CS 254 Machine Learning
Final Project Report

Maxfield Green, Galen Byrd, William Wiltshire III

1. Introduction

Land use and composition is of key interest to many unique stakeholders including developers and conservationists. Land type classification is a costly and time-consuming task for which there are many possible solutions. Classifying land use and composition in person can be cumbersome and requires in field man-hours in potentially hazardous conditions and environments. Using satellite imagery to classify urban and rural land types would benefit all efforts for land development and assessment. We will start with the following 6 classes; barren land, trees, grassland, roads, buildings and water bodies.

2. Problem Definition and Algorithm

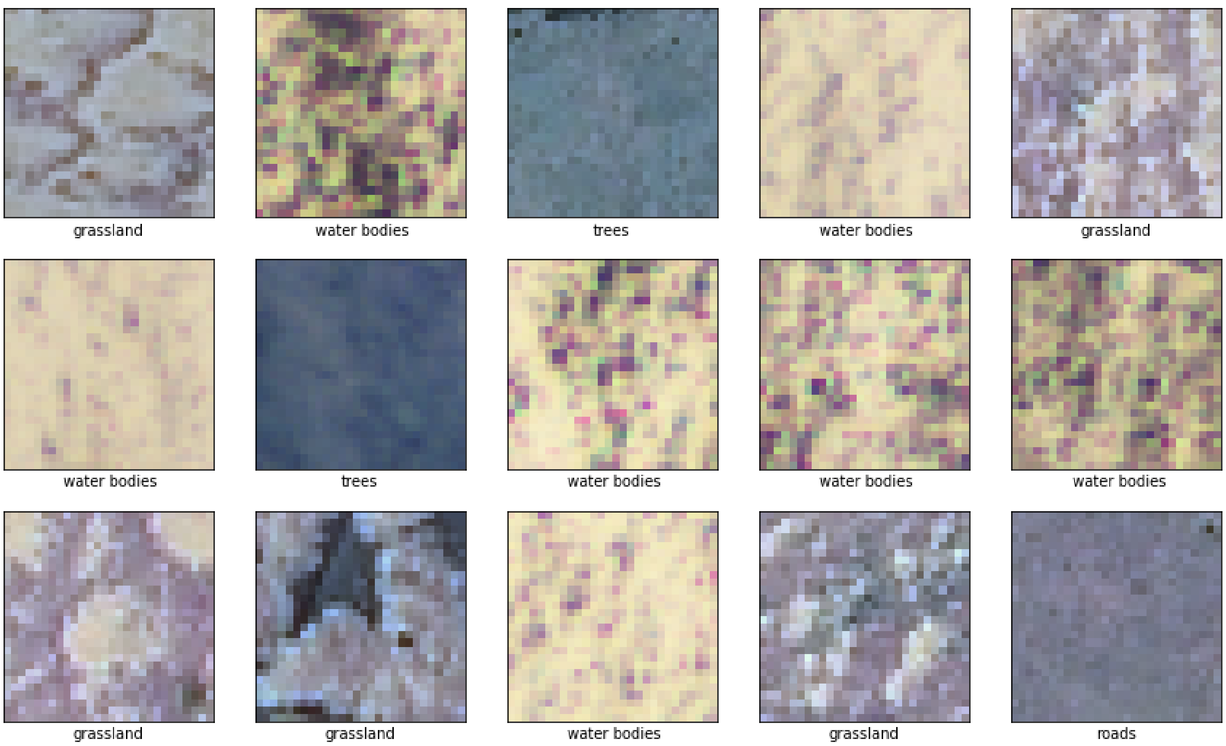
2.1 Task Definition

Our Convolutional Neural Network classifies the land cover of satellite images containing a single cover type. Given a satellite image of a landscape of uniform ground cover, our CNN will determine the type of cover on the ground. This is a difficult task as the data presents high amounts of variance and the images are hard for a human to classify with high accuracy, let alone a machine.

2.3. Dataset

Our dataset is comprised of a total of 405,000 image patches. The data were extracted from the SAT-6 Airborn Dataset released by the National Agriculture Imagery Program(NAIP). The images span across the entire United States. The original raw data is 65 terabytes. The Nasa Ames Research Center within the Advanced Supercomputing division performed several data cleaning operations to prepare the data for learning. For this project, we used refined data published in the Ames study. The original image tiles were 6000x7000 pixels, taken at a 1-meter “ground sample distance”. The Ames Research Group randomly sampled this raw data set extracting 1500 image tiles. The group then labeled the 1500 images using a novel image labeling tool. They further broke down the images by sliding a 28x28 pixel window over each image creating a large set of images labeled by land cover[6], each representing one land over type.

The resulting data set is comprised of 405,000 images patches of size 28x28. The data contains 6 land cover classes including barren land, trees, grassland, roads, building and water bodies. 324,000 images were selected to train the model while 81,000 were held out for testing purposes. We selected a few matrices and plotted them in grayscale below as a sample of the input data.



Sample of Labelled Images from Altered SAT - 6 Dataset

The dataset barely fits into python GUI available memory, therefore, we were required to use the VACC after making an attempt to use Amazon's free cloud computing services.

2.2 Algorithm Definition

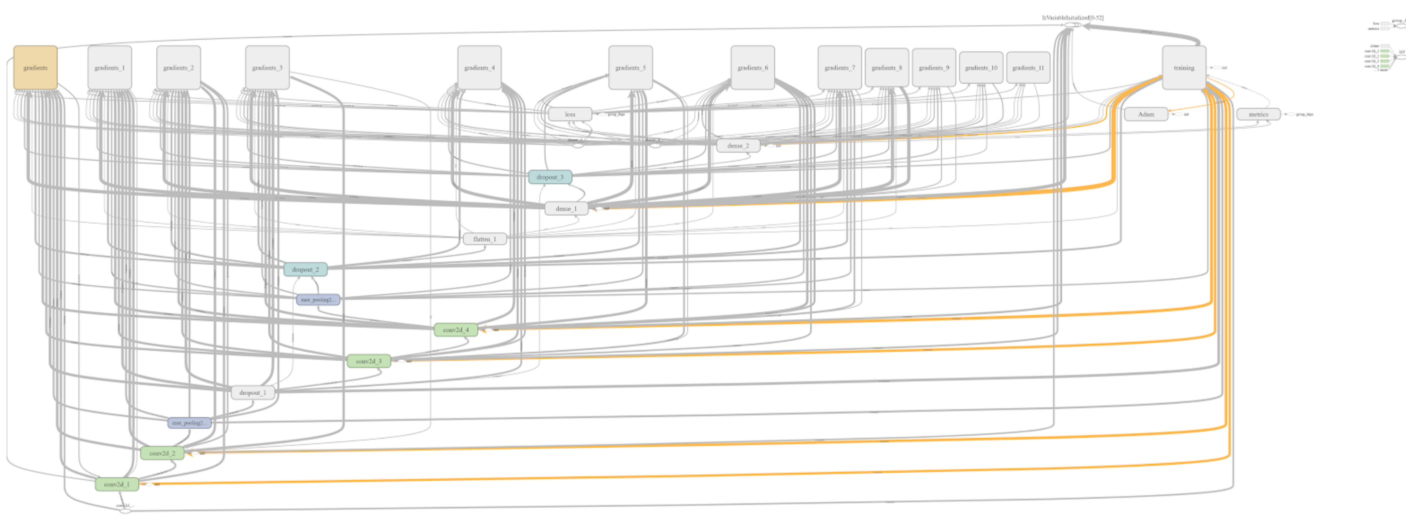
To classify land cover, a Convolutional Neural Network (CNN) is used. The CNN is built using Keras with a TensorFlow backend.

As is the case with most CNNs, ours is a stack of layers that transform the original image layer by layer until a class score is generated, predicting the label of the image. In our case, this is the land cover type. The network will learn filters that activate when visual features common to the images are passed over. The parameters of each convolution will be trained by a gradient descent of our loss function. This way, the CNN can predict the training data labels.

In addition to convolutional layers, we use several pooling layers with matrix size (2,2) in order to downsample the dimensions of the input. Dropout is also used to combat oversampling. We read that it is typical to dropout neurons with probability 0.5, so we initially choose to test these values. The final softmax function transforms the output into the probability distribution for each label.

There are 13 layers in our CNN comprised of a series of convolutions, max pooling steps, dropouts and dense functions. The structure of our network was advised by a Kaggle Kernel written by Bhumit Adivarekar. Our baseline system takes in the 28x28 images. The specific architecture of the CNN is outlined below:

Using TensorBoard, we were able to generate a visualization of our CNN.



The visualization is slightly hard to see given the size limitations of this paper, but we found it very helpful in understanding how the layers pass information between one another.

One of our goals to improve the algorithm, was to use the GPU nodes on the VACC to increase the number of parameters we could test and the amount of time we could train the network. We did exactly this. We would also like to test well-known architectures such as ResNet, AlexNet, and GoogLeNet on our data to determine if there is an increase in performance.

3. Experimental Evaluation

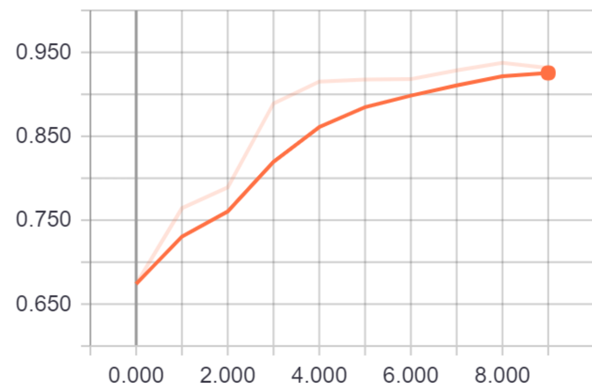
Our hypothesis is whether information about land cover can be embedded in a satellite image and then classified by a convolutional neural network. The measure we are currently using to evaluate the results of this hypothesis is accuracy and loss.

Accuracy is a decent way to measure the performance of a model but ideally we would measure the recall as well. Using only accuracy is not robust to highly skewed data. We are currently still learning to use tensorboard to generate performance measures. It would be ideal to have a variety of measures that highlight our model's strengths and expose its weaknesses. Below we have plots that visualize the accuracy and loss of the classifier over several training/testing cycle.

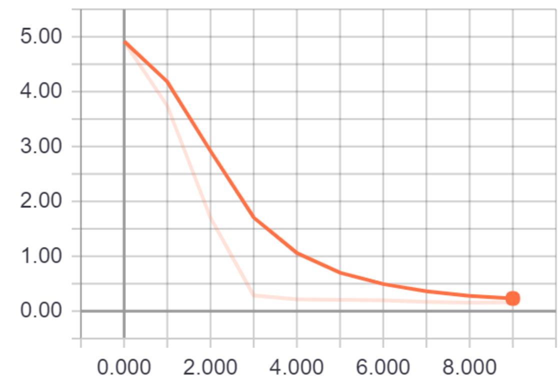
Ideally, we would continue the test/train cycle for many epochs. However, due to the large scale of the dataset, running the model for even 10 epochs is very computationally costly. On my CPU with RAM of 15 gb, one epoch of training on the full dataset takes about 3-4 hours.

3.2 Results

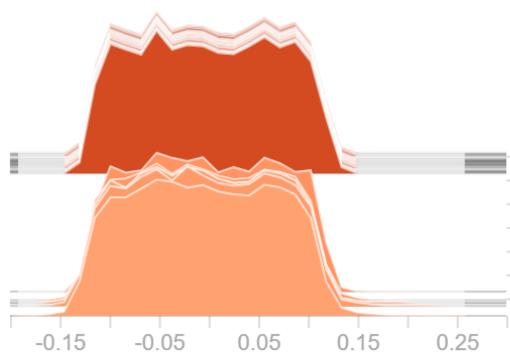
Below are loss and accuracy plots over 10 epochs of training and testing of our highest accuracy hyperparameters.



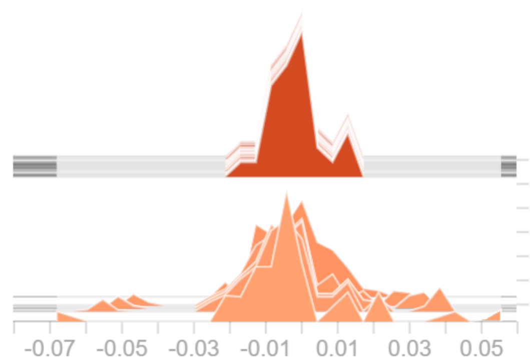
Accuracy



Loss



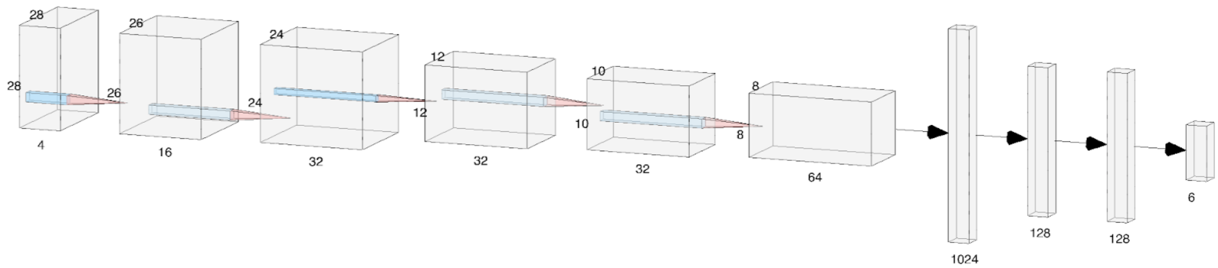
Convolution/Kernel



Convolutional Gradient

These are sample histograms from a single convolution layer over 10 epochs. Kernel and gradient densities are derived from TensorBoard. We tuned the dropout rate, number of filters in our convolutional layers and number of neurons in our dense layers. We tuned the number of filters by multiplying different layers in the model by different scalars. We tuned using a grid search with varying levels of granularity for the parameters. We also tested different learning optimizers incase the landscape of the loss function suited one in particular. We then found the ideal hyperparameters for our model to be learning rate 0.001 with Adam Optimizer, 0.2 constant drop out between three layers, an increasing set of filters starting at 16 and maxing at 64, and a dense layer with 128 neurons. These were determined to perform the best based on parameter tuning with k-folds cross validation, testing every considered operation.

Many hours over night on personal CPU’s and the GPU node on the VACC was used to hyper tune the model.



A list of layers to a company the visualization.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 26, 26, 16)	592
conv2d_2 (Conv2D)	(None, 24, 24, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 12, 12, 32)	0
dropout_1 (Dropout)	(None, 12, 12, 32)	0
conv2d_3 (Conv2D)	(None, 10, 10, 32)	9248
conv2d_4 (Conv2D)	(None, 8, 8, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 64)	0
dropout_2 (Dropout)	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 128)	131200
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 6)	774

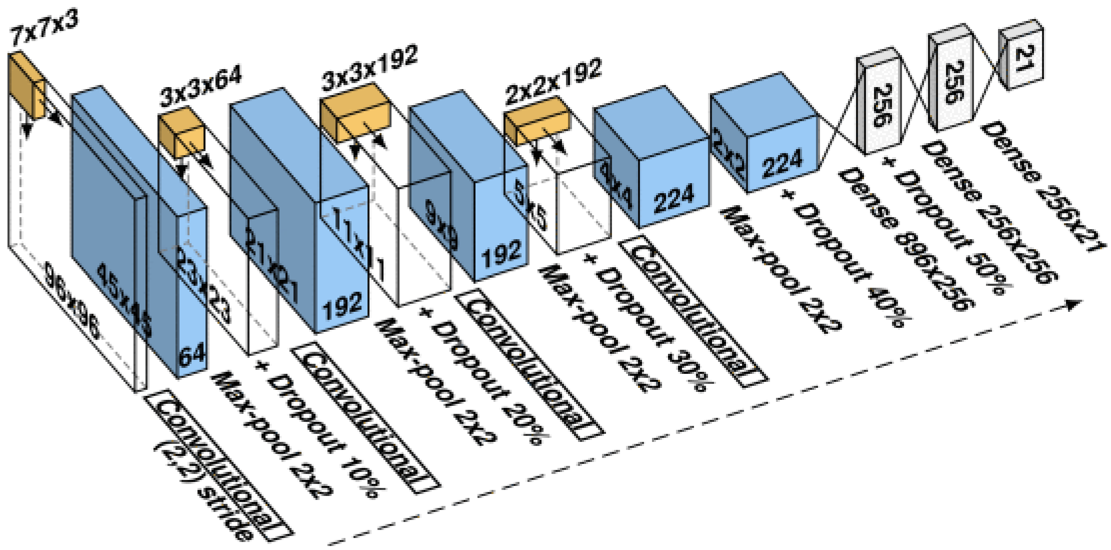
3.3 Discussion

Our hypothesis is supported, as we are able to classify the land types from satellite images at a rate of 93.25%. Other methods[2] are reporting 98% accuracy with the same data set. We found that tuning hyperparameters was a difficult and computationally costly task as we tried many different combinations of hyperparameters. Since there is no way to know which combination of hyperparameters will give us the most accurate algorithm, we are forced to test all the possible combinations that we hope might make the algorithm more accurate. Additionally, we would like to know if the error contributing to the loss is spread evenly over labels, or if there are particular labels that are harder for the classifier to deal with. A serious drawback to the model is that it can only classify images that contain one type of land cover as opposed to multiple different cover types caught in the same frame. This type of model would be very useful when boundaries are well known and uniform but possibly less useful for general classification.

4. Related Work

There have been many different solutions to the land classification problem throughout over the past decade. The current benchmark for this type of land cover classification is well in the 97% (SAT-6,1).

Alternative benchmark land-use classifier datasets, such as the one at UC Merced are very close to being fully solved. Recent projects have achieved up to 93.48% accuracy(UC Merced). This particular success was achieved with an interesting DCNN architecture. The architecture is displayed below:



To combat overfitting, they algorithm the set of neurons dropped at each epoch changes randomly. This way the downsampling is evenly distributed across the network. Originally, the algorithm progressively increased *p* dropout as the dimensions of the output were reduced. The algorithm moved from 10% to 20% to 40% and the final dropout has p = 0.5. Our final algorithm keeps the dropout constant at 20%. Generally, this maximizes regularization, however, the dropout can be optimized and fine-tuned specifically to different layers of the network (Baldi).

In the 2017 paper, “Analysis of Land Use & Land Cover Classification Using Machine Learning Algorithm SVM”, the researchers solved the same land classification problem using a different method, Support Vector Machines. The algorithm splits the data into different land use classes. They trained their SVM model with four different kernels: RBF, Sigmoid, Linear, and Polynomial. Each kernel was then analyzed and compared to show their respective advantages/disadvantages. (Shivpuje) Our problem is essentially the same, but we use a completely different method, image classification using convolutional neural networks. While we are not using their algorithm, it is still interesting to note that the problem of land use classification can be approached in many different ways.

Another study extended the findings from UC Merced by attempting to classify urban land use patterns. In this study, they chose to use a random forest as opposed to SVM or a CNN. While their accuracy has not reached levels as high as simple land classifiers, they have reached an accuracy of 79% (Yao). They use Traffic Analysis Zones to see how the urban land use pattern of a specific region may change over time due to different factors, like population growth and urbanization. This study is a good example of a further application of the simple land use classifier we are attempting to create.

5. Code and Dataset

The data is approximately 2.3 Gigabytes therefore we won't upload it with the report. However, it can be found in here:
https://drive.google.com/uc?id=0B0Fef71_vt3PUkZ4YVZ5WWNvZWs&export=download.
Our code is included in the .zip file we submitted.
There is a script for our final model that loads and preprocess data.

6. Conclusion

A convolutional neural network can be learned to classify land type using satellite image data. Our model has achieved an accuracy of 93.25%. We believe that this is a great accomplishment for us, as this is quite close to the benchmark of this dataset using more complex, computational intense and time dependent methods. With this type of technology, a subset of land types could be classified in real time as a plane flys over landscape.

Bibliography

(1) Budhiraja, A. (2016, December 15). Learning Less to Learn Better‑Dropout in (Deep) Machine learning. Retrieved from https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

(2) P. Baldi and P. Sadowski. The Dropout Learning Algorithm. Artificial Intelligence, 2014. In press.

(3) Shivpuje Prakash R, Dr. Deshmukh Nilesh, "Analysis of Land Use & Land Cover Classification (LU/LC) Using Machine Learning Algorithm SVM", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.4, Issue 11, page no. pp35-37, November-2017, Available at :http://www.jetir.org/papers/JETIR1711008.pdf

(5) Yao, Yao, et al. *SENSING URBAN LAND-USE PATTERNS BY INTEGRATING GOOGLE TENSORFLOW AND SCENE-CLASSIFICATION MODELS*. 2017, arxiv.org/pdf/1708.01580.pdf.

(6)Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert Dibiano, Manohar Karki and Ramakrishna Nemani, DeepSat - A Learning framework for Satellite Imagery, ACM SIGSPATIAL 2015.