CS230

Early Crop Type Image Segmentation from Satellite Radar Imagery

Joanna Zou
Civil Engineering
Stanford University
zouj@stanford.edu

Walter T Dado
Earth Systems Science
Stanford University
wdado@stanford.edu

Ryan Pan
SCPD
Stanford University
yp93@stanford.edu

Abstract

We investigate the capacity of synthetic aperture radar (SAR) data for early-season crop classification. Specifically, we develop an image segmentation approach to discern land cover and crop types by the middle of the main growing season. We employ a UNet architecture, achieving overall accuracy of 75% on held out data in the US Midwest.

1 Introduction

Information on the distribution of crop types early on in a season can be useful in predicting food shortages or surpluses (1), particularly in developing countries where crop data is relatively sparse. Traditional methods, such as ground or aerial surveys of crop distributions, tend to be tedious to conduct and inconsistent. While past work has examined using satellite imagery for crop type mapping, the performance of learning models trained on spectral images is reduced from variability in cloud coverage, weather, and other atmospheric conditions.

Our objective is to develop a learning model which identifies crop types near the planting period using synthetic-aperture radar. We use exclusively Sentinel-1 SAR data (12), instead of adding spectral (visible wavelength) data as well, for two main reasons. Firstly, SAR has the advantage of 'seeing' through clouds, and thus having more consistent, high-quality data coverage across the world. Although the United States Midwest (our study area) is not particularly cloudy, we want to build an approach that can generalize well to equatorial regions which have much higher growing season cloud cover (18). Secondly, we want to explore the specific capacity of Sentinel-1 for early season crop classification, given that (a) early seasons across the globe bring rain and clouds, rendering spectral data unhelpful, and (b) early season crop classification using SAR data remains relatively unexplored.

2 Related Work

Previous efforts at crop type mapping using spectral satellite imagery have employed a wide range of modelling approaches. More basic machine learning approaches include pixel-based random forests and change detection approaches (4; 6). Increasingly, though, deep learning approaches have gained popularity to decrease pre-processing and automatically learn complex features. Examples include vanilla deep neural networks (5), CNN implementations such as ResNet, LeNet, and CaffeNet to leverage spatial context within the imagery, and RNNs such as bidirectional LSTMs to extract signal from temporal elements of crop development (7; 11).

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

However, for our specific problem, we seek to fully segment images by less than halfway through the growing season. This limits the extent of our temporal signal, and makes bidirectional LSTMs inappropriate for the task. Furthermore, though pixel-wise approaches could work, they miss valuable signal in the surrounding pixels. To this end, Wang et al. employ a UNet architecture to fully segment images for crop type classification with good success even in a label-limited context (16). Rustowicz et al. (10) extend this work by employing a convolutional LSTM which simultaneously incorporates both temporal and spatial signal.

Our context, with a different data set that uses only SAR data over a limited time frame, calls for an approach which combines elements of the above. Because we have fully labeled data over a wide spatial extent but only a short time series, we will experiment with convolutional UNet and combined ConvLSTM approaches which can appropriately incorporate spatial and temporal data.

3 Dataset and Features

Our dataset comes from the Sentinel-1 satellite, which collects Synthetic Aperture Radar (SAR) images (12). SAR data involves 'active' (sending out) microwaves in the C-band (certain wavelength/frequency), and measuring the returned backscatter. Backscatter is influenced by both the vegetation on the ground and the moisture content of both vegetation and soil. As a result, Sentinel-1 has been used in many agricultural contexts including soil moisture mapping, yield prediction, and crop classification (13; 14; 15) Our work, as detailed below, will focus on crop classification.

We define a 450,000 km^2 study area over the United States Midwest, as in Wang et al. (16). Within this region, we create an 8x8 grid of equally sized patches. Then, for the purposes of this work, we randomly select 21 of these grid cells for analysis (due to exporting data limits). In order to collect our image data, we employ Google Earth Engine, an online cloud computing platform (17). For our predictors, we have 3 image bands: 2 polarizations (VV, VH) and the incidence angle of the transmission. However, since temporal changes over the growing season can provide valuable signals, we incorporate three observations of each of these predictors. For each of the first three months of the growing season, defined as April through June, we take a pixel-wise median composite of available Sentinel-1 imagery. The resulting data set has 9 bands: 3 time steps for each of the 3 predictors. For this analysis, we only look at the 2018 growing season. Then, we randomly sample 600 pixels throughout our defined grids and gather the 256x256 pixel grid around it as a single image. In total, we have 12,600 images for training, validation and test. For developing the model, we allocate 8400 to training (14 zones), 2400 to validation (4 zones), and 1800 to test (3 zones). Each image is fully labeled at the pixel level.

Our labels come from the Cropland Data Layer, a product maintained by the US Department of Agriculture that segments the country into 30 meter labeled pixels including labels for all major crop and land cover types (8). These labels, though, have 255 unique values, which for the context of our problem would complicate model building. Instead, we aggregate into 12 classes based on the 11 most common land cover types in our study region and a class for 'other'. The classes and relative frequencies are listed in Table 5.

4 Methods

As mentioned above, we seek to fully segment images. One of the most popular architectures for such a task is the UNet model. The UNet, first proposed by Ronneberger et al. (3), works via repeated convolutions and pooling layers along a 'contraction' path to capture context and signal in the image, then passes that data along an 'expansion' path which allows for the final output to be the same dimensions as the input. The UNet approach is particularly appropriate for this application because we have both a limited number of bands (3) per image and a limited number of time steps. We leverage the capacity of the UNet to develop complex context in the contraction path (as the number of filters increase by powers of 2 each block), while still effectively outputting fully-segmented images. On the other hand, the UNet does not explicitly make use of the temporal signal (though it can implicitly do so with the inclusion of image bands retrieved from multiple time points). Given that the changes in land cover, as crops begin their growth, might contain valuable

signal, a second approach proposed by Shi et al. (2015) (9) is implemented to combine both spatial and temporal processing via a convolution LSTM network (ConvLSTM). We experiment with both of these frameworks, comparing the preferred implementations on a validation set, before providing a final estimate of generalization error on the held-out test set for the chosen model.

Specifically, we choose a UNet model with 10 'blocks' that are split into two main paths: the "contraction path" (blocks 1-5), in which the input 256 x 256 x 9 (HxWxC) image is down-sampled using convolutions and pooling 5 times to an 8 x 8 x 1024 shape, then the "expansion path" (layer blocks 6-10), in which the image is up-sampled and combined with the corresponding contraction path image to recover the original image dimensions (256 x 256) and predict pixel labels (Figure 1). For our loss function, we choose to employ cross-entropy loss after softmax activation on the final layer.

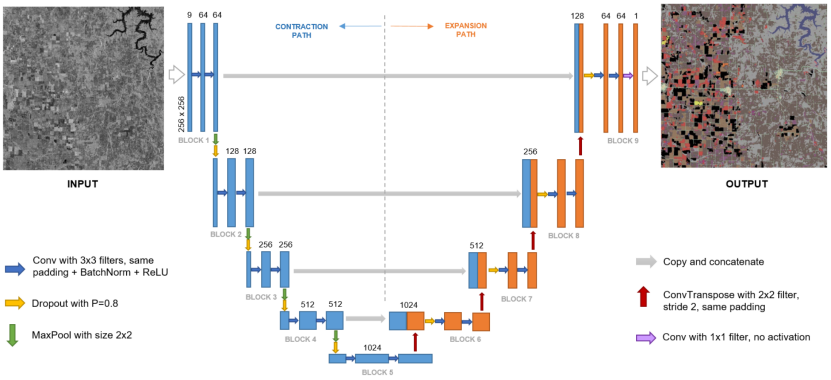

Figure 1: Final UNet architecture.

For the ConvLSTM approach, we stack several ConvLSTM layers, followed by 3D or 2D convolutions to output a 2D segmented image. The ConvLSTM layer is of a similar nature to a typical LSTM, but with 2D matrices representing pixel values of the bands instead of a 1D vector of values as inputs, outputs, and hidden states. We make use of Tensorflow's implementation of said structure, which is based on the code from the original paper. We tune the number of layers and filters as described below. Unfortunately, ConvLSTM layers are quite computationally expensive, limiting the number of filters per layer. Inspired by Shi et al (9), who introduced the approach, we essentially use the ConvLSTM layers to 'encode' signal and context from the images, and 'decode' using the subsequent convolutional layers.

5 Experiments and Results

Our initial basic UNet model, labeled as "M1" in Table 5, utilizes only convolutional and max-pooling layers in the contraction and expansion paths. We evaluate the performance of the model over 20 epochs of training with the full dataset. A relatively small mini-batch size of 16 is chosen since each sample image in the training set of 8,400 images and development set of 2,400 images contains a large number of pixel labels (256 x 256 = 65,536). The baseline model performs with 76.27% accuracy on the training data and 67.35% on the validation data, indicating potential overfitting of the training data.

In order to improve upon the baseline UNet performance, we iteratively introduce new components or change existing components of the baseline architecture. For the following set of experiments, we compare the training and validation performance of the baseline model to that of iterations on the model over 20 epochs:

- *Change optimization algorithm.* After changing the optimizer from stochastic gradient descent (SGD) to Adam, training accuracy improved by 5.3%. We started with SGD but the learning curves had shown drastic oscillation over within 10 to 15 epochs. Consequently,

we decided to experimented on Adam expecting the momentum factor could better regulate the gradient descend progress.

- *Introduce learning rate decay.* While the Adam optimization algorithm already has a effect of learning rate decay with each iteration, we additionally apply the Keras function ReduceLROnPlateau with a patience factor of 5. This function reduces the optimization learning rate by a factor of 0.2 when the previous 5 epochs show no improvement. This showed to increase accuracy by an additional 4.2%.
- *Regularize using dropout.* The UNet architecture in M2 is modified with two different Dropout schemes in order to cut down high variance: one where the Dropout layer is implemented before MaxPooling in Blocks 4 and 5 of the contraction path, modeled after (19) (M4); and one where Dropout layers are implemented after MaxPooling in all blocks, modeled after (20) (M5). Using a dropout rate of p=0.5, it was found the second Dropout scheme resulted in a greater regularization effect.
- *Tuning the dropout hyperparameter.* Using the Dropout scheme in M5, the model was tested at three values of p: 0.1, 0.5, and 0.8. The highest dropout rate of p=0.8 resulted in the greatest improvement in training accuracy (by 7.42%) and in validation accuracy (by 12.02%) compared to M2. Besides, it also provided the best performance on handling overfitting.
- *Tuning the learning rate hyperparameter.* The model in M6 and M7 combines the previous modifications of using Adam optimization, learning rate decay, and Dropout with p=0.8; the difference between them is the learning rate parameter in Adam. We find that higher training and validation accuracy is achieved using the larger initial learning rate parameter of α=0.001, as this likely prevents the optimization from premature convergence when also implementing learning rate decay to ensure convergence at an optimum.

Table 5 summarizes the results of the experiments, using training accuracy and validation accuracy as performance metrics.

Table 1: Model iterations and final accuracy over 20 epochs.

Iter. No.	Modification	Training Accuracy	Validation Accuracy
M1	Baseline UNet	76.27%	67.35%
M2	Change optimizer from SGD to Adam	81.62%	67.39%
M3	M2 + Introduce learning rate decay	85.83%	69.54%
M4	M2 + Dropout (p=0.5) in Blocks 4 and 5 before MaxPool	82.66%	66.94%
M5	M2 + Dropout (p=0.8) in all blocks after MaxPool	83.50%	76.06%
M6	M2 + M3 + M5 + Learning rate = 0.001	82.36%	78.46%
M7	M2 + M3 + M5 + Learning rate = 0.0001	80.77%	75.80%

We do not perform significant iteration on the ConvLSTM model, limited by the computational expense (more than 2 fully-convolutional CLSTM layers overloaded the GPU). The two main methods were comparing 3D convolutions on the output sequence from CLSTM layers versus 2D convolutions using the output final state of the CLSTM layers. We found that using 2D convolutions to output a segmented image performed better, and plateaued in performance much sooner than the UNet implementations. The preferred model employed 2 layers of 2D convolutions on the final state output from the 2 CLSTM layers. After 15 epochs, validation accuracy plateaued at 75%. F1 scores from this implementation on the validation set is in Figure 3.

The final UNet model M6 is illustrated in Figure 1. By 30 epochs, the training accuracy plateaus to a value of 82.36% and validation accuracy plateaus to a value of 78.46%, improving upon the performance of M1 by 11.11% (Figure 4, see appendix). On the held-out 1,800 test images, the test accuracy is 74.83% and the average F1 score across all classes is 0.75. This result suggests that the modifications chosen from the experiments have reduced the variance in the performance of the model, such that the discrepancy between the training and validation performance of the model is small. Moreover, the consistency between the test accuracy and the training/validation accuracy further indicates that the model is not overfitting the training data.

It is observable on the confusion matrix (Figure 2) that the final model made considerable errors in predicting a proportion of the crop types, such as Alfalfa and Winter Wheat. We believe the errors

were made for two reasons. Firstly, and most importantly, our data set is too skewed (Figure 3) for the model to learn the patterns of the minor classes. Secondly, a crop type could be classified incorrectly to another type because of similar bandwidth value or confusions brought up by particular agricultural activities. For example, as shown in Figure 5 (see appendix), the bandwidth value of alfalfa does not increase over time, as we might expect from a crop, because farmers periodically harvest it to feed livestock.

Finally, we recognize that the model does not quite reach the levels of performance demonstrated elsewhere in the literature (11; 10), which achieve average F1 scores greater than 0.8 and up to 0.91 for similar style task in Germany, or accuracy into the 90s in Eastern Europe (15). However, given the limitations imposed by our task (short temporal sequence, no spectral data) we feel this is a reasonable level of performance, with great promise for expansion to better data sets and new regions.

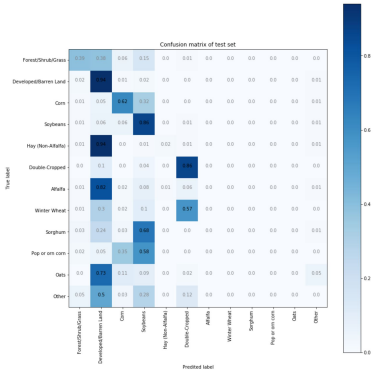

Figure 2: Confusion matrix of preferred UNet specification on the held-out test set.

Class	Frequency	UNet F1	CLSTM F1
Forest, Wetland, Grassland, etc.	47%	.55	.45
Developed Land, Cities, Barren	9.3%	.88	.88
Corn	18.5%	.71	.71
Soybean	21%	.76	.65
Other hay (non alfalfa)	2.5%	.02	0
Double Cropped Winter Wheat/Soybean	0.8%	.39	.27
Alfalfa	0.3%	0	0
Winter Wheat	0.2%	0	0
Sorghum	0.01%	0	0
Pop/Orn Corn	0.01%	0	0
Oats	0.01%	0	0
Other Crops	0.1%	.002	0

Figure 3: Frequency of crop types, and F1 scores of preferred UNet and ConvLSTM approach

6 Conclusion/Future Work

In conclusion, we find that deep learning frameworks are able to segment land cover types in a high-intensity agricultural region adequately by midway through the growing season. The preferred model, a UNet architecture, achieves an average F1 score of 0.75, while a ConvLSTM approach performs similarly, but with much greater computational expense. We find ourselves limited by infrequent temporal resolution and skewed classes.

Future work could fill in the need for additional, higher temporal resolution data as well as better addressing the skewed class problem. As Sentinel-1 continues to ramp up their program, data will be available at up to 6-day intervals, compared to the 12-24 day returns currently available. This will greatly increase the amount of data available for an early season classification task, with great potential for improvement. Given the promise shown by ConvLSTM models with just 3 time steps, we expect they might be highly relevant in this future context. Further experimentation with encoder/decoder models in this vein leveraging spatial and temporal variation seem promising.

7 Contributions

W. Dado: gathered, exported, and assembled the data set using Google Earth Engine and appropriate data parsing functions in Python. Implemented the ConvLSTM-based models and performed experimentation with them. R. Pan: implemented interactions on the UNet model by adding new components and tuning hyper-parameters. Evaluated, demonstrated, and interpreted results produced by the models. J. Zou: tested various UNet architectures from literature, compared performance of model iterations, and tuned hyperparameters. All contributed to the writeup.

References

[1] Becker-Reshef, I., Justice, C., Sullivan, M., Vermote, E., Tucker, C., Anyamba, A., Small, J., Pak, E., Masuoka, E., Schmaltz, J., Hansen, M., Pittman, K., Birkett, C., Williams, D., Reynolds, C., Doorn, B. (2010). Monitoring Global Croplands with Coarse Resolution Earth Observations: The Global Agriculture Monitoring (GLAM) Project. Remote Sensing, 2(6), 1589–1609. https://doi.org/10.3390/rs2061589

[2] Aung H.L., Uzkent B., Burke, M., Lobell, D., and Ermon S. (2020). "Farm Parcel Delineation Using Spatio-temporal Convolutional Networks". In: arXiv:2004.05471 [eess.IV].

[3] Ronneberger, O., Fischer, P., Brox, T. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: arXiv:1505.04597 [cs.CV].

[4] Conrad, C., Dech, S., Dubovyk, O., Fritsch, S., Klein, D., Löw, F., Schorcht, G., Zeidler, J. (2014). Derivation of temporal windows for accurate crop discrimination in heterogeneous croplands of Uzbekistan using multitemporal RapidEye images. Computers and Electronics in Agriculture, 103, 63–74. https://doi.org/10.1016/j.compag.2014.02.003

[5] Cai, Y., Guan, K., Peng, J., Wang, S., Seifert, C., Wardlow, B., Li, Z. (2018). A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach. Remote Sensing of Environment, 210(April 2017), 35–47. https://doi.org/10.1016/j.rse.2018.02.045

[6] Bargiel, D. (2017). A new method for crop classification combining time series of radar images and crop phenology information. Remote Sensing of Environment, 198, 369–383. https://doi.org/10.1016/j.rse.2017.06.022

[7] Scott, G. J., Marcum, R. A., Davis, C. H., Nivin, T. W. (2017). Fusion of Deep Convolutional Neural Networks for Land Cover Classification of High-Resolution Imagery. IEEE Geoscience and Remote Sensing Letters, 14(9), 1638–1642. https://doi.org/10.1109/LGRS.2017.2722988

[8] Boryan, C., Yang, Z., Mueller, R., Craig, M. (2011). Monitoring US agriculture: The US department of agriculture, national agricultural statistics service, cropland data layer program. Geocarto International, 26(5), 341–358. https://doi.org/10.1080/10106049.2011.562309

[9] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems, 2015-Janua, 802–810.

[10] Rustowicz, R., Cheong, R., Wang, L., Ermon, S., Burke, M., Lobell, D. (2019). "Semantic Segmentation of Crop Type in Africa: A Novel Dataset and Analysis of Deep Learning Methods." Computer Vision and Pattern Recognition (CVPR) Workshops (IEEE Conference), 1, 75–82.

[11] Rußwurm, M., K rner, M. (2018). "Multi-temporal land cover classification with sequential recurrent encoders." ISPRS International Journal of Geo-Information, 7(4), 1–19.

[12] "Sentinel-1 Satellite." European Space Agency. Web. https://sentinel.esa.int/web/sentinel/missions/sentinel-1/satellite-description. Accessed May 2020.

[13] Paloscia, S., Pettinato, S., Santi, E., Notarnicola, C., Pasolli, L., Reppucci, A. (2013). Soil moisture mapping using Sentinel-1 images: Algorithm and preliminary validation. Remote Sensing of Environment, 134, 234–248. https://doi.org/https://doi.org/10.1016/j.rse.2013.02.027

[14] Zhuo, W., Huang, J., Li, L., Zhang, X., Ma, H., Gao, X., Huang, H., Xu, B., Xiao, X. (2019). Assimilating Soil Moisture Retrieved from Sentinel-1 and Sentinel-2 Data into WOFOST Model to Improve Winter Wheat Yield Estimation. Remote Sensing, 11(13), 1618. https://doi.org/10.3390/rs11131618

[15] Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A. (2017). Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. IEEE Geoscience and Remote Sensing Letters, 14(5), 778–782. https://doi.org/10.1109/LGRS.2017.2681128

[16] Wang, S., Chen, W., Xie, S. M., Azzari, G., Lobell, D. B. (2020). Weakly supervised deep learning for segmentation of remote sensing imagery. Remote Sensing, 12(2), 1–25. https://doi.org/10.3390/rs12020207

[17] Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment, 202, 18–27. https://doi.org/10.1016/j.rse.2017.06.031

[18] Whitcraft, A. K., Vermote, E. F., Becker-Reshef, I., Justice, C. O. (2015). Cloud cover throughout the agricultural growing season: Impacts on passive optical earth observations. Remote Sensing of Environment, 156, 438–447. https://doi.org/10.1016/j.rse.2014.10.009

[19] Lamba, H. (2019). Understanding Semantic Segmentation with UNET: A Salt Identification Case Study. Towards Data Science. https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

[20] Gupta, Divam (2019). Image Segmentation Keras : Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras. Github. https://github.com/divamgupta/image-segmentation-keras

Appendix

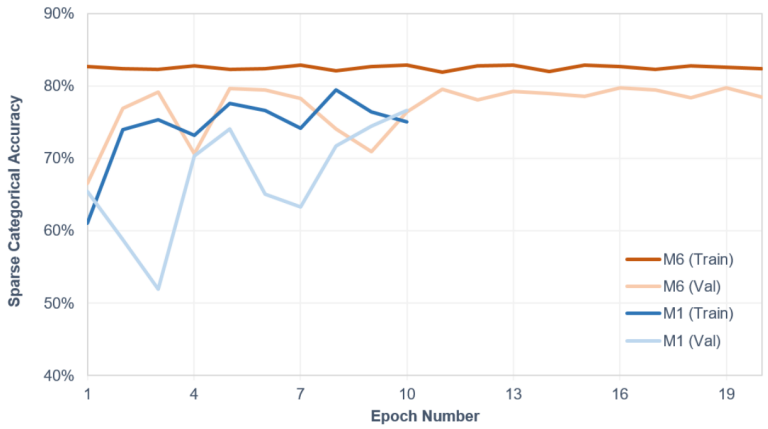

Figure 4: Comparison of M1 (baseline) and M6 (final model) using training and validation accuracy over 20 epochs of training.

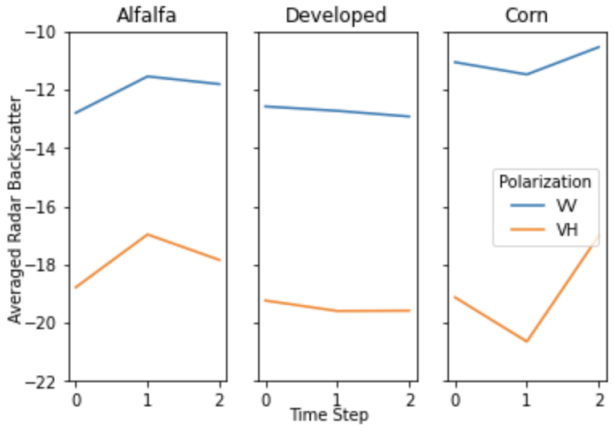

Figure 5: Averaged backscatter values for the two polarizations for 3 classes. We observe that alfalfa does not exhibit the increasing pattern we might expect from crops, perhaps explaining some of why it is poorly identified by the model.