Nighttime Light Predictions from Satellite Imagery

Neal Jean, Rachel Luo
Department of Electrical Engineering
Stanford University
{nealjean, rsluo}@stanford.edu

Jae Hyun Kim
Department of Computer Science
Stanford University
jakekim@stanford.edu

Abstract

The lack of reliable poverty data in developing countries poses a major challenge for making informed policy decisions and allocating resources effectively in those areas of the world. However, one good indicator of both economic development and population density is nighttime light intensity. In this paper, we present a method of predicting the nighttime light intensity distribution of a region given the corresponding daytime satellite image. We fine-tune a Convolutional Neural Network (CNN) pre-trained on ImageNet, and then test the classifier’s performance in predicting values of nighttime light intensity. Experiments show that we can achieve up to 70% accuracy on a three-class (low, medium, high nightlights intensity) classification task. The eventual goal is to use the CNN features learned from the nightlights classification task to predict indicators of interest to the international scientific community, such as poverty, wealth, or health outcomes.

1. Introduction

The rise of Big Data in recent years has precipitated an explosion of new applications; in computer vision especially, deep learning is now used everywhere. However, one area that could benefit from further attention is the development of tools and allocation of resources for third-world countries. One challenge that has been impeding progress in the area of developmental economics is the difficulty of obtaining reliable data in these areas of the world [17].

We hope to improve this state of affairs by using nighttime light intensities to extract informative features from the corresponding daytime satellite images. Nighttime light intensities are often used as proxies or predictors for other quantities of interest, such as poverty, wealth, and health metrics—however, as scalar quantities, they are limited in the amount of information that they can convey [3, 5]. Nighttime lights are also affected by two sources of noise: overglow, the propensity for light to spread into areas surrounding the actual source, and blooming, the magnification of light over certain terrain types such as water, ice, and sand [15].

We propose to leverage modern deep learning techniques to extract useful landscape features from high-resolution satellite imagery. Recent developments in the remote sensing industry, including new entrants to the market in the form of startups such as Planet Labs and Skybox, suggest that we will soon have affordable sub-meter resolution imagery covering the entire global land surface each day [7]. In this paper, we demonstrate the use of proxies for economic development (nighttime light intensities in this case) as noisy labels for training a convolutional neural network (CNN) as a general-purpose satellite image feature extractor. We show that the CNN learns a low-dimensional feature representation of the input images that is predictive of the corresponding nightlight intensities, with up to ~70% accuracy on a three-class classification task. In future work, researchers in various fields could use the trained CNN to produce rich feature vectors with which to make predictions about socioeconomic indicators of interest.

2. Related Work

Various deep learning architectures have been explored for image recognition tasks. Chatfield et al. [2], for instance, propose three types of CNN architectures: the CNN-F (Fast), the CNN-M (Medium), and the CNN-S (Slow), which differ in stride size and pooling window size. These relatively simple architectures show performance close to state-of-the-art algorithms. Although ResNet by He et al. [8] shows better classification performance, the former architectures have the advantage of being shallower structures. The recent success of CNNs trained on ImageNet [16] has also launched various attempts to repurpose the feature representation learned from the neural net to generic tasks. Pan and Yang [14] define transfer learning and explore applications of transfer learning between different domains. Donahue et al. [4] show that deep features learned from training on the ImageNet dataset achieve performance nearing state-of-the-art algorithms for object recognition, domain adaptation, subcategory recognition, and scene recognition.
Oquab et al. [13] further show that mid-level image representations can be transferred to different domains. Moving from one image domain to another often changes the input dimension, and Long et al. [11] suggest fully convolutional networks that replace the fully connected layers with convolutional layers for the fine-tuning process.

There have been relatively few attempts to use vision techniques in the aerial image domain, but recent large-scale machine learning approaches have shown impressive results. Kluckner et al. [10] apply covariance descriptors to a multi-class randomized forest framework for semantic classification of aerial images. Mnih and Hinton [12] use large-scale neural networks, with additional help from local spatial coherence of the output labels, to detect roads in high-resolution aerial images. Despite the recent successes, applying CNN to aerial images is yet a largely unexplored domain, mainly due to the scarcity of labeled data. Xie and Jean [18] use transfer learning with nightlight intensity data to predict poverty in the African region. This project is an extension of that research with higher resolution satellite images and nighttime lights data.

3. Dataset and Features

3.1. Daytime Satellite Images

We would like to accurately predict nighttime light intensities given the corresponding daytime satellite images. To obtain the daytime satellite images, we use the Google Static Maps API to sample images given a particular latitude and longitude. The Static Maps API provides images at a range of resolutions; we use zoom level 17, which has a pixel-level resolution of approximately 1.2 meters. At this zoom level, 400-by-400 pixel images correspond to the 0.5 km resolution of the nighttime lights data. An example of a daytime satellite image is shown in Fig. 2.

![Daytime satellite image](image)

Figure 2. Daytime satellite image of a 0.5 km x 0.5 km square in Kampala, Uganda. Kampala is the capital of Uganda, and by far its largest city. This image is a typical example of an African urban landscape.

3.2. Nighttime Light Intensity Data

To obtain nighttime light intensities to use as training labels, we look at nighttime satellite images captured by the Earth Observation Group of the National Oceanic and Atmospheric Administration (NOAA)—specifically, we use data from the Visible Infrared Imaging Radiometer Suite (VIIRS) satellite, which provides images at a 15 arc-second resolution (~0.5 km). This resolution is twice as high as the 30 arc-second grid of the previous Defense Meteorological Satellite Program (DMSP) data.

The VIIRS dataset consists of monthly observations starting in January 2014 and continuing through the present day. For each month, the NOAA provides two satellite products: monthly average nighttime light intensities and the number of cloud-free observations per month are saved in separate GeoTIFF raster data files. Since the majority of our daytime imagery was taken in 2015, we choose to use VIIRS data from January-December 2015. One possible
source of noise is that some areas may use different relative amounts of lighting depending on the time of year—for example, we can imagine that colder parts of the world may use more electricity for heating and outdoor lighting in the winter if conditions are poor, while warmer parts of the world may actually decrease electricity use in the winter. To get a better nighttime light proxy of economic development, we decided to average nighttime light intensity values across the 12-month period, weighted by the number of observations in each month to produce an annual value. We then use these observation-weighted annual averages as our ground truth nighttime lights values for each grid point on the global land surface.

3.3. Dividing the Nighttime Lights into Classes

Because nighttime light intensities are defined along a continuous distribution, we discretize them into classes with a range of intensities in each class. To find a reasonable division of these light intensities, we first look at the distribution of all nighttime light intensities in Africa. Taking the VIIRS satellite images of the region, we filter out any points outside of Africa, obtain the nighttime light intensity value for each remaining pixel point location of interest, and then plot a distribution of all of these values. At the 0.5 km resolution that we have, there are over 143 million grid points in Africa. The resulting plot is shown in Fig. 3.

![Figure 3. Nighttime light intensities in Africa. There are many points with intensities close to zero. Note that the number of locations is on a log scale.](image)

Because each class should be meaningfully different visually, we bin the intensities as follows: 0 to $8 \text{nW/cm}^2\text{-sr}$, 8 to $35 \text{nW/cm}^2\text{-sr}$, and 35 to $200 \text{nW/cm}^2\text{-sr}$. The lowest light intensity class includes mostly desolate areas, the middle intensity class includes areas with some human activity such as farms or suburbs, and the highest intensity class includes most of the cities. When building our training set, we ignore images with light intensities above 200 because there are very few such images (as seen in Fig. 3), and the few that we do see look like outliers (for instance, desolate sand patches). This is supported by the known weaknesses of nighttime light intensity measurements—the effects of blooming and overglow often lead to misleadingly high intensity measurements over certain types of terrains, such as sand and bodies of water. One example daytime satellite image from each class is shown in Figs. 4, 5, and 6.

![Figure 4. Daytime image from Class 1: 0 to 8.](image)

![Figure 5. Daytime image from Class 2: 8 to 35.](image)

However, from Fig. 3 we can see that the classes are very unbalanced—the lowest light intensity class holds many orders of magnitude more points than the highest. To combat this uneven distribution, we randomly sample an equal number of images from each class for training and testing purposes, keeping in mind that our eventual goal is to learn useful feature representations.

3.4. Sampling Images from Each Class

To prevent the CNN from learning to simply classify every image as the lowest light intensity class (and instead learn features that could be useful for predicting other quantities), we balance our training data by sampling the same
ages from each class and augment our data by rotating each nighttime light intensity distribution. We sample 50,000 images from each class. For the lowest two light intensity classes, our procedure is as follows: we sample a random latitude and longitude location in Africa and find its nighttime light intensity value (the average yearly value described in Section 3.2 above). If this intensity is within the desired class range, we attempt to sample the corresponding daytime satellite image. This procedure takes too long for the highest intensity class, since randomly sampling locations throughout Africa rarely provides a nighttime light intensity above 35. Thus, for this class, we restrict our sampling to pixel point locations that were determined to be within the desired range while creating the histogram of the nighttime light intensity distribution. We sample 50,000 images from each class and augment our data by rotating each image 90°, 180°, and 270°, for a total of 200,000 training images per class. We also sample an additional 10,000 images from each class to use as the test set.

4. Methods

4.1. Convolutional Neural Networks

Deep learning approaches are often used in vision tasks due to their ability to capture hierarchical representations of data. CNNs include convolutional layers in the neural network, and their translational invariance and local connectivity allow them to solve various vision problems such as image recognition and video analysis [1]. A CNN is a feed-forward artificial neural network that usually consists of four main layer types: the convolutional layer, the ReLU layer, the pooling layer, and the fully-collected layer. Unlike a standard neural network, the neurons in each layer of a CNN are arranged in three dimensions: width, height, and depth. Additionally, neurons in a convolutional layer are not connected to every neuron in the previous layer; instead, each neuron is connected only to local regions, computing a dot product between its weights and the region it is connected to. A ReLU layer is generally placed after a convolutional layer, thresholding all activations to zero. A pooling layer downsamples the activations along the width and height, compressing the spatial dimensions. After several repetitions of [Conv-ReLU-Pool] layers, CNN models sometimes use fully-connected layers as the final layers of the network. A fully-connected layer computes class scores as in standard neural networks. An example of a common ConvNet architecture is as follows: INPUT → [[CONV → RELU]*N] → POOL?)*M → [FC → RELU]*K → FC

This layered architecture allows CNNs to learn various features of images while training, with low-level features in the earlier layers of the network and high-level features in the later layers. The output of the final fully-connected layer can be interpreted as the class score for each class, and the output of the second to last fully-connected layer can be used as a feature vector for any other task of interest.

4.2. Transfer Learning

Transfer learning is a learning process in which the domains, tasks, and distributions used in training and testing are different [14]. Our target label is nighttime intensity data, but its limited number of classes (3) and highly unbalanced nature hinder the CNN from learning useful features without help from another domain. Thus, we begin with a CNN model pre-trained on the ImageNet dataset and fine-tune it using satellite images and nighttime light intensity data through transfer learning. ImageNet is an object classification image dataset of over 14 million images with 1000 class labels that are widely used in computer vision tasks. [10] CNN models trained on the ImageNet dataset are recognized as good generic feature extractors, with low- and mid-level features such as edges and corners that are generalizable to many new tasks [4] [13]. ImageNet data comprises object-centric images while satellite images are from a bird’s-eye view, and therefore the two datasets have different feature distributions. However, the low-level features from an ImageNet-trained CNN are also present in birdseye view images, so using a pre-trained model facilitates the construction of high-level features for satellite images as well.

4.3. Predicting Nighttime Light Intensity

We begin the training process with the VGG-F model, a CNN model pre-trained on ImageNet, and fine-tune it using satellite images and nighttime intensities [2]. VGG-F has five convolutional layers and three fully connected layers, followed by a loss layer, as seen in Fig. [7]. This model takes as its input a 224 × 224 pixel image, and outputs 1000 values that correspond to the 1000 classes of ImageNet.

The Google Static Maps API returns 400 × 400 pixel images, so we must modify the VGG-F architecture. In particular, we construct a fully convolutional model by converting the fully connected layers of VGG-F to convolutional layers. The last three fully-connected layers are replaced with...
convolutional layers with strides of 6, 1, and 1, respectively. The new output has dimensions $2 \times 2 \times 4096$, representing 4096 features for each of four (overlapping) quadrants of the image. An average pooling layer is appended to the end to produce an average value of the features over the four quadrants. This average value is then used as the final input to the softmax classifier to predict the nighttime light intensity.

4.4. Training the CNN

This modified CNN model is trained with Caffe using as its initial parameters those learned from the pre-training process on the ImageNet dataset [9]. Minibatch gradient descent with momentum is used for fine-tuning, and random mirroring is used for data augmentation. 50% dropout is applied to the first two of the final three convolutional layers. For our first few trials, the learning rate begins at $1 \times 10^{-6}$, a hundredth of the final learning rate of the VGG model, as recommended in [6]. Most other hyperparameters are initially the same as in the VGG model trained for ImageNet. The convolutional layers that replace the fully connected layers are randomly initialized.

5. Results and Discussion

Much of the effort involved in this project was spent on the creation of a large labeled training dataset. In the remaining time, we ran four total trials training the CNN with different hyperparameter settings, each of which will be described in detail below. All CNN training was done on a GeForce GTX 680 GPU.

5.1. Trial 1

For the first trial, we used a CNN model pretrained on ImageNet: the VGG-F. This is an eight-layer CNN with three fully-connected layers at the end. In our implementation, we changed the fully-connected layers to convolutional layers so that the model can take in arbitrarily sized images. We then added an average pooling layer to obtain a score for each class.

In order to get started training our CNN quickly, we used an approximate mean subtraction scheme where we randomly sampled 0.05% of our training set of 600,000 daytime images and computed the mean pixel value in each of the three color channels. Before running future trials, we took the time to compute the mean pixel value for the full training set.

We chose to use 32 images per training batch for this first trial, which means that 18,750 iterations were needed for one epoch. We trained for a total of 19,090 iterations with a learning rate of $1 \times 10^{-6}$, so we saw each of our training images approximately once. For this initial trial, we did not decay our learning rate at all. Each time we evaluated the test accuracy and loss, we ran images in batches of 8 for 1,000 iterations, so we would get an estimate of the test accuracy and loss over 8,000 images.

Our accuracy and loss plots are shown in Figs. 8 and 9. The test accuracy starts to level off around 68%, but we did not run the trial to convergence—we decided to stop the trial early and restart training using the true training mean pixel values.

5.2. Trial 2

For the second trial, we recomputed the mean RGB values using all 600,000 images in our training set. We then
trained for a much longer time to reach convergence—281,400 iterations, just over 15 epochs. We began with a learning rate of 1e-6 and halved it after every 10,000 iterations, so the learning rate would decrease by a factor of about 4 after each epoch. The ending learning rate was 3.73e-15.

We used the same training batch size of 32 images, and all other hyperparameters remained the same. There did not seem to be a significant gap between the training and test curves, so we did not change the regularization strength. Since we are classifying our images into three nighttime light intensity bins, our starting training accuracy was around 33%, as expected. We evaluated the test accuracy and loss after every 500 iterations—at the last evaluation, at iteration 281,000, we had a test accuracy of 69.9%. The highest test accuracy achieved at any iteration was 70.3%. Accuracy and loss curves for trial 2 are displayed in Figs. 10 and 11.

5.3. Trial 3

Since the training accuracy looked very noisy in the first two trials, we doubled the training batch size from 32 to 64 this time around. This was also about the maximum batch size that would fit in the memory of the GPU. The test accuracy curves looked very smooth since there were 1000 test iterations with batch sizes of 8, which meant that the test accuracy used a large portion of the images in our test set. Thus, we decreased the number of test iterations from 1000 to 250 to speed up training. Our accuracy and loss curves are shown in Figs. 12 and 13. We trained just until convergence this time, a total of 58,400 iterations. We achieved a test accuracy of just under 70%, roughly the same performance as before, but with slightly less noisy training curves.

5.4. Trial 4

Since the accuracy and loss plots converged very quickly in the first three trials, we decided to lower our learning rate while keeping the batch sizes of Trial 3. We decreased the
Figure 12. Training and test accuracies vs. iteration number for Trial 3.

Figure 13. Training and test loss vs. iteration number for Trial 3.

initial learning rate by a factor of 10 to 1e-7, and then cut the learning rate in half every 10,000 iterations as before. We ran this trial for 66,400 iterations, and the accuracy and loss curves are shown in Figs. 14 and 15. In this case, the learning rate may be too low, as the final accuracy achieved was only about 65-66%.

5.5. Confusion Matrix

A normalized confusion matrix was produced after 280,000 iterations of Trial 2 and is shown in Fig. 16 for 10% of our test data. The CNN achieved an overall accuracy of 2055 out of 3000, or 68.5%. We can see that the values in the matrix are highest along the diagonal, signifying a relatively high true positive rate. As can be seen from the figure, the lowest nighttime class has the highest accuracy, and the highest nighttime class has the next highest accuracy. These statistics make sense, since daytime satellite images from both classes have relatively distinct features and the majority of the images fall into the lowest class. Most of the mistakes seem to occur when the true class is the middle one. This also makes sense: the middle class has more features that overlap with each of the other classes, and there are two directions in which a mistake could be made (either lower or higher) instead of just one. For the mistakes that are made in the lowest and highest classes, the CNN usually seems to predict the middle class rather than the class at the opposite end of the spectrum. These statistics again seem reasonable, since the middle class presumably has more features in common with the lower and higher classes than they do with each other.

6. Conclusion and Future Work

Our fine-tuning process for training the VGG-F CNN to classify nighttime light intensities seemed to be a relatively robust process. Only a few experiments with different hyperparameters were necessary before we were able to achieve around 70% accuracy on our three-class classification task. It is possible that with more time and experimentation, an even more effective setting of hyperparameters could be found, but training a CNN to perform signif-
Learning techniques to bolster research efforts in a wide range of scientific domains and increase human understanding of many of the major challenges facing the world. Researchers and policymakers often lack the necessary technical background to develop sophisticated computational approaches—we could provide the tools that are needed to solve many of their real-world problems.

References

