CS230

Aerial Imaging Classification with a Lightweight Network for Edge Computing

Brian Chan
Department of Computer Science
Stanford University
bchan17@stanford.edu

Abstract

Edge computing and satellite imagery trends demand lightweight algorithms for computational analysis. In this paper, I investigate applying SqueezeNet, a neural network with AlexNet-level accuracy and 510x less size, to the task of aerial imaging land use classification with the NWPU-RESISC45 dataset.

1 Introduction

Recent advancements in satellite technology continue to increase the affordability, accessibility, and accuracy of aerial satellite imaging. The rising demand for edge computing capabilities and aerial imaging insights is a motivation for the development of fast and less compute-intensive algorithms. Such processes can be run on assets in the field such as the satellites themselves.

Aerial imaging insights, particularly over time, are valuable for efforts ranging from urban planning and economic analysis [5] to military efforts, in both urban, rural, developed, and developing regions.

The input to our algorithm is an image from satellite point of view. I investigating using a neural network derived from SqueezeNet, an extremely lightweight version of AlexNet, to predict what type of land use the pictured region can be classified as. I utilize the NWPU-RESISC45 dataset, one of the most difficult benchmark datasets for land use classification.

2 Related work

2.1 Current State

Aerial image and land use scene classification is a hot problem many researchers are engaged with, particularly due to the informational advantages and insights that high performing algorithms promise to deliver.

The publishers of the NWPU-RESISC45 dataset, Cheng et. al, have established a benchmark for the performance of various state of the art deep learning methods on NWPU-RESISC45 [3]. It is observed that, after transfer learning, fine-tuned versions of AlexNet, VGGNet-16, and GoogLeNet perform quite well with overall accuracies of approximately 85.16, 90.36, and 86.02 respectively. They also establish NWPU-RESISC45's advantage over other popular datasets up until then, which will be covered in the "Dataset" section of this paper. Their pretrained weights developed off of ImageNet comes with both pros and cons; it is beneficial for the basic reasoning of transfer learning helping create accurate algorithms due to our lack of established benchmark aerial imaging data, but it is also not ideal due to most images in ImageNet being natural, ground-level photos of daily items;

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

it is argued that aerial imaging is quite different with different types of edges and low-level features that transfer learning might not accommodate.

Directly related is the original paper documenting SqueezeNet [4], where Iandola et al. were able to train a neural network with 50x less parameters than AlexNet that achieved Other small networks for scene classification of remote sensing images are also investigated by Chen et al. [2]. They experiment with Teacher-Student training, where a student model utilizes ground truth data to learn while also attempting to match the output of a large pre-trained model (the "teacher"). This is the general state of the art direction that is beneficial, in aiming to create small networks that are at least as accurate as larger networks.

2.2 Novel Approaches with Tradeoffs

Other quite novel solutions have also been explored in the pursuit of higher performance. One is the utilization of multiple networks to arrive at predictions, coupled with majority voting [6]. The authors liken it to a "Hydra", and a strength of this is higher robustness given some of the hydra heads may be better suited for a specific task, operating on the philosophical direction that there is no all-encompassing algorithm. However, this algorithm evidently requires more computational resources, but is nevertheless a unique method.

Other techniques include the fusing of two channels [9], where the authors fuse extracted features from both the raw image as well as one with a saliency algorithm applied to it to derive conclusions. Simultaneous, multiscaled input for deep learning is also investigated in a paper by Luus et al. [1] that concludes a single neural network can be trained with multiscale views to achieve better performance, where they conclude with accuracies of 93.48% while versus 85.37% accuracy with the conventional method of scale invariant feature transform (SIFT).

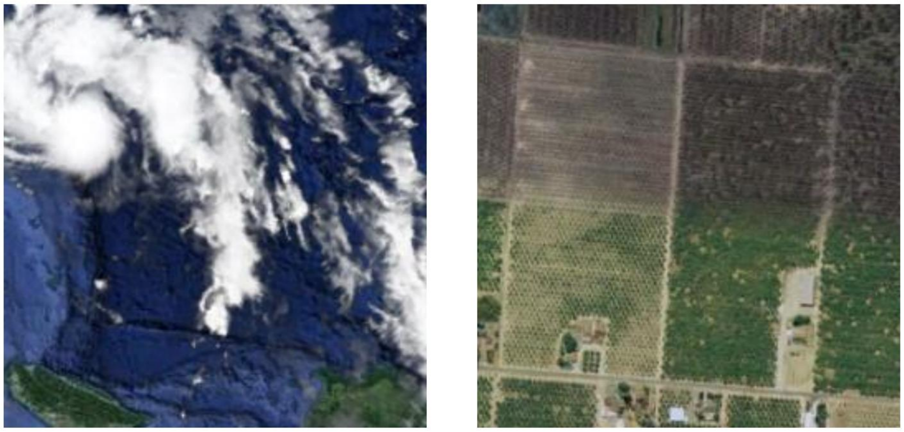
Overall, there are many approaches to developing higher performance for image scene recognition, with inherent tradeoffs between accuracy and computational resources.

3 Dataset and Features

The dataset I am utilizing is NWPU-RESISC45, a 2017 dataset presented by Gong Cheng, Junwei Han, and Xiaoqiang Lu [3]. It contains 31,500 256x256 RGB images split into 45 classes of 700 images each.

It is one of the most comprehensive datasets regarding land use. In Cheng et al.'s paper proposing the dataset, it is compared to other popular datasets such as the UC Merced Land Use Dataset and the Siri-Whu dataset [3]. The UCM dataset only has 12 classes and 100 samples per class, while the Siri-Whu dataset largely focuses on only urban scenes from China.

NWPU-RESISC45 has a more diverse assortment of classes as well as more samples per class. The authors developed it while keeping in mind the desire to have both urban and rural classes as well as scenes classifiable by large features and small features alike. A few examples:



Sample images from the "cloud" and "rectangular_farmland" classes.

Sample images from the "airport" and "island" classes.

The dataset includes scenes including large features such as dense_residential, chaparral, mobile_home_park, river; smaller features such as storage-tank and basketball court; and even classes such as sea-ice and cloud. Features are extracted through convolutional layers and the SqueezeNet algorithm pretrained on ImageNet.

For all models, the input images are resized to 64x64x3; resizing from their original size of 256x256x3 while preserving the 3 RGB channels. Furthermore, the data is split per class into 80:10:10 ratios for training:validation:test sets, which is 560:70:70 image ratios.

4 Methods

As a baseline, I implement transfer learning using a VGG-16 network trained on ImageNet, as it is shown that VGG-16 performs better than AlexNet and GoogLeNet. [3]. In a tradeoff for performance, I investigate a more lightweight network with VGG-16 in only adding 1 or 2 fully connected (FC) layers after the convolutional layers. After features are extracted through VGG-16's of Convolutional layers, the added FC layers are followed by a softmax of 45 nodes for the algorithm to predict the probabilities of a sample being classified one of the 45 classes. The loss utilized per example, for all models in this paper, is that of categorical cross-entropy:

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

M = total number of classes, $y_{o,c}$ = 1 if this observation o's true class is c, $p_{o,c}$ = predicted probability of the observation o belonging to c

This incurs a penalty that is calculated only based off of the predicted probability for which class the sample truly belongs to, as $y_{o,c}$ is 1 for the class the image belongs to and 0 otherwise. Furthermore when analyzing the correct class, the log error exponentially punishes predictions the farther away from the correct prediction (1.0) they are.

The next model I implemented uses transfer learning from a model called SqueezeNet [8], originally implemented in Caffe [4]. The selling point of this algorithm is that it has AlexNet-level accuracy, while having 50x less parameters and overall 510x less size, making it a perfect algorithm to adapt and investigate for our motivations and purpose.

I utilized a Keras version and its pretrained weights, then removed the top layers and replaced them with various FC layers (a hyperparameter I tuned). By using the pretrained weights, my algorithm is able to extract the features using SqueezeNet to get quality feature insights without needing to train SqueezeNet from scratch. The FC layers I train myself then allow the algorithm to learn to make decisions on what features align with what class.

Furthermore, I additionally iterate on the model by exploring the fine-tuning of later weights in the SqueezeNet model to make better predictions for our specific task.

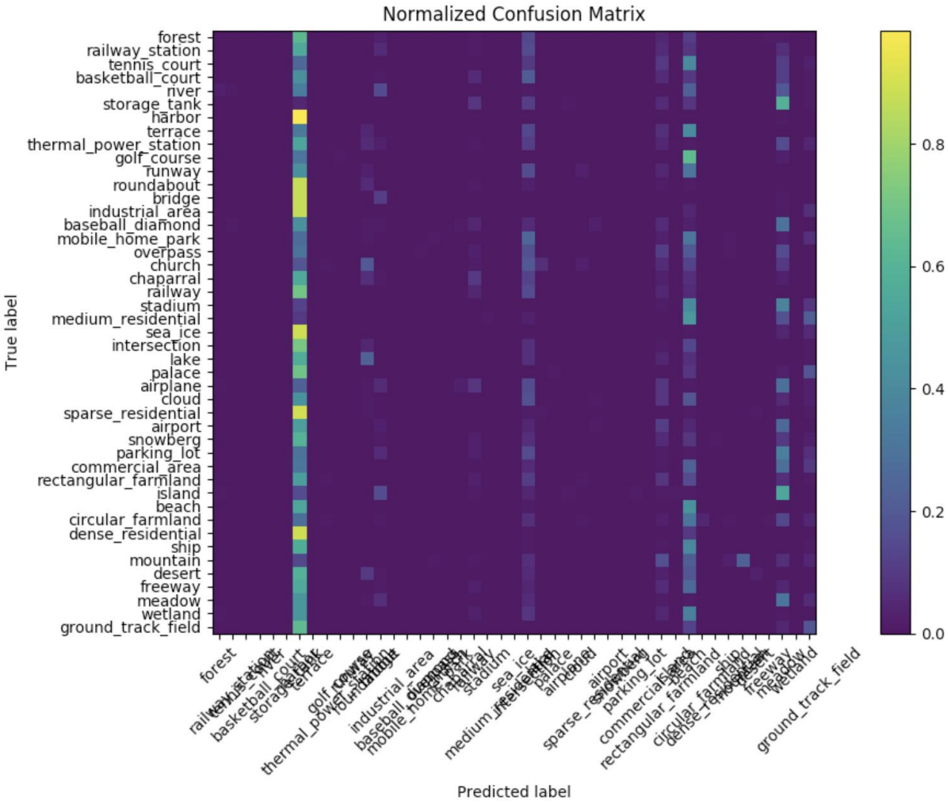
In my experiments, I also explore tuning various hyperparameters including the number of FC layers and nodes in FC layers, the optimizer used, and the learning rate.

5 Experiments and Results

Model	FC Layers	Batch Size	Learning Rate	Train Acc	Val Acc	Layers Frozen
vgg_transfer_baseline	[512]	64	.01	.744	.6924	31/33
sqz_transfer	[512] x2	64	.001	.3177	.3606	38/42
sqz_transfer	[1024]	64	.001	.3085	.3886	38/40
/textitsqz_transfer	[1024] x2	64	.001	.3546	**.4063**	38/42
sqz_transfer	[2048]	64	.001	.1865	.2889	38/40
sqz_transfer	[2048] x2	64	.001	.3790	.4038	38/42
tuned_sqz	[1024] x2	64	.1 * e^6	**.4164**	.3249	33/42
tuned_sqz	[512] x2	64	.1 * e^4	.3417	.2414	33/42

Model descriptions: vgg_transfer_baseline is the hefty baseline model used with transfer learning and applied to our dataset. For our experimental models, sqz_transfer are the models that utilized SqueezeNet with weights pretrained on ImageNet, and either 1 or 2 fully connected layers afterwards. tuned_sqz are the models that unfroze the last convolutional block of SqueezeNet to allow fine-tuning of those SqueezeNet weights during training.

Notably, the fine-tuned models had lower accuracies than those that used transfer learning and only updated the final FC layers. This indicates that the tuned models were likely overfitting, which is corroborated by their respectively low validation accuracies. With the italicized model having the highest validation accuracy, running it on the test set, provides the following confusion matrix [7]:



From this confusion matrix, it appears that the algorithm had a high frequency of predicting a few certain classes, which is intuitively understood from evident vertical bands. We can analyze this by calculating the precision, recall, and F1 score performances.

P - Precision: True Positives / (True Positives + False Positives)
R - Recall: True Positives / (True Positives + False Negatives)
F1 Score: 2 * P * R / (P + R)

Across 45 classes the average recall was .091 and the average precision .0025, which shows that this model as a whole was unfortunately very poorly performing.

From the beginning, this general imbalance in the classes that were predicted was tough because the data in the number of samples is balanced per class. In response, I instituted dropout layers, the final results which are reported in the results table enough.

Overall, it appeared that these iterations of SqueezeNet and Fully Connected layers are not robust applications for scene classification of aerial imaging as they all performed quite far below the baseline I implemented with VGG.

6 Conclusions and Future Work

I investigated developing a small neural network that would be able to produce results on par with AlexNet and VGG-16 for land use scene classification. I found that the highest performing model was the one denoted as sqz_transfer_2, which utilized transfer learning with SqueezeNet and 2 FC layers following it. Though they were all close, this algorithm potentially outperformed the others as it struck a balance between being able to model more complex functions out of the lower complexity of the preceding SqueezeNet itself.

If I had more time, and especially with other team members, I would investigate further architectures to take advantage of transfer learning with SqueezeNet, especially with regards to tuning hyperparameters, performing transfer learning with different models, and selecting the numbers of layers to freeze in SqueezeNet itself.

The various models developed were not very robust, but we gained new information regarding the capabilities of smaller and lighter networks as applied to remote sensing image classification problems, specifically the new and challenging NWPU-RESISC45 dataset. Furthermore, we gained insight about the transfer learning capabilities of SqueezeNet as applied to this problem.

7 Contributions

I worked by myself on this project.

8 Code

The code used in this project is found here: https://github.com/brianjychan/landuse

References

[1] F. P. S. Luus ; B. P. Salmon ; F. van den Bergh ; B. T. J. Maharaj. “Multiview Deep Learning for Land-Use Classification”. In: *IEEE* (2015). DOI: 10.1109/LGRS.2015.2483680.

[2] Guanzhou Chen et al. “Training Small Networks for Scene Classification of Remote Sensing Images via Knowledge Distillation”. In: *Remote Sensing* 10.5 (2018). ISSN: 2072-4292. DOI: 10.3390/rs10050719. URL: http://www.mdpi.com/2072-4292/10/5/719.

[3] Gong Cheng, Junwei Han, and Xiaoqiang Lu. “Remote Sensing Image Scene Classification: Benchmark and State of the Art”. In: *CoRR* abs/1703.00121 (2017).

[4] Forrest N. Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size”. In: *CoRR* abs/1602.07360 (2016). arXiv: 1602.07360. URL: http://arxiv.org/abs/1602.07360.

[5] Neal Jean et al. “Combining satellite imagery and machine learning to predict poverty”. In: *Science* 353.6301 (2016), pp. 790–794.

[6] Rodrigo Minetto, Mauricio Pamplona Segundo, and Sudeep Sarkar. “Hydra: an Ensemble of Convolutional Neural Networks for Geospatial Land Classification”. In: *CoRR* abs/1802.03518 (2018).

[7] Sklearn. “confusion matrix”. In: *Sklearn* (2018). URL: `https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html`.

[8] Jesper Wohlert. “keras-squeezenet”. In: *GitHub repository* (2017). URL: `https://github.com/charlespwd/project-title`.

[9] Fuxian Liu Yunlong Yu. “A Two-Stream Deep Fusion Framework for High-Resolution Aerial Scene Classification”. In: *Computational Intelligence and Neuroscience* 2018 (2018). URL: `https://www.hindawi.com/journals/cin/2018/8639367/ref/`.