CS230 Project: Deep Learning for Semantic Segmentation of Remote Sensing Imagery

Sherrie Wang
Institute of Computational & Mathematical Engineering
Stanford University
sherwang@stanford.edu

Will Chen
Dept of Computer Science
Stanford University
wic006@stanford.edu

Nick Guo
Dept of Computer Science
Stanford University
nickguo@stanford.edu

Abstract

Accurate segmentation of remote sensing data could benefit applications such as crop yield forecasting and food security, but is hindered by a lack of segmentation labels. In this work, we train two convolutional neural networks on a multi-label image classification task and transfer the learned features to segmentation using class activation maps (CAMs). Our models achieve high classification accuracy, but we observe a sizable gap between classification and segmentation performance and that deeper models do not yield an advantage over simpler models in either assessment.

1 Introduction

Identifying the location and characteristics of croplands would greatly benefit agricultural development, food security assessment, and poverty reduction. This is especially important in Africa, where the population is projected to increase by 1.3 billion people between 2017 and 2050, and in sub-Saharan regions where over a quarter of people are food insecure.

The main challenge of applying deep learning to segment remote sensing data is the lack of datasets with segmented labels. The world of natural images has COCO, PASCAL, ADE20K, and more, which contain tens of thousands of hand-annotated photos to indicate what objects are pictured and where; there is no analogous dataset for satellite imagery, due perhaps to the difficulty of training humans to recognize objects in remotely sensed data. As a result, satellite datasets in regions of high impact have only image-level labels.

We therefore start our methodological development in the United States, where ample crop segmentation ground truth is available thanks to the USGS's Cropland Data Layer (CDL). We simulate the data-poor setting by training a deep convolutional network on a multi-label classification problem. Inputs to the model are satellite images with 7 optical channels, and the output is a segmentation prediction obtained from the outputs and weights of the neural network's last layers.

2 Related work

In the field of remote sensing, most work to map cropland has done so at an individual pixel-level, not taking into account the spatial context around that pixel. Studies from 2017 have begun to apply convolutional neural networks (CNNs) to create context-aware land cover maps, but architectures are still quite shallow and rudimentary [1, 2, 3, 4, 5]. The maps have also been constrained to highly local regions due to small amounts of data available.

Within computer vision, there has of course been much research on image segmentation. However, crop data available from regions of the world like Africa are not tagged at a pixel-by-pixel level;

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

we only have data for whether or not crops appear in a general swath of land. Thus our task differs from traditional segmentation tasks. There has been promising recent work on deep learning techniques able to derive segmentation from end-to-end learning [6], so we explore the efficacy of such techniques in this paper. In particular, the work by Zhou et al. demonstrates that their class activation mapping (CAM) technique allows models that are trained for classification tasks to then localize class-specific image regions from the target image. This has powerful applications to our segmentation task since we are limited to end-to-end learning on the data from Africa.

3 Data

We export a dataset of Landsat 8 imagery in the Midwestern United States using Google Earth Engine. Here we describe the dataset and CDL as a source of ground truth.

3.1 Landsat 8

Landsat is a series of Earth-observing satellites jointly managed by the USGS and NASA. Landsat 8 provides moderate-resolution (30 m) satellite imagery in seven surface reflectance bands: ultra blue, blue, green, red, near infrared, shortwave infrared 1, and shortwave infrared 2 [7]. Images are collected on a 16-day cycle and often affected by different types of contamination, such as clouds, snow, and shadows [8]. Remote sensing scientists often solve this problem by generating pixel-level composites of several images [9].

We generate and export a median composite for the year 2016 over the corn belt of the United States, covering parts of Missouri, Iowa, Illinois, Indiana, and Kentucky (Figure 1). The image spans 4.5 degrees latitude and 8.0 degrees longitude and contains just over 500 million 30-by-30 meter pixels.

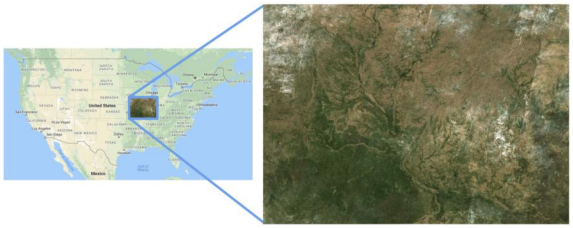

Figure 1: Landsat 8 median composite showing our study area in the midwestern United States. The image was exported piece-wise using Google Earth Engine.

3.2 Cropland Data Layer (CDL)

The Cropland Data Layer (CDL) is a raster geo-referenced land cover map collected by the USDA for the entire continental United States [10]. It is offered at 30 m resolution, so that each Landsat 8 pixel has a corresponding CDL label. CDL includes 132 detailed classes spanning field crops, tree crops, developed areas, forest, water, and more. In our dataset of the corn belt, we observe 78 CDL classes. The four most common classes — deciduous forest, corn, soybean, and grassland/pasture — account for 85% of the dataset. The remaining classes are each less than 5% of the dataset; we aggregate them all into a single "other" class. From here we treat CDL labels as ground truth and use them to evaluate the performance of our neural network.

4 Methods

4.1 Image processing

We divide our 500 million pixel Landsat image into a grid of 200k patches of size (50,50). Each patch has 7 channels corresponding to the 7 Landsat bands described above. Since the Landsat composite contains NaN values where the satellite sensor failed, we remove image patches with more than 50% NaN readings and set the rest of NaN values to zero. Our final dataset has 194k patches.

The patches were downloaded from Google Earth Engine as TIF files, and we converted each patch into a .tfrecords file for streamlined processing in TensorFlow. TFRecords enabled us to train using all of our data despite it being significantly larger than the available memory on our machines.

4.2 Multi-task learning

To create a method that can be used in settings that lack full segmentation ground truth but have image-level labels, we define a multi-label classification task on which to train networks that will then be transferred to the segmentation task.

The task is to detect whether 5 classes appear in a given patch; the 5 classes are the 4 most common CDL classes observed in our dataset and an "other" class for all other CDL classes. The segmentation labels are converted into 5-dimensional binary vector labels. An element of this vector equals 1 if the corresponding CDL class appears in more than 5% of the patch, and 0 otherwise. E.g. if a patch contains only corn and soybean pixels, its label would be $[0, 1, 1, 0, 0]$, corresponding to ["other", "corn", "soybean", "deciduous forest", "grassland/pasture"]. The loss function is

$$J(W^{[l]}, b^{[l]}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{C} -(y_j^{(i)} \log \hat{y}_j^{(i)} + (1 - y_j^{(i)}) \log(1 - \hat{y}_j^{(i)})) \tag{1}$$

where $W^{[l]}, b^{[l]}$ are the weights and biases at layers l of our model, $y_j^{(i)}$ is the ground truth binary label for class j of example i, and $\hat{y}_j^{(i)}$ is the model output for class j of example i.

4.3 Shallow CNN architecture

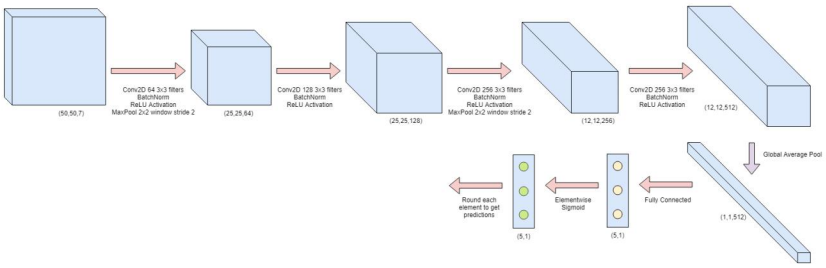

Figure 2: Our shallow CNN model architecture, adapted from the CS 230 example code.

The shallow model is adapted from a simple shallow CNN as provided by the course. It is a sequence of convolutional layer, and ReLU activation repeated 4 times with a max pooling layer at the end of every other block. This is followed by a global average pooling layer, fully connected layer, and final sigmoid layer to get a binary classifier for each of the 5 classes. We used the default parameters along with batch normalization.

4.4 ResNet-50 architecture

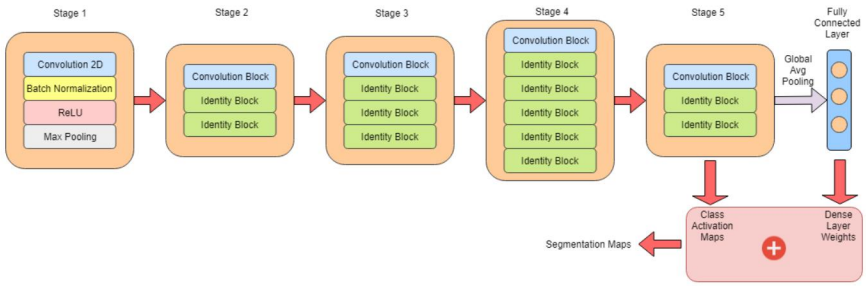

Figure 3: Our ResNet-50 model architecture, adapted from the CS 230 example code.

We adapted a ResNet-50 architecture for the multi-label classification task. The model architecture is detailed in Figure 3 and contains a sequence of convolutional and identity blocks. We modified the convolution block strides to 1 so the last convolutional layer outputs (12,12) for each filter (allowing us to get relatively high-resolution segmentation). This is followed by a global average pooling layer to aggregate each filter's information along spatial dimensions, as suggested in [6]. Last are the usual fully-connected layer and elementwise sigmoid to get image-level class activations.

4.5 Segmentation using class activation maps

To derive segmentation from a network that outputs a vector prediction, we follow the work of Zhou *et al.*[6] and create class activation maps (CAMs) using CONV layer outputs and dense layer weights.

For an input image, let $f_k(x, y)$ be the output of the last convolutional layer at position (x, y) and filter k, w_k^c be the weight of filter k for class c in the dense layer kernel, and b^c be the dense layer bias for class c. Then the class activation map for class c is defined as

$$CAM^c(x, y) = \sum_k w_k^c f_k(x, y) \tag{2}$$

For our task, we have 5 class activation maps of dimension $(12, 12)$. We upsample this image to $(50, 50)$ by repeating each pixel 4 times in each dimension and adding padding. To create the segmentation prediction, we train a multinomial logistic regression on a small number of images to map each pixel of 5 class activation values to a prediction of which class should be selected.

5 Results and Discussion

5.1 Neural network hyperparameters

We started experimentation with the default hyperparameters as provided by the course: learning rate = 0.001, batch size = 32, epochs = 25, momentum = 0.9, and Adam optimizer. The shallow model performed well with these parameters. For the ResNet-50, we increased the number of epochs to 50 since the model took longer to learn. We ran experiments varying batch size (e.g. 64 and 512), but this did not affect model performance. The ResNet-50 initially overfit to the training set, with validation loss significantly larger than training loss. We therefore experimented with incorporating an L2 loss for each of the weights in the model as well as dropout after each ResNet stage. From our experiments, we chose to use L2 regularization with a coefficient of 0.01.

Our primary metric is segmentation accuracy on the validation data. We took the extracted class activation maps and computed accuracy by comparing the argmax along each pixel of the class activation maps with the ground truth pixel value.

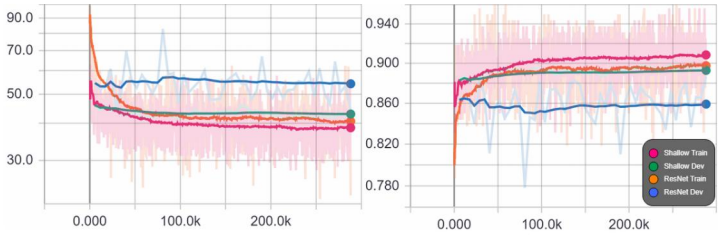

Figure 4: Loss (left) and accuracy (right) plots for the shallow CNN and ResNet models.

5.2 Shallow and deep networks can perform well on the classification task

Both the shallow model and the ResNet-50 models perform very well on the multi-label classification task, achieving 89.9% and 88.6% test accuracy respectively at their best epochs. Accuracy is calculated as the number of correct predictions across the 5 classes for all samples. Since the 5 classes are equal to 1 78.4%, 59.5%, 63.8%, 64.6%, 73.3% of the time, guessing the majority label would yield an accuracy of 67.9%. The task is therefore easy enough for a shallow network to learn, and deeper architectures do not lead to gains in accuracy. Train and dev losses and accuracies are shown

for the two models in Figure 4. Precision and recall are also high for the 5 classes; confusion matrices are shown in Figure 5.

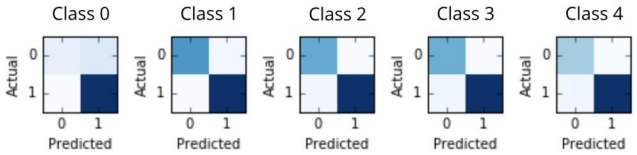

Figure 5: Confusion matrices for the shallow CNN's multi-label classification.

5.3 High classification accuracy does not translate to high segmentation accuracy

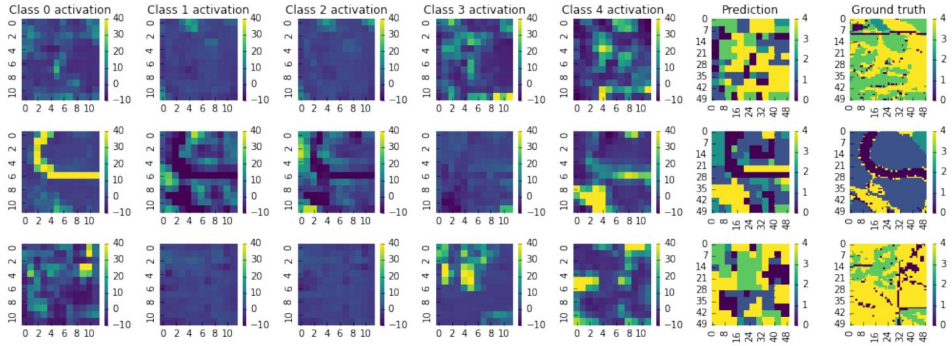

Figure 6: Segmentation predictions from our best-performing model, the shallow CNN. Each row shows 5 class activation maps, our prediction, and ground truth for an example image. Segmentation accuracies are 0.42, 0.51, and 0.48 respectively.

	Task Accuracy	Segmentation Accuracy
Shallow CNN	89.9%	57.0%
ResNet-50	88.6%	51.7%

Table 1: Shallow Network vs ResNet-50 performance

Segmentation accuracy obtained from CAMs was lower than task accuracy: the shallow model achieved 57.0% and the ResNet 51.7% accuracy. We tried a variety of ways to combine the 5 CAMs into a segmentation map, including (1) taking the argmax across classes, (2) normalizing the maps by the mean and standard deviation or median across spatial dimensions before taking the argmax, and (3) fitting a multinomial logistic regression on the CAMs to predict the correct class at each pixel for a small number (500) of images. We found that the logistic regression worked best, and justify using segmented ground truth in this procedure with the feasibility of generating hundreds of segmentation labels by hand in future datasets.

Despite relatively low segmentation accuracy, there is noticeable correspondence between the CAMs and the ground truth. In Figure 6, we see class 4 highly activated for correct areas in all 3 images, and class 0 highly activated for the river in image 2.

6 Conclusion/Future Work

Both shallow CNN and ResNet-50 performed well on the multi-label classification task over 5 crop classes. Despite the difficulty of the problem, both the shallow network and ResNet-50 are able to achieve above 50% segmentation accuracy with some preprocessing using weights learned from a simple logistic regression model. This indicates that the task does not require the extra expressiveness that the ResNet-50 model provides and that both models likely learn a similar representation. Given more time, things we would like to explore include: experimenting with different models for segmentation such as U-net, looking at how a model trained on a crop/no-crop binary classification task would translate to segmentation, and finally utilizing the temporal features of the data.

7 Contributions

GitHub repository:

`https://github.com/nickguo/cs230_project`

Sherrie: data acquisition and processing, class activation maps, segmentation evaluation, milestone writeup, poster, report

Will: model and training/evaluation infrastructure, model regularization, experiments, milestone writeup, poster, report

Nick: data pipelining and TensorFlow setup, TFRecords, milestone writeup, poster, report

References

[1] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 881–893, Feb 2017.

[2] A. B. Hamida, A. Benoit, P. Lambert, L. Klein, C. B. Amar, N. Audebert, and S. Lefèvre, "Deep learning for semantic segmentation of remote sensing images with rich spectral content," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 2569–2572, July 2017.

[3] N. Kussul, A. Shelestov, M. Lavreniuk, I. Butko, and S. Skakun, "Deep learning approach for large scale land cover mapping based on remote sensing data fusion," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 198–201, July 2016.

[4] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Computer Vision – ACCV 2016* (S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, eds.), (Cham), pp. 180–196, Springer International Publishing, 2017.

[5] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, pp. 8–36, Dec 2017.

[6] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015.

[7] D. P. Roy, M. A. Wulder, T. R. Loveland, C. E. Woodcock, R. G. Allen, M. C. Anderson, D. Helder, J. R. Irons, D. M. Johnson, R. Kennedy, T. A. Scambos, C. B. Schaaf, J. R. Schott, Y. Sheng, E. F. Vermote, A. S. Belward, R. Bindschadler, W. B. Cohen, F. Gao, J. D. Hipple, P. Hostert, J. Huntington, C. O. Justice, A. Kilic, V. Kovalskyy, Z. P. Lee, L. Lymburner, J. G. Masek, J. McCorkel, Y. Shuai, R. Trezza, J. Vogelmann, R. H. Wynne, and Z. Zhu, "Landsat-8: Science and product vision for terrestrial global change research," *Remote Sensing of Environment*, vol. 145, pp. 154–172, Apr. 2014.

[8] A. K. Whitcraft, E. F. Vermote, I. Becker-Reshef, and C. O. Justice, "Cloud cover throughout the agricultural growing season: Impacts on passive optical earth observations," *Remote Sensing of Environment*, vol. 156, pp. 438–447, Jan. 2015.

[9] W. Lück and A. van Niekerk, "Evaluation of a rule-based compositing technique for Landsat-5 TM and Landsat-7 ETM+ images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 47, pp. 1–14, May 2016.

[10] "Usda national agricultural statistics service cropland data layer. published crop-specific data layer [online].," 2016.