Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data

Jiaxuan You and **Xiaocheng Li** and **Melvin Low** and **David Lobell** and **Stefano Ermon**
Department of Computer Science, Stanford University
{jiaxuan, mwlow, ermon}@cs.stanford.edu
Department of Management Science and Engineering, Stanford University
chengli1@stanford.edu
Department of Earth System Science, Stanford University
dlobell@stanford.edu

Abstract

Agricultural monitoring, especially in developing countries, can help prevent famine and support humanitarian efforts. A central challenge is yield estimation, i.e., predicting crop yields before harvest.

We introduce a scalable, accurate, and inexpensive method to predict crop yields using publicly available remote sensing data. Our approach improves existing techniques in three ways. First, we forego hand-crafted features traditionally used in the remote sensing community and propose an approach based on modern representation learning ideas. We also introduce a novel dimensionality reduction technique that allows us to train a Convolutional Neural Network or Long-short Term Memory network and automatically learn useful features even when labeled training data are scarce. Finally, we incorporate a Gaussian Process component to explicitly model the spatio-temporal structure of the data and further improve accuracy. We evaluate our approach on county-level soybean yield prediction in the U.S. and show that it outperforms competing techniques.

Introduction

It is estimated that 795 million people still live without an adequate food supply (FAO 2015), and that by 2050 there will be two billion more people to feed (Dodds and Bartram 2016). Ending hunger and improving food security are primary goals in the 2030 Agenda for Sustainable Development of the United Nations (United Nations 2015).

A central challenge of addressing food security issues is yield estimation, namely being able to accurately predict crop yields well before harvest. Agricultural monitoring, especially in developing countries, can improve food production and support humanitarian efforts in light of climate change and droughts (Dodds and Bartram 2016).

Existing approaches rely on survey data and other variables related to crop growth (such as weather and soil properties) to model crop yield. These approaches are very successful in the United States, where data are plentiful and of relatively high quality. Comprehensive surveys of weather parameters such as the Daymet (Thornton et al. 2014) and land cover types such as the Cropland Data Layer (Boryan et al. 2011) are publicly available and greatly facilitate the crop yield prediction task. However, information about weather, soil properties, and precise land cover data are typically not available in developing countries, where reliable yield predictions are most needed.

Remote sensing data, on the other hand, are globally available and relatively inexpensive. It is frequently used in computational sustainability applications, such as species distribution modeling (Fink, Damoulas, and Dave 2013), poverty mapping (Xie et al. 2016; Jean et al. 2016; Ermon et al. 2015), climate modeling (Ristovski et al. 2013), and natural disaster prevention (Boulton, Shotton, and Williams 2016). Multi-spectral satellite images, which include information in addition to the visible wavelengths (RGB), have fairly high spatial and temporal resolution, and contain a wealth of information on vegetation growth and thus on agricultural outcomes. However, useful features are hard to extract since the data are high-dimensional and unstructured.

In this paper, we propose an approach based on modern representation learning ideas, which have recently led to massive improvements in a range of computer vision tasks (Krizhevsky, Sutskever, and Hinton 2012; Karpathy et al. 2014). We overcome the scarcity of training data by employing a new dimensionality reduction technique. Specifically, we treat raw images as histograms of pixel counts, and use a mean-field approximation to achieve tractability. Deep learning architectures, including CNNs and LSTMs, are then trained on these histograms to predict crop yields. While this approach performs well, it does not explicitly account for spatio-temporal dependencies between data points, e.g., due to common soil properties. We overcome this limitation by incorporating a Gaussian Process layer on top of our neural network models. We evaluate our approach on the task of predicting county-level soybean yield in the United States. Experimental results show that our model outperforms traditional remote-sensing based methods by 30% in terms of Root Mean Squared Error (RMSE), and USDA national-level estimates by 15% in terms of Mean Absolute Percentage Error (MAPE).

Related Work

Remote sensing data have been widely used for predicting crop yields in the remote sensing community (Bolton and Friedl 2013; Johnson 2014). However, all existing approaches we are aware of rely on hand-crafted features,

which are believed to compactly summarize most of the information related to vegetation growth contained in the raw (multi-spectral) images. Some widely used features include Normalized Difference Vegetation Index (NDVI) (Quarmby et al. 1993; Johnson 2014), two-band Enhanced Vegetation Index (EVI2) (Bolton and Friedl 2013), and Normalized Difference Water Index (NDWI) (Satir and Berberoglu 2016). While significant effort has been devoted to feature engineering, existing features are fairly crude indexes that depend on a small number (usually two) of the available bands. Inspired by recent successes in computer vision and speech recognition and in contrast to existing approaches, we propose the use of modern representation learning ideas from AI to automatically discover relevant features from the raw data. Our experimental results suggest that our learned features are much more effective, and that bands that are typically ignored could play an important role.

Second, high-order moments of the features are rarely explored in existing approaches. In most settings, ground truth average yield data are provided over a region as the regression output, while features are given as input for all the locations within that region. Most approaches either calculate the mean (first moment) of the features over the region of interest (Johnson 2014) or do sampling (Kuwata and Shibasaki 2015). In contrast, our model works directly with the entire *pixel distribution* over a region. Using a mean-field assumption to achieve tractability, we are able to learn features from the transformed normalized histograms.

Third, most previous studies assume that crop yields are mutually independent and identically distributed over space and time. Therefore, crop yields are predicted with a regression model separately for each location (Bolton and Friedl 2013; Johnson 2014). However, spatial and temporal correlations that are not explained by the available covariates are likely to be presented (e.g., due to soil properties). Thus, we propose the use of a Gaussian Process (GP) layer on top of our neural architectures to explicitly account for spatial and temporal dependencies across data points.

Preliminaries

We start by reviewing the building blocks of our model, then elaborate on our approach.

Deep Learning Models

Deep learning models can be viewed as complex non-linear mappings that can learn hierarchical representations of the data. Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long-short Term Memory (LSTM) networks are some typical architectures (LeCun, Bengio, and Hinton 2015). They are typically composed of a set of layers such that the output of one layer is the input of the next. CNNs and LSTMs are used in our proposed model.

A DNN is the basic form of feed-forward neural network, built using fully connected layers. Each fully connected layer takes a vector $\boldsymbol{x} \in \mathbb{R}^n$ as input followed by a non-linear function $f(\cdot)$ (usually a rectified linear unit (ReLU) or tanh) and finally outputs a vector $\boldsymbol{c} \in \mathbb{R}^{\hat{n}}$ such that

$$\boldsymbol{c} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}),$$

where $\boldsymbol{W} \in \mathbb{R}^{\hat{n} \times n}$ is the weight matrix and $\boldsymbol{b} \in \mathbb{R}^{\hat{n}}$ the bias.

A CNN is mainly composed of three types of layers: convolutional, pooling, and fully connected. Its convolutional layer shares weights across the first two input dimensions (2-d convolution) and thus greatly reduces the number of parameters. A convolutional layer takes a tensor $\boldsymbol{x} \in \mathbb{R}^{h \times w \times d}$ as input, followed by a nonlinear function (usually ReLU) and sometimes a pooling layer (usually max-pooling), and outputs a tensor $\boldsymbol{c} \in \mathbb{R}^{\hat{h} \times \hat{w} \times \hat{d}}$ given by

$$\boldsymbol{c} = p\left(f(\boldsymbol{W} * \boldsymbol{x} + \boldsymbol{b})\right),$$

where $p(\cdot)$ is the pooling function, $f(\cdot)$ is the nonlinear function, $\boldsymbol{W} \in \mathbb{R}^{l \times l \times \hat{d}}$ is a weight matrix defining a convolutional filter, "$*$" is a 2-dimensional convolution operator over dimensions h and w, and $\boldsymbol{b} \in \mathbb{R}^{\hat{h} \times \hat{w} \times \hat{d}}$ is the bias term.

An LSTM is a special type of Recursive Neural Network (RNN) that takes sequential data as input (Hochreiter and Schmidhuber 1997). For each time step t, it maintains a hidden state vector $\boldsymbol{h}_t$ that depends on the previous state $\boldsymbol{h}_{t-1}$, and provides an output $\boldsymbol{o}_t$ that is a function of the hidden state $\boldsymbol{h}_t$. The mappings from $\boldsymbol{h}_{t-1}$ to $\boldsymbol{h}_t$, usually encoded as LSTM cells, and the mappings from $\boldsymbol{h}_t$ to $\boldsymbol{o}_t$, usually represented as fully connected layers, share parameters across all time steps.

Gaussian Process Modeling

A Gaussian Process (GP) is a non-parametric probabilistic model defined as a collection of random variables $\{f(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathcal{X}}$, for which any finite subset has a joint Gaussian distribution (Rasmussen 2006), denoted as

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right),$$

where the mean function $m(\boldsymbol{x})$ represents the expectation $E[f(\boldsymbol{x})]$ and the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ defines the covariances $\text{cov}(f(\boldsymbol{x}), f(\boldsymbol{x}'))$.

In this paper, we use a linear GP model defined as

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{\beta},$$

where $\boldsymbol{x} \in R^d$, $f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$ is a zero-mean GP modeling the residuals of a linear model, $\boldsymbol{h}(\cdot)$ is a fixed set of basis functions, and $\boldsymbol{\beta}$ is an independent random variable with Gaussian prior $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B})$. For the kernel function, the squared exponential kernel is commonly used,

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2r^2}\right),$$

where $\|\cdot\|_2$ denotes the L^2 norm, σ and r are hyperparameters for the kernel. Since we only have access to noisy observations in practice, we add an extra Gaussian noise term (with variance σ_e^2) to the covariance

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_{SE}(\boldsymbol{x}, \boldsymbol{x}') + \sigma_e^2 \delta_{\boldsymbol{x}, \boldsymbol{x}'},$$

where $\delta_{\boldsymbol{x}, \boldsymbol{x}'}$ is the Kronecker delta. See (Rasmussen 2006) for more details on the linear GP model.

Proposed Approach

Problem Setting

We consider the problem of predicting the average yield of a type of crop (e.g., soybean) for a region of interest based on a sequence of remotely sensed images taken before the harvest. Specifically, we are interested in the average yield per unit area in a given geographical region, e.g., a county or district. As input, we are given a sequence of multispectral images $(\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(T)})$ covering the area of interest. Each multispectral image $\boldsymbol{I}^{(t)}$ corresponds to a different time t within a year, and is a tensor $\boldsymbol{I}^{(t)} \in \mathbb{R}^{l \times w \times d}$, where l, w are the number of horizontal and vertical pixels, and d is the number of bands per pixel. Note that a general "crop mask" identifying pixels corresponding to farmland is available worldwide at 500 m resolution (DAAC 2015). While we can mask out pixels that do not correspond to farmland, we do not generally know which pixels correspond to the particular crop we are targeting (e.g., soybeans).

Our goal is to learn a model that maps these raw image sequences to the average crop yield. Intuitively, this is possible since factors related to plant growth are captured in the images (Lobell et al. 2015). As training data, we are given a set

$$D = \left\{ \left((\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(T)}, \boldsymbol{g}_{\text{loc}}, g_{\text{year}})_1, y_1 \right), \cdots, \left((\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(T)}, \boldsymbol{g}_{\text{loc}}, g_{\text{year}})_N, y_N \right) \right\}$$

of image sequences, geographic locations $\boldsymbol{g}_{\text{loc}}$, years g_{year}, and corresponding ground truth crop yields $y_i \in \mathbb{R}^+$. We will also consider the (harder) problem of making real-time predictions based on sub-sequences $(\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(t)})$ for $t < T$. This corresponds to the problem of forecasting the yield before the harvest date in an online manner, when only a subset of the remotely sensed data are available.

From Raw Images to Histograms

Given the scarcity of labeled training data ($|D|$ can be less than $10,000$), directly training a deep model end-to-end is not feasible. Pre-training on popular benchmarks from computer vision like Imagenet is also not appropriate, because remotely sensed images are multi-spectral and taken from a bird's eye viewpoint. We therefore designed a dimensionality reduction technique under the assumption of *permutation invariance*. Our approach is based on the following intuition: we don't expect the average yield to depend (much) on the *position* of the image pixels, since they merely indicate the locations of the cropland. While some dependence on the position is possible (e.g., due to soil properties or elevation), to achieve tractability we ignore these dependencies.

Assuming permutation invariance holds, only the *number* of different pixel types in an image (pixel counts) are informative. In other words, there is no loss of information in mapping the high-dimensional image into a histogram of pixel counts [1]. Assuming pixel values in digital images are discrete and can take up to b different values per band, the resulting histogram would have b^d bins, which might not be practical (e.g., in our application each band intensity can take $b = 256$ different values, and $d = 9$). Therefore, we separately consider each band $\boldsymbol{I}_k$ in an image $\boldsymbol{I}^{(t)}$ where index t is omitted for notational simplicity, discretize the pixel values into b bins and produce an histogram $\boldsymbol{h}_k \in \mathbb{R}^b$ for each individual band $k = 1, \cdots, d$. By concatenating all $\boldsymbol{h}_k$ into $\boldsymbol{H} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_d)$, we obtain a compact representation of the original multi-spectral image. By treating each band independently, we are implicitly making a mean-field assumption (Parisi 1988), i.e., we are assuming that the (normalized) histogram of a multi-spectral image I can be approximated as a product of simpler (normalized) histograms $\boldsymbol{h}_i$ over individual bands.

From Histograms to Crop Yield

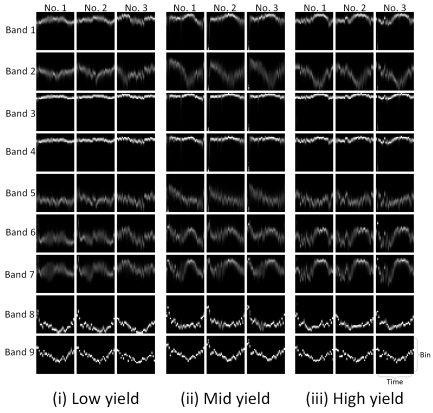
While the histogram approach outlined in the previous section can drastically reduce the dimensionality on the input data, the desired mapping $(\boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(T)}) \mapsto y$, is still highly non-linear and complex. Rather than hand-crafting features, we leverage ideas from representation learning and use deep models to automatically learn relevant features from data.

The sequential nature of the inputs $(\boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(T)})$ suggests the use of temporal models, such as LSTMs. We use an LSTM architecture that takes sequences of vectors as input, and add a fully connected layer on the last LSTM cell to finally yield the prediction y corresponding to the input sequence, as is shown in Figure 1b. To fit the model, we first flatten each histogram $\boldsymbol{H}^{(t)} \in \mathbb{R}^{b \times d}$ into a vector $\boldsymbol{S}^{(t)} \in \mathbb{R}^r$, $r = b \times d$, then feed the sequence $(\boldsymbol{S}^{(1)}, \cdots, \boldsymbol{S}^{(T)})$ into the network. L2 loss is used for the regression task. To prevent overfitting, we regularized the network by adding a dropout layer with dropout rate 0.75 after each state transition.

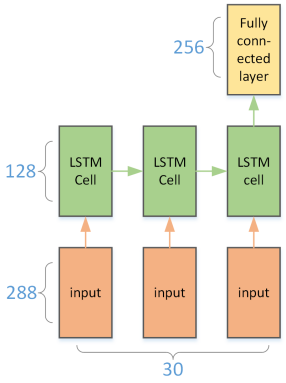
Inspired by the success of CNN architectures on sequential data (Karpathy et al. 2014), we also use a CNN architecture to model the non-linear mapping. We stack $(\boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(T)})$ into a 3-D histogram $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{b \times T \times d}$, where $\boldsymbol{H}^{(t)}$ is the t^{th} component in the second dimension of $\boldsymbol{\mathcal{H}}$. We feed the 3-D histograms as input to the CNN, and the convolution operation is performed over the "bin" and "time" dimensions. Some typical 3-D histograms are shown in Figure 1a. The visualization exhibits distinct visual patterns corresponding to different crop yield conditions (high vs low yield), indicating that our CNN might be able to learn useful features.

The structure of our CNN model is shown in Figure 1c. We note that in our case we don't want the location invariance property given by the pooling layer (LeCun, Bengio, and Hinton 2015), since different locations in the histogram have different physical meanings. We solve the problem by replacing the pooling layer with a stride-2 convolutional layer to reduce the size of the intermediate feature maps. We

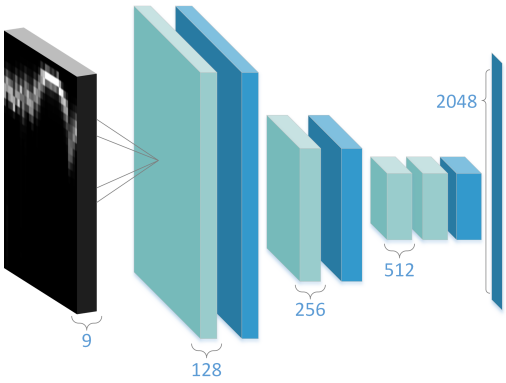
[1] Given the pixel counts from a histogram, one can reconstruct an image equivalent under the permutation invariance assumption by arbitrarily placing the pixels.



(a) 3-D histogram visualization



(b) The LSTM structure



(c) The CNN structure

Figure 1: Visualization of the input data and used architectures. **Left:** Figures of typical 3-D histograms $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{b \times T \times d}$ flattened in the band dimension d under (i) low crop yield, (ii) mid crop yield, and (iii) high crop yield conditions are shown in the left panel. Each row corresponds to a different spectral band, while each column represents an individual data point. Each square is a slice of $\boldsymbol{\mathcal{H}}$, where the x-axis corresponds to the "time" dimension T, and the y-axis to the "bin" dimension b. Brighter pixels indicate higher pixel counts in that bin. There exists distinctive visual differences between high yield and low yield conditions (for example in the second and the seventh bands). **Mid:** The adopted LSTM structure. **Right:** The adopted CNN structure, where stride-1 convolutional layers are in light blue, stride-2 convolutional layers are in dark blue and a fully connected layer is attached at the end.

use batch normalization to facilitate gradient flow (Ioffe and Szegedy 2015), and dropout with rate 0.5 to prevent overfitting after each convolutional layer.

Integrating the Spatio-temporal Information: Deep Gaussian Process

There are many features relevant to crop growth that are not revealed in remote sensing images, such as the soil type, fertilizer rate, etc. These features could be inherent to specific locations (e.g., soil type) and may not change significantly over time, and thus could exhibit spatial and temporal patterns. To illustrate this point, we draw a variogram (Cressie and Hawkins 1980) on the absolute prediction error of the CNN model introduced in the previous section (trained on the data described in the Experimental section below) in Figure 2. A variogram illustrates the variance across data points as a function of their geographical distance. The result shows that the errors corresponding to data points that are spatially closer tend to vary less (lower variance). Therefore, it suggests that we can reduce the error by incorporating a Gaussian Process model on top of the deep models previously described (Hinton and Salakhutdinov 2008; Wilson et al. 2015).

The analysis above indicates that the errors could correlate with each other spatially and temporally. This motivates us to design a linear Gaussian Process model where the mean function is linear with respect to the deep features, i.e., the last layer's input in our architectures, and the covariance kernel depends on the spatio-temporal structure. More concretely, let $\boldsymbol{x} = (\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(T)}, \boldsymbol{g}_{\text{loc}}, g_{\text{year}})$ denote an original data point, $\boldsymbol{h}(\boldsymbol{x})$ denote the feature vector extracted from the deep models based on $(\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(T)})$, and

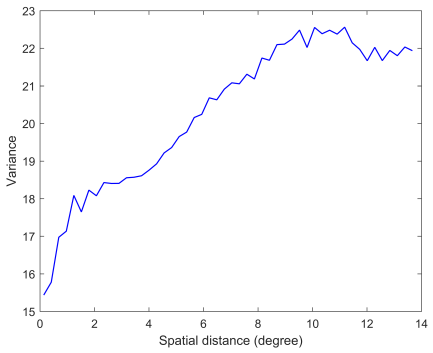


Figure 2: A variogram on the absolute prediction error of the proposed CNN model.

$\boldsymbol{g} = (\boldsymbol{g}_{\text{loc}}, g_{\text{year}})$. Then in our Deep Gaussian Process model we have

$$y(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^T \boldsymbol{\beta}, \text{where } f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}')),$$

and $\boldsymbol{h}(\cdot)$ is a set of basis functions corresponding to the final layer in our deep models, $\boldsymbol{\beta}$ follows a Gaussian prior $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B})$, and the kernel function is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left[-\frac{\|\boldsymbol{g}_{\text{loc}} - \boldsymbol{g}'_{\text{loc}}\|_2^2}{2r_{\text{loc}}^2} - \frac{\|g_{\text{year}} - g'_{\text{year}}\|_2^2}{2r_{\text{year}}^2}\right] + \sigma_e^2 \delta_{\boldsymbol{g},\boldsymbol{g}'}.$$

We choose $\boldsymbol{b}$ as the weight vector in the last layer of our deep models and $\boldsymbol{B} = \sigma_b I$, while treat σ, σ_b, σ_e, r_{loc} and r_{year} as hyperparameters. During training, we conduct a

grid search for the optimal hyperparameter values based on cross-validation performance, using the closed-form expressions in (Rasmussen 2006).

Experiments

Data Description

To compare with prior work, we evaluate our model in the United States and choose soybean as the target crop since it has been widely investigated in prior work (Bolton and Friedl 2013; Johnson 2014).

The input data we use include remote sensing data on surface reflectance, land surface temperature, and land cover type derived from the MODIS satellite, which are available worldwide (DAAC 2015). We use multi-spectral images collected 30 times a year, from the 49^{th} day to the 281^{th} day at 8-days intervals. We discretize all the images using 32 bins to compute the pixel histograms. The resulting input histogram is $\boldsymbol{\mathcal{H}} = (\boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(T)})$, $\boldsymbol{H}^{(t)} \in \mathbb{R}^{b \times d}$ with $b = 32, d = 9$, and $T = 30$. The ground truth output data are the yearly average soybean yields at the county-level measured in bushels per acre, publicly available on the USDA website (USDA 2016).

We select 11 states in the U.S. that account for over 75% of the national soybean production and use data from 2003 to 2015, resulting in $|D| = 8945$ data points in total. All sources of remote sensing data are cropped according to county borders, while non-crop pixels are removed with the help of general world-wide land cover data (DAAC 2015). More details are provided in the appendix.

Competing Approaches

We compare our model with widely used crop yield prediction models. The baseline methods include ridge regression (Bolton and Friedl 2013), decision trees (Johnson 2014), and a DNN (Kuwata and Shibasaki 2015) with 3 hidden layers and 256 neurons each. Their input is a sequence of $T = 30$ average NDVI values for the region of interest. Each element of the sequence is computed by first averaging the corresponding image $\boldsymbol{I}^{(t)}$ across the region, and then calculating the NDVI value (which is a scalar). Note that traditionally precise pixel masks (e.g., soybean mask) are used to remove irrelevant pixels in input images and weather data are also used as input, but for comparison these models are provided with the same inputs as our proposed model, i.e., only remote sensing data. The hyperparameters in these models are optimized in cross-validation.

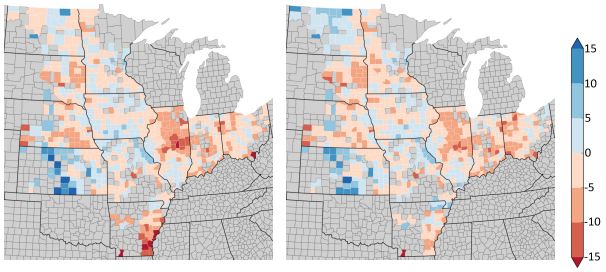
Results

We report the Root Mean Square Error (RMSE) of our county-level predictions in Table 1. The result is averaged over 2 runs to account for the random initialization and dropout during deep model training. Each row corresponds to predictions made for that year, using a model trained on data from all preceding years. Learning rates and stopping criteria are tuned on a held-out validation set (10%). Our results demonstrate that our CNN and LSTM approaches outperform competing methods significantly. By adding the GP component, our models achieve even better performance, with 30% reduction of RMSE from the best competing methods.

	Baselines			Deep models			
Year	Ridge	Tree	DNN	LSTM	LSTM + GP	CNN	CNN + GP
2011	9.00	7.98	9.97	5.83	5.77	5.76	**5.7**
2012	6.95	7.40	7.58	6.22	6.23	5.91	**5.68**
2013	7.31	8.13	9.20	6.39	5.96	**5.50**	5.83
2014	8.46	7.50	7.66	6.42	5.70	5.27	**4.89**
2015	8.10	7.64	7.19	6.47	**5.49**	6.40	5.67
Avg	7.96	7.73	8.32	6.27	5.83	5.77	**5.55**

Table 1: The RMSE of county-level model performance.

To show that the GP has the capability to reduce spatially correlated errors, we plot the prediction errors of the CNN model for year 2014 in Figure 3. As previously shown in the variogram of Figure 2, it is apparent that errors are spatially correlated (Where red means underpredicting and blue means overpredicting). After adding the GP component, the correlation is reduced. Intuitively, we believe the errors are due to properties that are not observable in remote sensing images (e.g., due to soil). The GP part learns these patterns from past training data and effectively corrects for them.



(a) CNN (b) CNN+GP

Figure 3: County-level error maps before and after adding the GP. The color represents the prediction error in bushel per acre.

Real-Time Prediction throughout the Year

In the U.S., soybean is often planted in May and June and harvested in October and November. Early crop yield predictions are essential for food security applications. To this end, we train and test our model on a sub-sequence of the input $(\boldsymbol{I}^{(1)}, \cdots, \boldsymbol{I}^{(t)})$ where $t < T$. Figure 4 shows the performance if we tried to predict the harvest each month in an online manner, given only the data available up to that point. We observe that none of the models perform well in early months, probably because there is not enough information yet on plant growth. But as we gather more information, all the models improve, and the gap between our models and competing approaches increases.

We further average our county-level predictions to compare with USDA annual yield estimates aggregated at the country level, in terms of Mean Absolute Percentage Error

	July	August		September		October	
	Ours	USDA	Ours	USDA	Ours	USDA	Ours
MAPE	5.65	3.92	**3.37**	4.14	**3.41**	**2.48**	3.19

Table 2: The MAPE of US-level model performance, averaged from 2009 to 2015.

(MAPE). Results show that our model outperforms USDA predictions by 15% on average in August and September. Note that USDA predictions are survey-based, while our techniques use cheap, passively collected data.

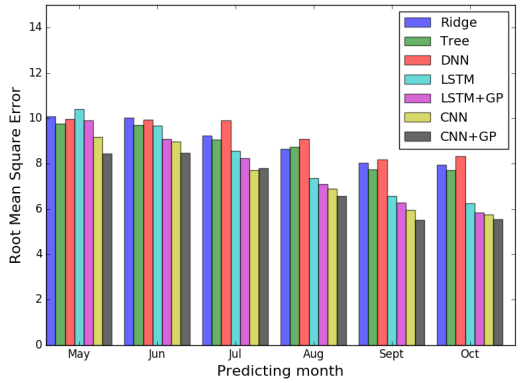


Figure 4: Model performance in each month measured in RMSE. The results are averaged from 2011 to 2015.

Understanding Feature Importance

To understand how our model is utilizing the input data, we provide an analysis inspired by the permutation test for random forests (Breiman 2001). More specifically, we consider the effect of randomly permuting the values of a specific feature over the entire data (without changing the other features). For our 3-D histogram input, we separately permute across time and band dimensions by shuffling a slice of the histogram across all the data, while holding the rest fixed. The average performances from 2011 to 2015 of the models trained on this perturbed data are shown in Figures 5 and 6.

The permutation test across bands in Figure 5 reveals two useful insights on the relative importance of different bands for yield prediction. Traditionally, band 2 as a near infrared band, is viewed as a key factor in revealing crop growth (Quarmby et al. 1993). While putting some emphasis on band 2, our model also focuses on band 7, a short-wave infrared band always ignored by traditional approaches. The high dependence on land surface temperature, shown as band 8 and 9, is also confirmed by previous work (Johnson 2014). Second, the importance of different bands varies over different phases in crop growth. Growth-related bands 2 and 7 are given higher relative importance in later months (when crop has grown), while temperature bands 8 and 9 are more significant in earlier months (when crop has not grown yet).

The permutation test across time in Figure 6 is also informative. Surveys show that soybean planting usually starts on

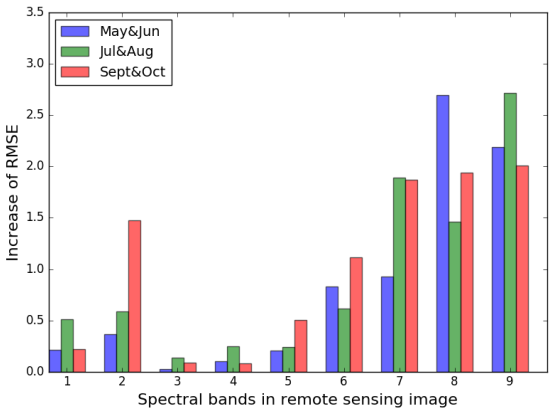


Figure 5: The increase of RMSE after permutation over bands. We evaluate predictions made in different months.

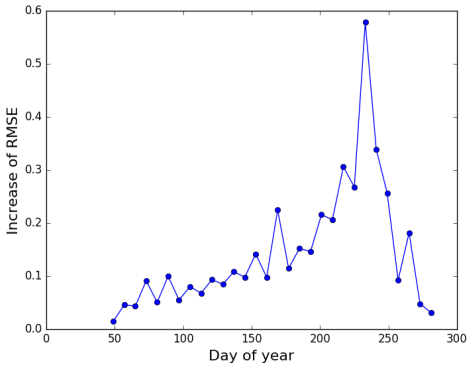


Figure 6: The increase of RMSE after permutation over time within a year. The model with complete data are used for evaluation.

day 110 and ends on day 190, while harvest usually starts on day 250 (USDA 2010). The trend in Figure 6 indicates that the most useful data are collected during the growing season, peaking at days just before the harvest (around day 240).

Conclusion

This paper presents a deep learning framework for crop yield prediction using remote sensing data. It allows for real-time forecasting throughout the year and is applicable worldwide, especially for developing countries where field surveys are hard to conduct. We are the first to use modern representation learning ideas for crop yield prediction, and successfully learn much more effective features from raw data than the hand-crafted features that are typically used. We propose a dimensionality reduction approach based on histograms and present a Deep Gaussian Process framework that successfully removes spatially correlated errors, which might inspire other applications in remote sensing and computational sustainability.

Acknowledgments

We gratefully acknowledge support from SwissRe, the Stanford Data Science Institute, NVIDIA Corporation through an NVIDIA Academic Hardware Grant, Stanford's Global Development and Poverty Initiative, and NSF grant 1522054 through subcontract 72954-10597.

References

Bolton, D. K., and Friedl, M. A. 2013. Forecasting crop yield using remotely sensed vegetation indices and crop phenology metrics. *Agricultural and Forest Meteorology* 173:74–84.

Boryan, C.; Yang, Z.; Mueller, R.; and 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.

Boulton, C. A.; Shotton, H.; and Williams, H. T. 2016. Using social media to detect and locate wildfires. In *Tenth International AAAI Conference on Web and Social Media*.

Breiman, L. 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statist. Sci.* 16(3):199–231.

Cressie, N., and Hawkins, D. M. 1980. Robust estimation of the variogram: I. *Journal of the International Association for Mathematical Geology* 12(2):115–125.

DAAC, N. L. 2015. The MODIS land products. *http://lpdaac.usgs.gov*.

Dodds, F., and Bartram, J. 2016. *The Water, Food, Energy and Climate Nexus: Challenges and an Agenda for Action*. Routledge.

Ermon, S.; Xue, Y.; Toth, R.; Dilkina, B.; Bernstein, R.; Damoulas, T.; Clark, P.; DeGloria, S.; Mude, A.; Barrett, C.; and Gomes, C. 2015. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in East Africa. In *AAAI Conference on Artificial Intelligence*.

FAO. 2015. The state of food insecurity in the world. meeting the 2015 international hunger targets: Taking stock of uneven progress.

Fink, D.; Damoulas, T.; and Dave, J. 2013. Adaptive spatio-temporal exploratory models: Hemisphere-wide species distributions from massively crowdsourced ebird data. In *AAAI*.

Hinton, G. E., and Salakhutdinov, R. R. 2008. Using deep belief nets to learn covariance kernels for gaussian processes. In *Advances in neural information processing systems*, 1249–1256.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computer Science*.

Jean, N.; Burke, M.; Xie, M.; Davis, M.; Lobell, D.; and Ermon, S. 2016. Combining satellite imagery and machine learning to predict poverty. *Science*.

Johnson, D. M. 2014. An assessment of pre-and within-season remotely sensed variables for forecasting corn and soybean yields in the united states. *Remote Sensing of Environment* 141:116–128.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. F. 2014. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25(2):2012.

Kuwata, K., and Shibasaki, R. 2015. Estimating crop yields with deep learning and remotely sensed data. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 858–861. IEEE.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436–444.

Lobell, D. B.; Thau, D.; Seifert, C.; Engle, E.; and Little, B. 2015. A scalable satellite-based crop yield mapper. *Remote Sensing of Environment* 164:324–333.

Parisi, G. 1988. *Statistical field theory*. Addison-Wesley.

Quarmby, N.; Milnes, M.; Hindle, T.; and Silleos, N. 1993. The use of multi-temporal NDVI measurements from avhrr data for crop yield estimation and prediction. *International Journal of Remote Sensing* 14(2):199–210.

Rasmussen, C. E. 2006. Gaussian processes for machine learning.

Ristovski, K.; Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2013. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*.

Satir, O., and Berberoglu, S. 2016. Crop yield prediction under soil salinity using satellite derived vegetation indices. *Field Crops Research* 192:134–143.

Thornton, P. E.; Thornton, M. M.; Mayer, B. W.; Wilhelmi, N.; Wei, Y.; Devarakonda, R.; and Cook, R. B. 2014. Daymet: Daily surface weather data on a 1-km grid for north america, version 2. Technical report, Oak Ridge National Laboratory (ORNL).

United Nations, G. A. 2015. Transforming our world: the 2030 agenda for sustainable development. *New York: United Nations*.

USDA. 2010. Harvesting dates for US. *Field Crops. Agricultural Handbook* (628).

USDA. 2016. USDA national agricultural statistics service. [Accessed: 2016-09-10].

Wilson, A. G.; Hu, Z.; Salakhutdinov, R.; and Xing, E. P. 2015. Deep kernel learning. *arXiv preprint arXiv:1511.02222*.

Xie, M.; Jean, N.; Burke, M.; Lobell, D.; and Ermon, S. 2016. Transfer learning from deep features for remote sensing and poverty mapping. *AAAI Conference on Artificial Intelligence*.