# Classifying Urban Land Use with a Convolutional Neural Network

E4040.2017Fall.SAR.report
Simon Rimmele sar2160
*Columbia University*

## Abstract

*This project aims to reproduce and improve upon a convolutional deep learning network for the accurate classification of land use in urban environments using satellite imagery data. A successful implementation of this network would offer precise assessments of urban land use, ideally transferable and quickly scalable to provide urban planning assessments for cities which do not currently have exhaustive land use data on hand. The success of the project is highly dependent on the reliability of training data and access to existing satellite imagery benchmarks trained to identify landscape features, which provide a starting point for learning the more granular physical features of an urban environment. While this project could not improve upon the existing reference model, it does develop an alternative model architecture which can use lower resolution inputs while maintaining accuracy and improving training speed significantly.*

## 1. Introduction

This project is based on the data and models developed in *Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale* [2]. While the paper developed several deep neural network architectures and trained models for multiple European cities, this project restricts itself to one city and the VGG16 architecture because of technical constraints.

## 2. Summary of the Original Paper

### 2.1 Methodology of the Original Paper

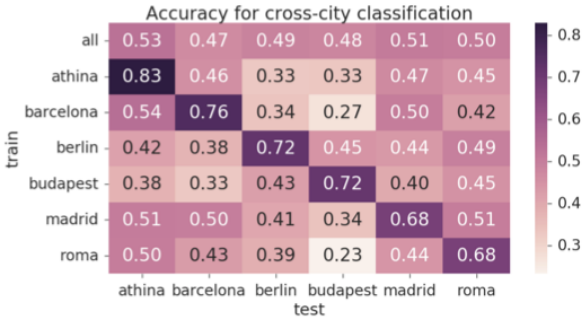
The paper being reproduced for this project used a variety of conventional convolutional neural network (CNN) architectures - including VGG16, VGG19, and ResNet50 implementations - to label satellite imagery of urban environments with associated land use categories.

### 2.2 Key Results of the Original Paper

Albert et., al were able to adapt several benchmark CNN architectures for the task of using satellite imagery to classify land use in urban environments. The authors' key goal was to adapt existing CNN architectures capable of detecting objects (e.g., trees, roads) from satellite imagery (DeepSat) [4] or items in a photograph (e.g., ImageNet) [3] into a neural network capable of classifying a satellite image by land use category e.g., *airport* or *high density urban fabric*.

A table of the accuracy achieved for selected cities in the original paper is included below. It is notable that the accuracy declines significantly when the network was trained and tested on separate cities, suggesting features are not entirely transferable across environments. Similarly, a network trained on sample imagery from all cities at once achieved only 50% accuracy.

**Figure: Albert et., al Network Test Accuracies [2]**




### 2.3 Related Notable Work & Results

DeepSat, the aforementioned satellite image labeling benchmark dataset, was released in 2015 and currently has been modeled to >99% accuracy using a variety of convolutional architectures. A CNN pretrained with weights from DeepSat is the starting point for the reference model.

## 3. Methodology

This project implements a VGG16 neural network from scratch to reproduce and experiment with the results of Albert et., al. VGG16 was chosen over VGG19 and ResNet architectures also implemented in the reference paper. This was because all three architectures performed similarly in the original experiment and VGG16 is the simplest architecture to implement. Additionally, VGG16 weights which have been pre-trained on ImageNet are freely available [5] in order to experiment with transfer learning and as a potential alternative to the unavailable DeepSat benchmark.

### 3.1. Objectives and Technical Challenges

The chief objective of this project is to independently reproduce Albert et., al's VGG16-based CNN for urban land use classification. While the VGG16 architecture is

not difficult to implement, this project did not have access to the original imagery data used in the paper, as well as the data for the benchmark DeepSat, which was one the datasets used for pre-training the reference network.

Hardware constraints also posed a technical challenge to implementation. The reference paper's architecture used a cluster of 48 CPUs for image collection and preprocessing as well as 4 GPUs with 12GB of memory apiece for training the networks. In order to work within the limitations of desktop computing and a single GPU, this project was constrained to collecting data and training a model for a single arbitrarily chosen urban area (Berlin).

## 3.2. Problem Formulation and Design

Classifying a topological satellite image by land use requires more than object recognition. A given topological satellite image's use can be defined by the relationship between objects in the image. This is the major conceptual difference from a conventional image recognition CNN such as ImageNet.

## 4. Implementation

The project was implemented entirely in Tensorflow. The exact reference model was reproduced (absent the unavailable DeepSat pretraining). In addition, the model was modified by tuning parameters and architectures in order to attempt to get better model performance without the pretraining DeepSat data.

## 4.1. Deep Learning Network

The network architecture used in this project reproduces the benchmark VGG16 architecture [3] while adding dropout between the first and second fully connected layers. This was done to mimic Albert et., al, who also included 50% dropout at this stage.

**Figure: VGG16 Architecture [2]**

Table 2: The VGG16 architecture [19].

| Block 1 | Block 2 | Block 3 | Block 4 | Block 5 | Block 6 |
|---|---|---|---|---|---|
| Conv(3,64) | Conv(3,128) | Conv(3,256) | Conv(3,512) | Conv(3,512) | FC(4096) |
| Conv(3,64) | Conv(3,128) | Conv(3,256) | Conv(3,512) | Conv(3,512) | FC(4096) |
| Max-Pool(2,2) | Max-Pool(2,2) | Conv(3,256) | Conv(3,512) | Conv(3,512) | FC($N_{classes}$) |
| | | Max-Pool(2,2) | Max-Pool(2,2) | Max-Pool(2,2) | SoftMax |

The AdaDelta optimizer was used as in the reference paper. One of the modified networks also used the Adam optimizer, but the performance difference was negligible.

## 4.2. Data Collection and Use

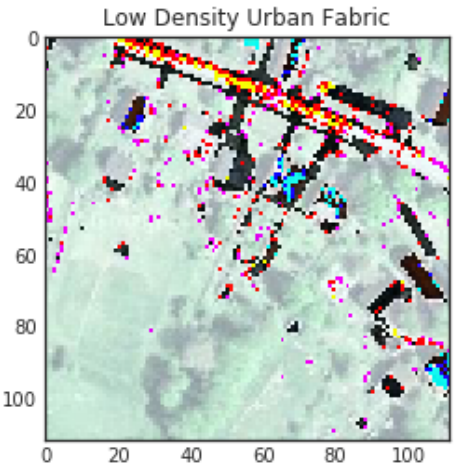
Because of aforementioned hardware constraints, this project focused on a single city. The goal was to train a VGG16 network using imagery collected from Google Maps and a corresponding land use classification from the UrbanAtlas project. While the original imagery was not available, Albert et., al made their data gathering modules publicly accessible [6]. Only small modifications were needed to run the data collection process for ~25k images on a single computer, which was divided into ~20k images for training and validation, and ~5k images for testing.

This data gathering process poses a possible inconsistency problem within the data. UrbanAtlas land use classification was created from 2005-2011, while Google Maps imagery is quite possibly much more up to date and the location may have changed uses in the intervening decade. For example, many images classified by UrbanAtlas as 'Construction Site' 5-10 years ago will now have completed structures on them. As such, there is a risk a deep learning model will apply classifications which are incorrect or inconsistent in the input data, potentially correlated with specific classes such as construction sites.

### 4.2.2 Image Augmentation

Like in the reference model, all images had the ImageNet means of each RGB channel subtracted prior to training. The reference paper also further augments the training images by cropping images randomly, which is also done in this project.

**Figure: Sample Image with Label**




### 4.2.3 Transfer Learning

One way to increase performance, whether through increasing accuracy or reducing runtime, is to initialize the network with weights from a different well-performing model. In this case, the convolutional layers of the network were initialized with weights from

ImageNet, as implemented in Caffe and converted for use in Tensorflow [5].

## 5. Results

The project tested results on three variations of the VGG16 model: a faithful reproduction of the reference model, a version pre-trained with ImageNet convolutional layer weights, and a hybrid model with regularization and different tuned parameters which was still initialized with the ImageNet weights.

## 5.1. Project Results

### 5.1.1 Direct Reproduction model

Directly copying the reference architecture and training it from scratch yielded somewhat disappointing results. 34% accuracy on the test set was achieved when training the model from scratch. While not being a reliable classifier, it was still three times more accurate than randomly choosing one of the 10 land use classes.

The relationship between training, validation, and testing error in this model suggests the network will overfit the training data. Training accuracy was often 5-15% higher than the best validation/test accuracy. The reference model does not have any regularization so this behavior is not unexpected.

### 5.1.2 Transfer Learning Model

A network preloaded with VGG16 ImageNet weights but otherwise identical to the reference model performed quite poorly. Accuracy results over epochs had much higher variance than the reference model. However, the transfer learning model achieved its best validation performance more quickly.

### 5.1.3 Regularized Model

A hybrid of the first two models performed comparably to the best traditional VGG16 model while also lowering the training time by 2/3 from over an hour to just 20 minutes. This network downsampled the imagery to 112x112 pixels, while reducing the width of the fully connected layers from two 4096 width layers to [2048, 1024]. Regularization was also applied through a small L2 penalty. The intermediate convolutional layers were still initialized with ImageNet weights.

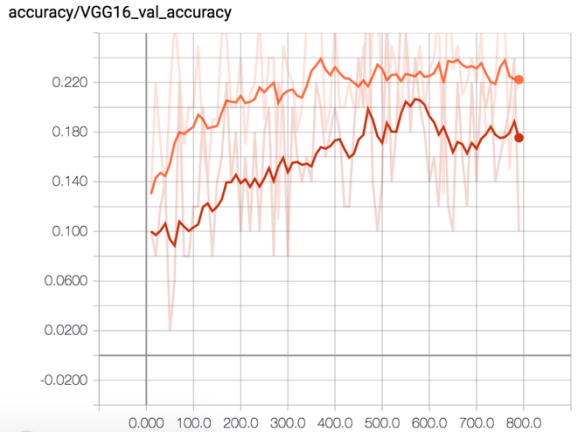


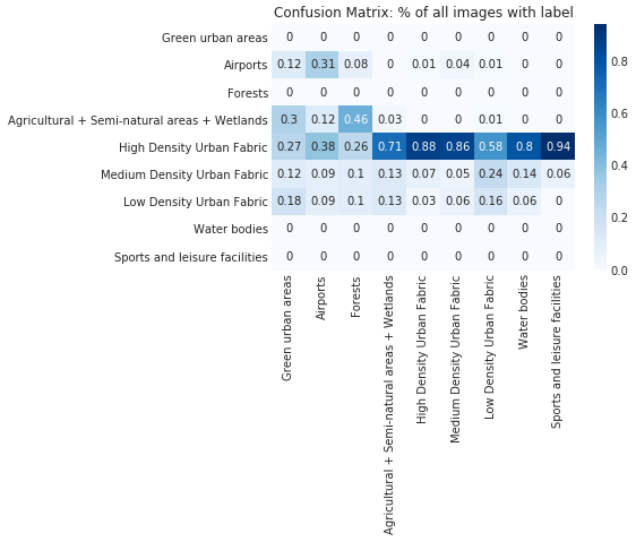

**Figure: Validation Accuracy of reference model [red] and hybrid model [orange] – *trend smoothed for legibility***

## 5.2. Comparison of Results

Albert et., al achieved 72% accuracy for Berlin in their work. Their best result was using ResNet-50 so their exact VGG16 performance is unpublished. However, they state that ResNet performed "only slightly" better than VGG16.

This project achieved 34% accuracy on the Berlin dataset, significantly lower than the reference paper. The unavoidable differences in implementation may have contributed to the result. This project was only able to use batch sizes of 50 images in comparison to 100 image batches in the original. Being unable to pre-train the network on a DeepSat benchmark may have also contributed to the lower performance. The confusion matrix may also yield some insight on the differences in model performance:

**Figure: Confusion matrix for best performing model**

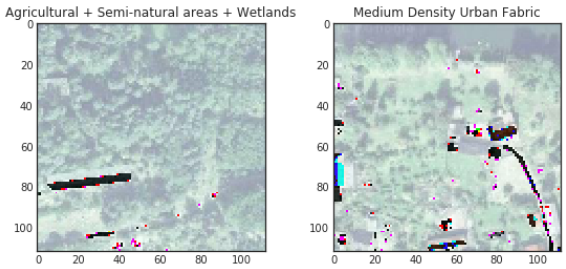



The network identifies the core *urban fabric* (cityscape) quite well, however it tellingly completely fails to

distinguish between *bodies of water* and *urban fabric*. The DeepSat model capable of 99% accuracy is explicitly trained to recognize *bodies of water* and *buildings* as categories. Applying transfer learning from DeepSat may be very beneficial in this case. Unfortunately ImageNet proved a poor substitute.

The network also consistently misclassified *green urban areas* as either *urban fabric* or *semi-natural+agricultural* land. This could be suggestive of ambiguity in the classifications themselves. What separates *green urban areas* from these other quite similar categories may not be strictly separable from a feature perspective. Two examples are provided below:

**Figure: Two images with ambiguous classifications**




If the network had been able to correctly differentiate among the described cases its accuracy would have jumped to well over 50%; much closer to the reference paper performance. Having a network already pre-trained on DeepSat may have made all the difference in this case.

## 5.3. Discussion of Insights Gained

This project highlighted the power of transfer learning and the importance of initialization when training deep neural networks. Having access to pre-trained weights from a well-trained model with similar features – DeepSat in the case of this project – potentially would be able to improve classification accuracy by over 30 percent.

Relatedly, initializing a deep neural network with weights from a successful model of identical architecture can aid in performance. Initializing the VGG16 network with a successful set of VGG16 weights trained on ImageNet resulted in higher initial accuracy and a more rapid improvement over epochs. Even when a pre-trained does not yield any improvement in classification accuracy, it may still offer performance improvements over a 'cold-start' of random initializations.

## 6. Conclusion

This project reproduced a VGG16 deep learning model from Albert et., al capable of identifying urban land use categories with 72% accuracy. The strict reproduction only achieved 34% accuracy, potentially because the DeepSat benchmark data used for pre-training in the reference paper was not available. However, an alternative implementation developed in this project is capable of reaching comparable accuracy of 33% on imagery with half the resolution while completing training 66% more quickly than the strict reproduction.

## 6. Acknowledgement

Many thanks to Stack Overflow.

## 7. References

[1] *Model files corresponding to networks* https://sar2160@bitbucket.org/ecbm4040/2017_assignment2_sar2160.git

[2] *Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale*. A. Toni Albert, J. Kaur, M.C. Gonzalez, 2017. In Proceedings of the ACM SigKDD 2017 Conference, Halifax, Nova Scotia, Canada.

[3] *Very deep convolutional networks for large-scale image recognition.* Simonyan, Karen, and Andrew Zisserman. arXiv preprint arXiv:1409.1556 (2014).

[4] *Deepsat: a learning framework for satellite imagery***.** Basu, Saikat, et al. " *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015.

[5] *VGG in TensorFlow* http://www.cs.toronto.edu/~frossard/post/vgg16/

[6] *Urban Environments Public Github Repository* https://github.com/adrianalbert/urban-environments

## 8. Appendix

### 8.1 Individual student contributions - table

| | sar2160 |
|---|---|
| Last Name | Rimmele |
| Fraction of (useful) total contribution | 100% |
| What I did | All work |