Stalking Farms: Predicting Agricultural Yield with Satellite Imagery

Suhas Chundi
Stanford
chundi72@stanford.edu

Peter Ming
Stanford
peteming@stanford.edu

Abstract

Monthly projections of crop yield in the United States are published by the US Department of Agriculture (USDA) using surveys, ground level measurements, and weather forecasts. These projects can assist farmers and governments in setting prices and managing imports and exports. However, these measurements are not available in low resource areas, which hinders global agricultural forecasting. Additionally, monitoring global agricultural yield is a critical component of developing strategies to combat food shortages and recognize trends in land management. High resolution satellite imagery has made recent advances in frequency and quality in recent decades. We propose predicting crop yields from satellite imagery data. We describe a method to preprocess and construct simple features from satellite data which can be used to train regression models. We also discuss several challenges due to predicting sparse data with high dimensional inputs.

1. Introduction

Scientists predict the combined effects of climate change and a rising population will continue to make food insecurity a growing issue [3]. Additionally, these effects will be most severely felt in developing nations, where regular monitoring of crop yields is not in place. Remote sensing using high resolution satellite data has already been deployed for monitoring deforestation, methane gas emission, and a other geological applications [1]. In this paper, we propose using publically available satellite data to monitor and predict crop yields. We use ML methods to produce estimates of yield, and train models using the World Agricultural Supply and Demand Estimate (WASDE) report produced by the USDA. Satellite data can be noisy and high dimensional, so we discuss a few preprocessing methods to condense this data. These models may then be applied to low resource areas to provide rough yield estimates to farmers and food distributors.

2. Literature Review

2.1. Remote Sensing

Previous work has explored using satellite imagery to understand supply and demand of commodities. Piette et. al regress satellite data against World Agricultural Supply and Demand Estimate (WASDE) reports, however their work is limited to linear regression on a single commodity (corn) and a single satellite band. They regress using the NDVI index, a normalized difference between red and near infrared reflection[10]. Remote sensing has improved significantly in recent years. A number of works use satellite data as input to machine learning algorithms to monitor geological processes. Examples include monitoring changes in forest biomass[1] and understanding how topography changes affect likelihood of landslides [13]. Yu et. al even used daily satellite imagery to identify and invest in emerging foreign markets by monitoring port activity[12].

While satellite imagery has the ability to monitor global changes almost in real time, there are several challenges in implementation that these works reference. Satellite data often requires supervised labels to train sophisticated models. Obtaining these labels over large regions is costly and inefficient. The quantity of training data is also limited by the frequency of labelling. Prior work has proposed using weakly supervised labels, such as commodity futures prices [10][12] to obtain labels with high frequency. While a few papers seek to estimate crop yield using satellite data, there are several gaps in the current literature that we hope to address with this work.

1. Some papers supplement their feature space with ground level measurements such as CO_2 concentration [8]. Ground level measurements are difficult to obtain over large areas with poor infrastructure. We seek to estimate yield using only publically available satellite data.

2. Piette et. al limited their scope to linear regression and single feature. We hope to address the same problem using a few more satellite derived features, for multiple commodities, and using more sophisticated algo-

rithms.

2.2. Algorithms

Papers using satellite data may infer trends from reflectance levels without any regression or classification. E.g. Hicks et. al monitor the presence of petroleum products in sea ice exclusively through reflectance of certain wavelengths[5]. Other papers implement simple regression models because there is often limited data, and simpler models have more interpretability. Examples include the support vector machine (SVM), random forest, multilayer perceptron (MLP), and linear regression [11][13] [8].

2.2.1 Graph Convolutional Networks

The input to these algorithms is mean reflectance of specific wavelength bands within a region of interest. An average over a region neglects the spatial component of satellite data. Prior work generally assumes this spatial information is unimportant. E.g. if one is interested in the total agricultural yield of a region, the performance of specific crop fields is unimportant, so long as a good aggregate estimate is formed. Another reason for this aggregation may be the irregular shape of satellite data. The set of satellite pixels corresponding to a specific commodity is an irregular shape and does not lend itself to CNNs or other classic computer vision architectures that derive features using spatial information. Graph Neural Networks (GNNs) have the ability to digest points directly [9]. Recent approaches construct a K-nearest neighbor (KNN) graph from a point cloud and use GNNs to conduct inference on the resulting input [9]. This method achieves gains over discretization methods by directly incorporating point geometries. More details of this technique are presented in the methods section.

3. Dataset

We regress our satellite data against yield projections given by the WASDE report as in [10]. Projections are given each month for the total quantity of corn, cotton, soy, and wheat that will be produced in the subsequent year. These projects are determined with surveys and extensive ground level measurements. Our dataset includes data since March 2013. We collect LandSat satellite data from this year to the present. Each pixel contains measurements for a 30m × 30m area. Seven reflectance measurements, corresponding to a specific wavelength range, are taken for each pixel. In order to filter satellite images for clouds and missing band measurements, we take the median of all images over a month (LandSat visits the continental US approximately once every week). Additionally, non-agricultural locations must be masked out to include only relevant pixels. This yields a noncontinuous irregular image shape, presented in Figure 1.

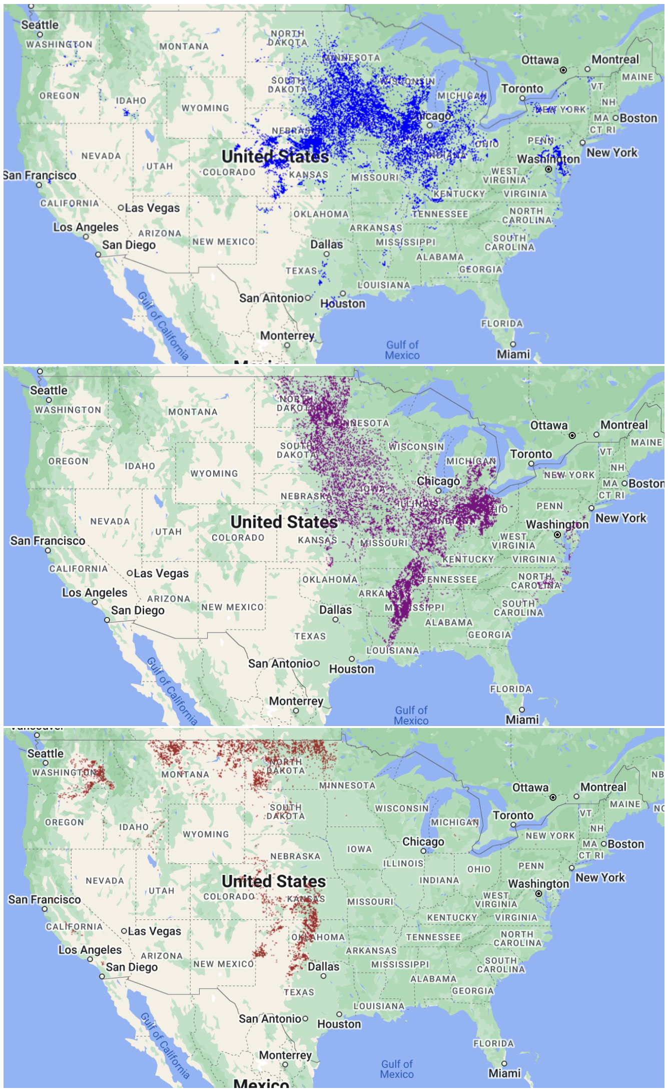

Figure 1. US corn (top) and soybean (middle) and wheat (bottom) fields for 2021. Note the irregular shape which does not easily lend itself to CNNs or other computer vision models. Additionally, there are a different number of active pixels for each commodity.

3.1. GCN Feature Space

Using this data, we create a feature space that can be an input to a graph convolutional network. An directed edge is constructed between each point and its $K = 5$ nearest neighbors.

3.2. Simple Regression Feature Space

For our simpler regression models, we take 10^{th}, 50^{th}, and 90^{th} percentiles for each feature across all pixels. This yields a total of $3 \times 7 = 21$ monthly features. With this transform, we provide a low dimensional approximation of the distribution of band measurements that can accommodate any number of points.

4. Methods

4.1. Graph Neural Network

We implement a graph convolutional network (GCN) to predict aggregate yield given the input graph. The GCN is implemented in pytorch geometric [2] and described in [7]. The convolution operation of a GCN can be described as

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

where:

- $H^{(l)}$ is the matrix of activations in the l-th layer $H^{(0)}$ is the reflectance features for each point.
- $W^{(l)}$ is the trainable weight matrix for the l-th layer.
- $\tilde{A} = A + I$ is the adjacency matrix with added self-loops.
- $\tilde{D}$ is the degree matrix of $\tilde{A}$.

The described operation computes the activation for node i with a linear layer over all nodes j. The mean output over all nodes such that (i, j) is an edge in the graph is passed through a nonlinearity σ (in this case ReLU) and is the next layer's activation for node i. Self loops are added to the adjacency matrix to allow the GCN to "remember" activations of previous layers. Note GCNs are generalizations of CNNs. By constructing appropriate edges between adjacent pixels or activations, one can recreate a CNN using a GCN architecture.

For our GCN architecture, we stack 3 GCN layers with a hidden dimension $h = 32$. Mean pooling and a linear layer is used to compute the final output. We chose a small GCN architecture to minimize overfitting on the training dataset, and selected hidden dimension size and number of hidden layers using cross validation. Our datasets consisted of a $103 - 15 - 10$ train-validation-test split. The train and validation datasets are chronologically before the test dataset, as we are interested in seeing if a model trained on previous data can generalize well to the current time period. Training was done using the Adam optimizer [6] with learning rate $\alpha = 0.001$ and for 80 epochs.

4.2. Baseline Methods

We implement a three simpler regression models (random forest, SVM, and linear regression) that do not use deep learning. These models take in the features described in section 3.2. Random Forest models develop a collection decision trees using subsets of the training data. Each decision tree is constructed using the split that minimized mean squared error. This produces trees with high variance, but the variance of the overall random forest is reduced due to the averaging of individual predictors. The number of decision trees was selected using K fold cross validation with $K = 10$. Support vector machines form regression as a convex optimization problem[4]. The objective is to find a hyperplane with minimal $L2$ norm such that residuals between the hyperplane and training data is ϵ. Support vector regression can also quickly compute high dimensional transforms of the feature space using the "kernel trick" [4]. The complete details of the SVM are outside of the scope of this paper, and will not be discussed in depth. Our SVM implementation uses the radial basis function as the kernel and selects the regularization hyperparameter C via cross validation. Finally, we implement linear regression with an $L2$ penalty (Ridge). The regularization term λ is selected using $K = 10$ fold cross validation. The loss function for Ridge regression can be written as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{P}||w_j||_2^2$$

where $y_i, \hat{y}_i$ represent the true and predicted outputs, and w is the vector of weights for the model.

Train Error for Regression Algorithms

Model	Corn	Cotton	Soybean	Wheat
Naive Mean	74.46	44.68	106.66	46.96
SVM	1.87	3.10	2.39	0.89
Linear Regression	2.95	11.78	7.41	6.31
Random forest	1.50	5.47	3.77	3.11
GNN	15.00	62.86	7.26	18.28

5. Results

Results for each model are presented in Table 1. Each crop has a different magnitude of yield. Corn averages approximately 300 million metric tons per year, whereas Cotton is on the order of 14 million metric tons per year. To account for this, we present results as percent errors:

$$\text{Mean Percent Error} = \frac{1}{N}\sum\frac{|y_i - \hat{y_i}|}{y_i} * 100\%$$

Where $y_i, \hat{y_i}$ represent the ground truth and predicted yields, respectively. Additionally, we include a naive method for comparison. For the naive method, we predict the yield for each year is the mean yield over the training set. We observe that all baseline models perform quite similarly, with linear regression outperforming the other models on two of the four tasks.

Test Error for Regression Algorithms

	Corn	Cotton	Soybean	Wheat
Model				
Naive Mean	105.97	54.10	36.91	81.25
SVM	**3.60**	18.58	6.97	13.58
Linear Regression	4.19	**11.81**	6.25	**12.24**
Random forest	4.61	17.93	**6.03**	12.70
GNN	1211	9512	1013	1375

Table 1. The average relative error of regression algorithms on the train set (top) and test set (bottom). The best results are shown in bold. We observe Linear Regression with a $L2$ penalty has the least tendency to overfit. All values are presented are percentages.

5.1. GNN performance

The GNN performs poorly on all tasks, and fails to fit even the training dataset well. One hypothesis for why this might be the case is the GCN architecture includes an unnecessary inductive bias. Graph convolutional networks assume there is information that needs to be aggregated along edges in the graph, i.e. that edge relationships encode some meaning. In our model, we construct a KNN Graph, encoding the assumption that the measurements of nearby farms is useful in predicting the productivity of a portion of land. This may be valid for example, if poor performance in one patch of land indicates a blight or pest problem. However, it appears this assumption does not necessarily hold in our remote sensing problem. Total agricultural yield is likely independent of the arrangement of farms. I.e. the productivity of some land is not affected by the productivity of adjacent land.

5.2. Baseline Method Performance

Each baseline method outperforms the naive mean method, suggesting the input features are valuable in determining crop yield. Figure 2 presents a time series plot of predicted crop yields. While the test fit is not as accurate as the train fit, we observe there may be an ability to predict trend in agricultural returns. Models generally perform best on Corn, then Soybean, Wheat, and Cotton. This also describes the number of pixels in satellite data that correspond to a particular commodity. I.e., corn is planted over the greatest area, and cotton the least. Our decision to embed satellite images based on percentile measurements is dependent on somewhat consistent distributions of band measurements, which is problematic with a small number of measurements.

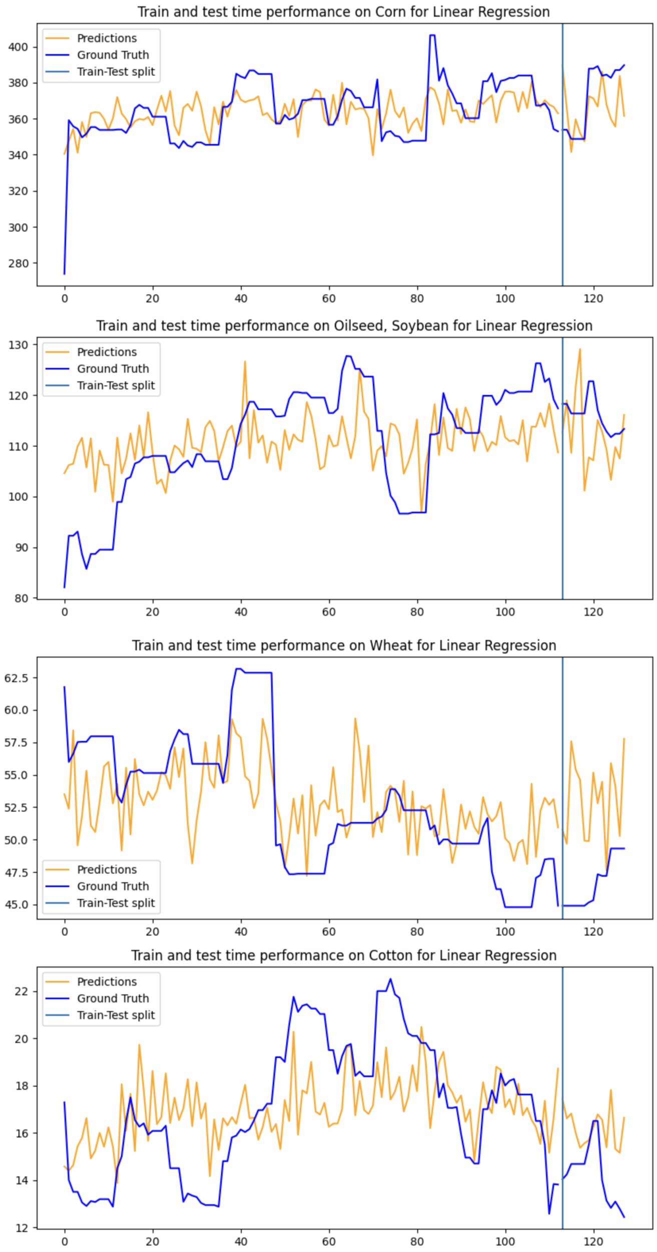

Figure 2. Linear Regression performance at train and test time. The y axis is measured million metric tons of commodity produced, and the x axis is the timestep for which the prediction was made. Note the linear regression model even picks up the uptick in yield in the test set for corn, even though it is slightly translated. Other instances of "predicting upticks" are present, though more work needs to be done to determine if this is can be predicted consistently.

6. Analysis of Weights

Because the linear regression model performs well on this task, we may learn what features are relevant by examining the weights of the trained model. Weights for the lin-

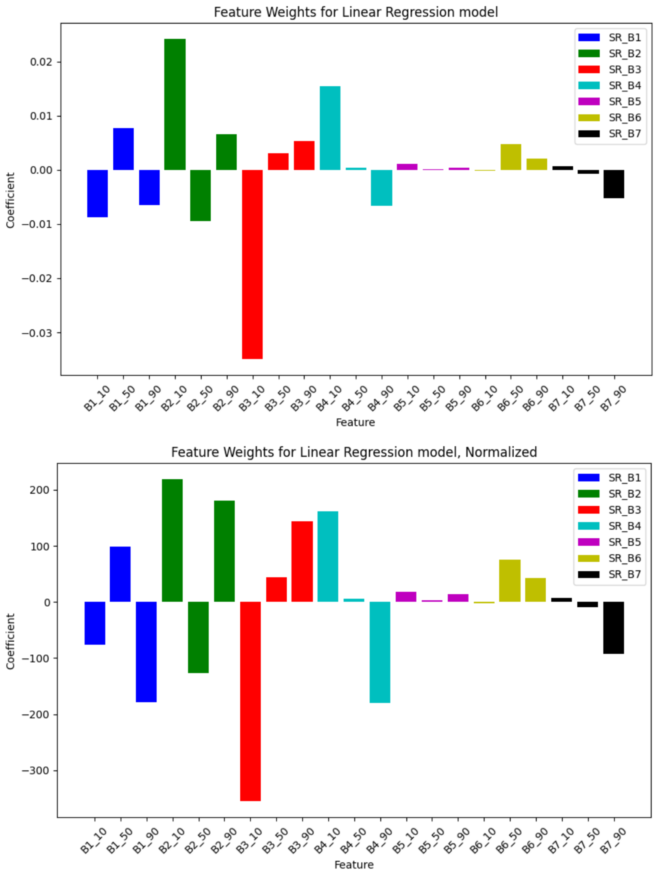

Figure 3. Linear Regression weights by input feature, trained on the corn dataset. Recall the features taken are the 10^{th}, 50^{th} and 90^{th} percentiles of each band measurement for all pixels corresponding to a commodity. From left to right, each set of three bars represent percentile measurements for blue (moisture), blue, green, red, near infrared, short wave infrared, and medium wave infrared wavelengths respectively.

ear regression model trained for the corn data are available in Figure 3. Because the magnitude of feature determines the magnitude of linear regression coefficient, we additionally include a plot of weights multiplied by average magnitude of the corresponding feature. Bands $1-7$ correspond to ultra blue (moisture), blue, green, red, near infrared, short wave infrared, and medium wave infrared wavelengths respectively. The bar plots illustrate that multispectral measurements may be useful in predicting yield, as opposed to prior work which regressed on single bands or a single band derived feature (e.g. NDVI) [10]. However, we do not reproduce these methods for our dataset, so cannot say to what extent multiband inference outperforms single feature regression.

7. Conclusion

We present a simple method for predicting crop yields from publically available satellite data. This yield information can be used for mitigating food shortages, commodities trading, and developing food distribution policies. We present a method to apply deep learning to noncontinuous satellite pixels that develops intermediate features through spatial information. This method introduces inductive bias that is likely unproductive for the intended regression. However, there is likely a simple relationship between our input features and regression target. As a result, linear regression, SVM and random forest regression show promising results on this task. We note our analysis is limited by the quantity of data available for this problem. Future work may evaluate these methods on European or Asian yield estimations.

7.1. Contributions and Acknowledgements

Contributions: Peter and Suhas equally contributed to both code and technical writing. Peter was more involved in accumulating the WASDE data, while Suhas obtained the LandSAT data. Both team members felt there was an appropriate split of work for this project.

Sharing project with MS&E 244: Suhas' MS&E 244 project involves deriving a trading strategy for agricultural commodities using WASDE data. We may also include, in that project, the results of predicting WASDE report information, and then trading based on predicted WASDE information. The bulk of our MS&E 244 project will consist of describing the methods used to trade based on WASDE information, and the satellite portion may be included as an interesting aside. The final report in MS&E 244 is due on June 12th, and as a result is incomplete. However, our current progress can be found here.

References

[1] W. Dong, E. T. A. Mitchard, Y. Chen, M. Chen, C. Cao, P. Hu, C. Xu, and S. Hancock. Comparing remote sensing-based forest biomass mapping approaches using new forest inventory plots in contrasting forests in northeastern and southwestern china, 2024. 1

[2] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric, 2019. cite arxiv:1903.02428. 3

[3] H. C. J. Godfray, J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, J. Pretty, S. Robinson, S. M. Thomas, and C. Toulmin. Food security: The challenge of feeding 9 billion people. *Science*, 327(5967):812–818, 2010. 1

[4] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. 3

[5] L. Hicks, M. Zabihi Mayvan, E. Asihene, D. S. Desmond, K. Polcwiartek, G. A. Stern, and D. Isleifson. Assessment of c-band polarimetric radar for the detection of diesel fuel in newly formed sea ice. *Remote Sensing*, 16(11), 2024. 2

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. 3

[7] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2017. 3

[8] O. Kira, J. Wen, J. Han, A. J. McDonald, C. B. Barrett, A. Ortiz-Bobea, Y. Liu, L. You, N. D. Mueller, and Y. Sun. A scalable crop yield estimation framework based on remote sensing of solar-induced chlorophyll fluorescence (sif). *Environmental Research Letters*, 19(4):044071, apr 2024. 1, 2

[9] D. Parikh, B. Zhang, R. Kannan, V. Prasanna, and C. Busart. Performance of graph neural networks for point cloud applications. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7, 2023. 2

[10] P. Piette. Can Satellite Data Forecast Valuable Information from USDA Reports ? Evidences on Corn Yield Estimates, June 2019. 1, 2, 5

[11] Y.-L. Tsai, J. Irvin, S. Chundi, A. Y. Ng, C. B. Field, and P. K. Kitanidis. Improving debris flow evacuation alerts in taiwan using machine learning, 2022. 2

[12] H. Yu, X. Hao, L. Wu, Y. Zhao, and Y. Wang. Eye in outer space: satellite imageries of container ports can predict world stock returns. *Palgrave Communications*, 10(1):1–16, December 2023. 1

[13] Y. Zhang, T. Ge, W. Tian, and Y.-A. Liou. Debris flow susceptibility mapping using machine-learning techniques in shigatse area, china. *Remote Sensing*, 11:2801, 11 2019. 1, 2