done using the widely-used deep learning framework Pytorch [6].

Using CNNs over the KNN model used in previous approaches offers the advantage of learning visual patterns/features, such as lines, boundaries, and textures, as opposed to taking the average value of pixels (as done with the SustainBench KNN model [11]).The architecture of the CNN as a sequence of layers allows each layer to use information learned in the previous layer to learn more complicated input-output relationships than in a KNN model, which simply predicts based on the closest (i.e., the closest average pixel value for the eight band) image in the training set. We hypothesize that these characteristics of CNNs would make them more suitable for child mortality rates than KNNs. We employ the mean-absolute-error (MAE) loss function to train CNNs in our experiments. This loss function is computed as the sum of the absolute differences between predicted outputs of the CNN $\hat{y_i}$ and the ground truth $y_i$ across a batch of training examples. The equation for MAE is as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y_i} - y_i|.$$

This loss increases as the average difference between the prediction and ground truth increases. A learning algorithm training a model with the MAE loss function would tune the model parameters to produce predicted outputs as close as possible to the expected outputs or ground truth for a given input satellite image.

To evaluate the success of our model, we have used the Pearson's $r^2$ coefficient of determination and the prediction accuracy. We use the $r^2$ coefficient in order to be consistent with the benchmark model provided by the SustainBench dataset [11]. The equation for the $r^2$ coefficient is as follows:

$$r^2 = \left(\frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}\right)^2.$$

While we understand that ultimately this is a regression problem rather than a classification one and to this extent the prediction accuracy seems like a strange metric, it nonetheless provides a useful guideline for the precision of our models.

To train our CNN models, we employ the mini-batch gradient descent algorithm with the Adam optimization algorithm. The Adam optimization algorithm is a combination of two other optimization algorithms: momentum and RMSProp. Momentum applies an exponentially-weighted average of the gradient across the last couple iterations to each model parameter. This reduces the noise and variance in the updates and allows the learning algorithm to avoid getting stuck at saddle points in the loss function landscape. RMSProp customizes the learning rate used to update each model parameter in order to help the model parameters converge more quickly. Please see the appendix for Adam optimizer's equations.

Mini-batch gradient descent involves applying the average derivative of a model parameter with respect to the loss function of multiple training examples rather than just one. This greatly accelerates training on GPUs.

## 5 Experiments/Results/Discussion

**Hyperparameter Tuning** For our model, we use default first and second moment parameters for the Adam optimizer ($\beta_1$=0.9, $\beta_2$=0.999) as these are the default values used in many deep learning frameworks. After some preliminary experimenting we found that batch size did not significantly affect model performance and thus decided to stick with a standard batch size of 64 for our model. The hyperparameters we decided to tune, in order, were: learning rate, $L^2$ regularization, ResNet model type, number of frozen ResNet layers, and which satellite bands were used. For this final satellite bands hyprparameter, we followed Kevin Bulter's article [2] in interpreting our different sets of bands. We decided to test out band combinations representing RGB, Agriculture, Land/Water, and Urban visualizations as these categories seemed most sensible for affecting child mortality rate.

For each hyperparameter, we found the optimal value and then carried this optimal value throughout the rest of our experiments. We selected each hyperparameter based on which value produced the highest $r^2$ coefficient on the validation set. We choose hyperparameters to optimize for $r^2$ coefficent as this is the metric that will be compared against the SustainBench's benchmark model. Below is a table of all our hyperparameter tuning trials with optimal values in boldface.

| Learning rate (1 epoch) | |
|---|---|
| lr | Val $r^2$ |
| 1e-6 | 0.000 |
| 1e-5 | 0.000 |
| 1e-4 | 0.116 |
| **1e-3** | 0.149 |
| 1e-2 | 0.044 |

| $L^2$ Regularization (10 epochs) | |
|---|---|
| Weight Decay | Val $r^2$ |
| 0 | 0.1748 |
| 1e-6 | 0.1715 |
| 1e-5 | 0.1752 |
| 1e-4 | 0.1695 |
| **1e-3** | 0.1771 |

From these, we see that while learning rate was very important, $L^2$ regularization had a less significant impact. Trying higher $L^2$ regularizations showed poor initial performance so we decided to set a weight decay of $0.001$ and move forward with tuning.

| ResNet Model (5 epochs) | |
|---|---|
| Model Name | Val $r^2$ |
| **ResNet-18** | 0.1752 |
| ResNet-34 | 0.1091 |
| ResNet-50* | - |

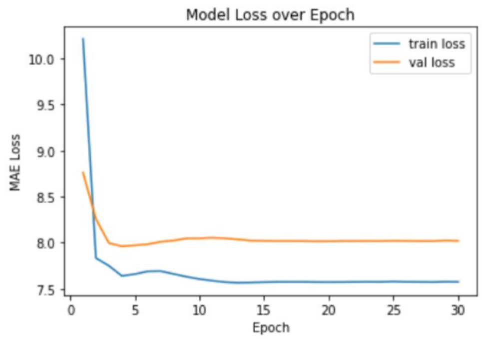
| # Frozen Layers (10 epochs) | |
|---|---|
| Frozen Layers | Val $r^2$ |
| 0 | 0.1771 |
| 3 | 0.1785 |
| **6** | 0.1790 |
| 9 | 0.0099 |

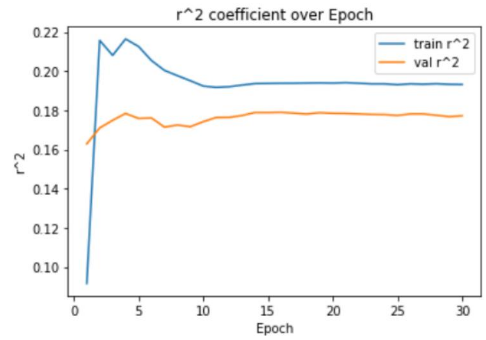
| Landsat8 Bands (10 epochs) | |
|---|---|
| Category | Val $r^2$ |
| **RGB** | 0.1790 |
| Agriculture** | 0.1030 |
| Land/Water | 0.1636 |
| Urban** | 0.0315 |

*ResNet-50 gave very poor performance after 1 epoch and so we stopped training early.
*When using agriculture and urban bands, we stopped the model and recorded results after 5 epochs as we saw poor comparative performance.

Overall, we see that the best model was ResNet-18 when freezing 6 layers, using the RGB channels of the input images, with a learning rate of 0.001 and Adam optimizer weight decay of $0.001$.

**Model Comparisons** Using the above hyperparameters, we trained our model over 30 epochs. We see that the model's train and val loss quickly decrease initially, slightly rise somewhere around epochs 5-10, and then continue to decrease before stabilizing around epoch 15. These patters are also reflected in the $r^2$ coefficients.

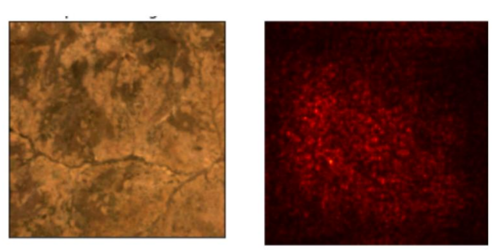



After running on the test set, we found a significant drop in $r^2$ performance. While our best model had a val $r^2$ of $0.1790$, it achieved a test $r^2$ of $0.0922$. We believe this may be due to the design of the SustainBench dataset; the geography and wealth of the countries in the train and val set may be more similar while the countries in the test set may be more different. Despite the poorer test $r^2$ coefficient, this still beats the benchmark and is at present the best-performing model at the given task.

| Models Comparison | | |
|---|---|---|
| Category | Val $r^2$ | Test $r^2$ |
| SustainBench kNN (Yeh et. al.) | 0.0395 | 0.0700 |
| CNN Baseline (Milestone) | 0.0109 | 0.0052 |
| **ResNet-18 + FC-167 (Final model)** | 0.1790 | 0.0922 |

**Saliency Map** Based on the following saliency map example, we see that the model fails to truly capture the land's features such as rivers or vegetation, which explains its large room for improvement.



## 6 Conclusion/Future Work

The best performing model was the ResNet-18 when freezing 6 layers, using the RGB channels of the input images, with a learning rate of 0.001 and Adam optimizer weight decay of 0.001.

With more time and resources, our team would have loved to explore different architectures, including VGGNet [7] and further explore deeper neural networks (including ResNet-34, ResNet-50) that could learn even more complex mappings. While we ran some tests with these models, we did not have the chance to run more detailed finetuning to optimize these models. We would also experiment with different band combinations from the satellite imagery in groups of three beyond the natural RGB bands, including healthy vegetation (bands 7, 3, and 1), as suggested by for Landsat 8 satellite images [2]. A natural next step for our project would be to create an ensemble-method model of ResNets across different band combinations where the model prediction is a linear combination of individual band combination predictions with learnable weight parameters.

## 7 Contributions

All team members–Raaisa Moktader, Shayana Venukanthan, and Tim Wu–contributed equally. Raaisa Moktader prioritized data preprocessing, visualization, and loading. Tim Wu prioritized running the model and hyperparameter tuning. Shayana Venukanthan prioritized model understanding with saliency maps and evaluation metrics.

## References


[1] Transforming our world: The 2030 agenda for sustainable development. `https://sustainabledevelopment.un.org/post2015/transformingourworld/publication`, 2015.

[2] Kevin Butler. Band combinations for landsat 8. In *Band Combinations for Landsat 8*. ArcGIS, 2013.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database, 2009.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] M. M. Jankowska, M. Benza, and J. R. Weeks. Estimating spatial inequalities of urban child mortality. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3903295`, 2013.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[8] Robert Sun, Christian Castellanos, and Andrew Nguyen. Deep multi-label classification for high resolution satellite imagery of rainforests, 2017.

[9] Min Xia, Wan'an Liu, Bicheng Shi, Liguo Weng, and Jia Liu. Cloud/snow recognition for multispectral satellite imagery based on a multidimensional deep residual network. *International Journal of Remote Sensing*, 40(1):156–170, 2019.

[10] Yuanpu Xie, Fuyong Xing, Xiangfei Kong, Hai Su, and Lin Yang. Beyond classification: Structured regression for robust cell detection using convolutional neural network. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 358–365, Cham, 2015. Springer International Publishing.

[11] Christopher Yeh, Chenlin Meng, Sherrie Wang, Anne Driscoll, Erik Rozi, Patrick Liu, Jihyeon Lee, Marshall Burke, David B. Lobell, and Stefano Ermon. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.


## 8 Appendix

```
count    105582.000000
mean         18.345021
std          12.160344
min           5.000000
25%          10.000000
50%          15.000000
75%          23.000000
max         166.000000
Name: n_under5_mort, dtype: float64
```

**Figure 1.** Overall data statistics.

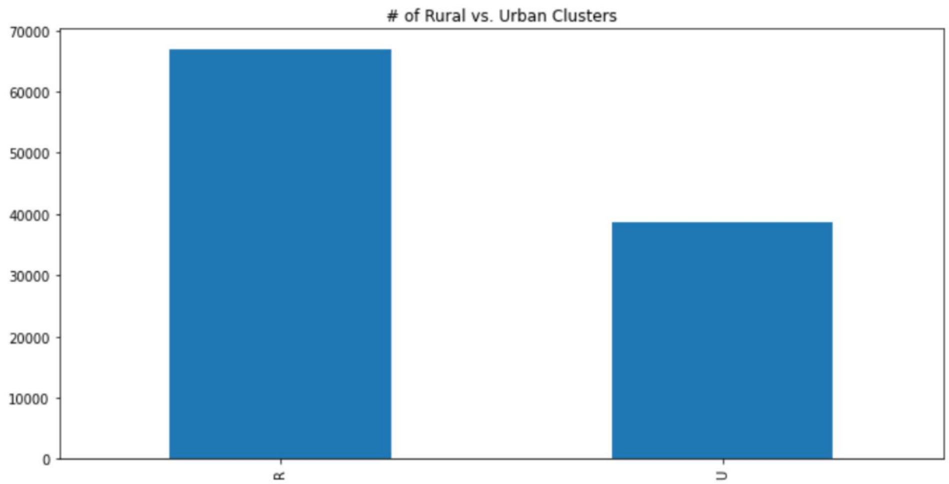



**Figure 2.** Number of Rural vs. Urban satellite images.

| | Train | Validation | Test |
|---|---|---|---|
| DHS Country Codes | 30 countries: AL, BD, CD, CM, GH, GU, HN, IA, ID, JO, KE, KM, LB, LS, MA, MB, MD, MM, MW, MZ, NG, NI, PE, PH, SN, TG, TJ, UG, ZM, ZW | 13 countries: BF, BJ, BO, CO, DR, GA, GN, GY, HT, NM, SL, TD, TZ | 13 countries: AM, AO, BU, CI, EG, ET, KH, KY, ML, NP, PK, RW, SZ |
| child mortality rate | 69,052 (65%) | 17,062 (16%) | 19,468 (18%) |

**Table 3.** Data splits used for training, validation, and testing.

**Adam Optimizer equations:**

$$W := W - \alpha \frac{V_{dW}^{corrected}}{\sqrt{s_{dW}^{corrected}} + \epsilon}$$

$$b := b - \alpha \frac{V_{db}^{corrected}}{\sqrt{s_{db}^{corrected}} + \epsilon}$$