Final Report: Deep Crop Yield Detection

Arman Irani, Merrick Campbell, Yuhua Situ

1 Problem Summary

1.1 Background

Predicting crop yield before harvest using neural networks would improve financial forecasting to reduce uncertainty in farm operations. Crop yield measures the quantity of food produced per acre. This measurement is an important metric for farmers' profitability. Every year, growers must meet ever-increasing consumer demand with fewer resources without sacrificing quality. In extreme cases, an insufficient yield could produce bankruptcy for the farmer and famine for the country. Several factors impact crop yield, but it is challenging to determine which factors have the greatest impact in real time since yield is usually only calculated at harvest time. We developed a Convolutional Neural Network that takes available soil, moisture, and sunlight data from satellites to predict yield for Corn, Soy, and Wheat.

1.2 Purposes

Our network will answer two primary questions: Can a neural network accurately predict crop yields for various crop types? Which factors are the most important in producing this prediction? We will attempt to answer these questions by first testing a network using a bundled set of inputs (weather, soil composition, water usage, etc) and then using a subset of input data.

1.3 Challenges

We anticipate the following challenges:

1. Not having a sufficiently large output dataset, which will be the crop yield, that might affect the performance under deep-learning.
2. Irrigation plays an important role in crop yield yet reliable and accurate irrigation data is difficult to get.
3. Aggregating and grouping datasets with different time and spatial intervals will likely require pre-processing before network training.
4. Missing features could also contribute to inaccurate results and our network should be robust under these conditions.

2 Proposed Structure

2.1 Deliverables

Our primary deliverable is a Convolutional Neural Network (CNN) that predicts crop yield using satellite data. This network predicts crop yield for one crop (corn, soybeans, or wheat) grown in the United States. Along with the network structure and code, we prepared a final report and presentation delivered during the 10th week of the Spring 2021 quarter.

2.2 Evaluation

We propose to use the following criteria to evaluate the performance of the trained network:

1. **RMSE** (Root Mean Square Error) for Prediction Accuracy
2. **MAPE** (Mean Absolute Error) for Model Performance

3 Related Work

One paper experimented using a hybrid CNN-RNN deep learning framework to predict crop yield based on environmental data and management practices[1], such as yield performance, management, weather, and soil. A systematic review[2] of current machine learning models and features used in recent publications regarding this topic. A study[3] investigated whether the coupling of crop modeling and ML models improved corn yield predictions in the US Corn Belt. A novel dimensionality approach used on remote sensing farmland data and then trained on CNN and LSTM[4] proved to outperform existing models.

4 Solutions

Our ideal solution would be a neural network tool that accurately predicts crop yield given the soil quality, applied water quantity, and sunlight for a given area. For our initial analysis, we focused on features found in satellite datasets. While a more accurate analysis would include more temporal variability and additional features (humidity, nuances between snowfall vs. rainfall, evaporation, etc.) we began with this feature set for our minimal viable product (MVP). We structured are yield model as follows:

$$Y\_\ yield = f(X\_\ water/acre,\ X\_\ sunlight/acre, X\_\ bulk\_\ soil\_\ quality)$$

These features are extracted from the ECMWF satellite dataset. This satellite cropland dataset can be used to select pixels from other satellite datasets such as the soil reanalysis and soil moisture. The yield labels came from the annual USDA crop survey where the USDA categorizes the quality of harvest as a weight quality blend per acre. Through our analysis, we narrowed our focus to a selection of three crops commonly grown in the United States: corn, wheat, and soy.

5 Schedule

In the schedule, we allocated two weeks for aggregating the multiple datasets into a unified spatial and temporal dataset for training our network. The next three weeks focused on developing the layer structure for our Deep Network. The final two weeks were devoted to refining the network, testing our hypothesis, and preparing the report. Table 1 shows the schedule and work breakdown for this project.

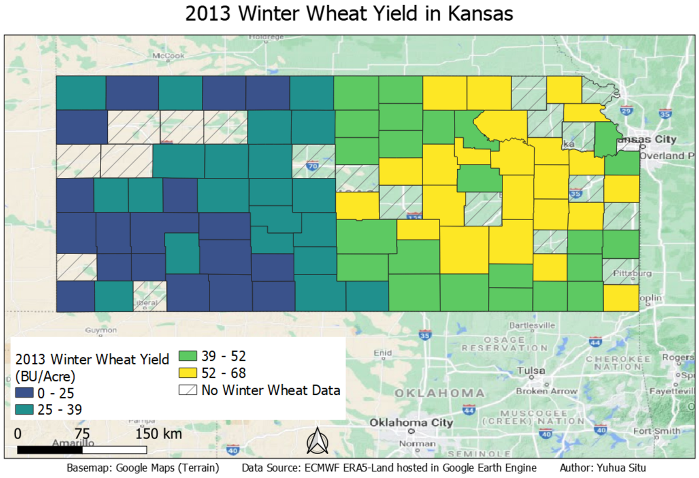
Table 1: Schedule and Work Break

Week	Deliverable	Key People
4	Aggregate Datasets	Yuhua
5	Refine Datasets	USDA Yield: Merrick Climate Reanalysis: Yuhua
6	Develop Network Structure	Arman
7	Train Network	Merrick, Arman (on Google CoLab!)
8	Train Network	All (on Google CoLab!)
9	Refine Results (& Test Additional Hypothesis)	Merrick
10	Demo + Report	Demo Slide: Yuhua Report: Merrick Submission:Arman

6 Dataset Preprocessing

6.1 Input Label Dataset

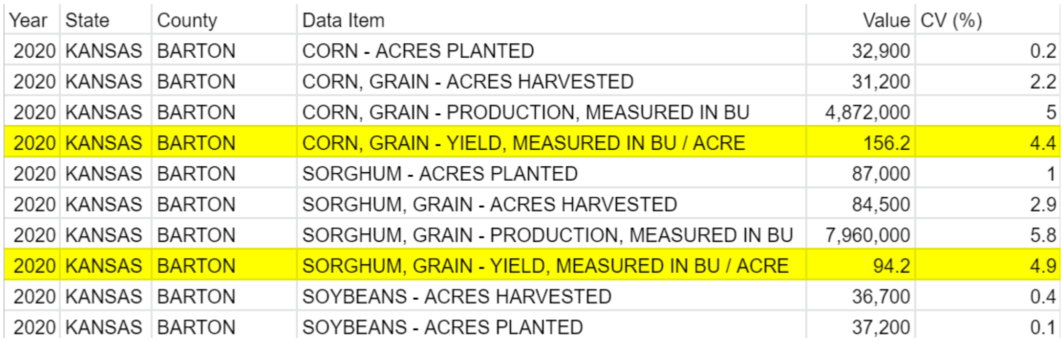
The input labels were provided by the United States Department of Agriculture (USDA) National Agricultural Statistics Service (NASS) [5]. This dataset is an annual snapshot of field conditions for every county in the United States. We downloaded the data from the USDA portal and then used Python, along with Pandas and NumPy, to restructure the dataset. Table 2 shows a sample collection of data contained in the USDA dataset while Figure 1 visuals the yield data for the state of Kansas. We wrote functions to select only the yield data for the crop of interest (Corn, Wheat, or Soybeans) and group the items in the dataset by year, county, and crop. Yield entries that were obviously incorrect, such as non-numeric entries were filtered from the dataset.


2013 Winter Wheat Yield in Kansas

Basemap: Google Maps (Terrain) Data Source: ECMWF ERA5-Land hosted in Google Earth Engine Author: Yuhua Situ

Figure 1: Overlay of the USDA yield data on Kansas, USA for 2013. Note that some counties did not produce winter wheat for a particular year

Table 2: Selection of USDA Yield Data



Year	State	County	Data Item	Value	CV (%)
2020	KANSAS	BARTON	CORN - ACRES PLANTED	32,900	0.2
2020	KANSAS	BARTON	CORN, GRAIN - ACRES HARVESTED	31,200	2.2
2020	KANSAS	BARTON	CORN, GRAIN - PRODUCTION, MEASURED IN BU	4,872,000	5
2020	KANSAS	BARTON	CORN, GRAIN - YIELD, MEASURED IN BU / ACRE	156.2	4.4
2020	KANSAS	BARTON	SORGHUM - ACRES PLANTED	87,000	1
2020	KANSAS	BARTON	SORGHUM, GRAIN - ACRES HARVESTED	84,500	2.9
2020	KANSAS	BARTON	SORGHUM, GRAIN - PRODUCTION, MEASURED IN BU	7,960,000	5.8
2020	KANSAS	BARTON	SORGHUM, GRAIN - YIELD, MEASURED IN BU / ACRE	94.2	4.9
2020	KANSAS	BARTON	SOYBEANS - ACRES HARVESTED	36,700	0.4
2020	KANSAS	BARTON	SOYBEANS - ACRES PLANTED	37,200	0.1

6.2 Input Feature Dataset

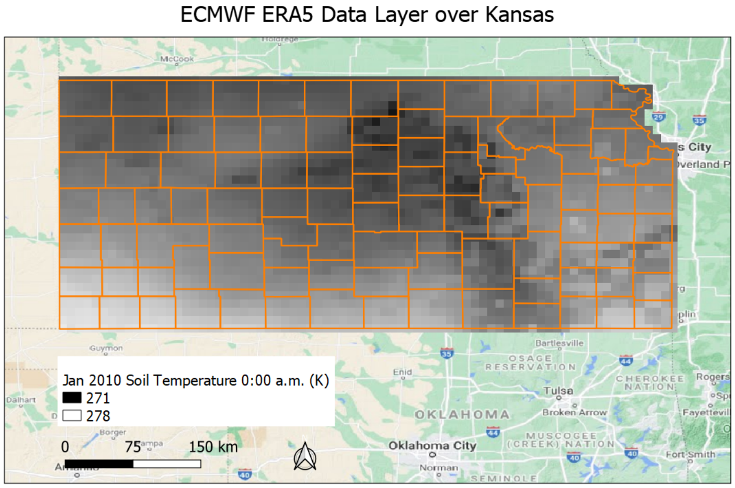
The input features were from a climate reanalysis dataset from ECMWF, namely ERA5-Land [6]. This dataset combines model data with observations from around the world into a consistent dataset. It is a publicly available collection of environmental parameters that indicate the level of downwelling solar radiation, air temperature at the surface level, and soil conditions that would impact the health of the crop. A complete list of features utilized are:

- Surface Temperature (2m, Dewpoint & Atmospheric)
- Surface Pressure
- Surface net solar radiation
- Soil Temperature (4 Depth Levels)
- Volumetric Soil Water Content (4 Depth Levels)
- Total Evaporation
- Total Precipitation
- Snowfall

  The 4 depth levels for soil temperature and soil water content are:

- Level 1: 0-7 cm
- Level 2: 7-28 cm
- Level 3: 28-100 cm
- Level 4: 100-289 cm

This dataset has a spatial resolution of 0.25° x 0.25°. Considering the solar, soil and atmospheric conditions across one county are similar at one time step, the data was binned at the county level for our analysis. The features are provided hourly for each month for the years from 2010 through 2020. This produces 12∗24=288 points for a year for each feature of interest. Figure 2 shows a visualization of the satellite data overlaid on the state of Kansas.



Basemap: Google Maps (Terrain) Data Source: ECMWF ERA5-Land hosted in Google Earth Engine Author: Yuhua Situ

Figure 2: Satellite environmental data from the ECMWF ERA 5 Reanalysis dataset overlaid on a map of Kansas, USA. The orange lines indicate the county boundaries which were used to bin the yield data for comparison purposes.

6.3 Dataset Integration Challenges

To merge these two datasets into a single dataset of input features and labels, we had to make a few key assumptions.

First, we assumed even distribution of fields and consistent field management practices across all counties with routine pest control and uniform irrigation. That way all pixels of the ECMWF data have the same weight and a simple arithmetic mean can be used to get the environmental condition for all fields in the county.

We also considered meteorological conditions with a low correlation between different months and low cumulative influences, therefore low need for a neuron to remember the state from the previous month. However, while there is a low correlation between months, there is a high correlation between neighboring hours of a month, therefore the need to extract features using convolution.

For the target crop of interest, we assumed minimal rotation with the same planting dates and harvest dates between all regions. In the meantime, we used environmental data from November to the next July in order to avoid interference from the time of harvest, so there are 216 data points of one variable (e.g. soil temperature at level 1) in the input. We also assumed uniform yield of the crop of interest for all fields in a county. We used data from the Central Valley of California, all counties in Kansas and about 40% counties selected arbitrarily in Iowa. Those counties are at a similar latitude range. And since USDA survey data did not distinguish between different types of corn, wheat, and soy, for our analysis, we neglected crop varieties.

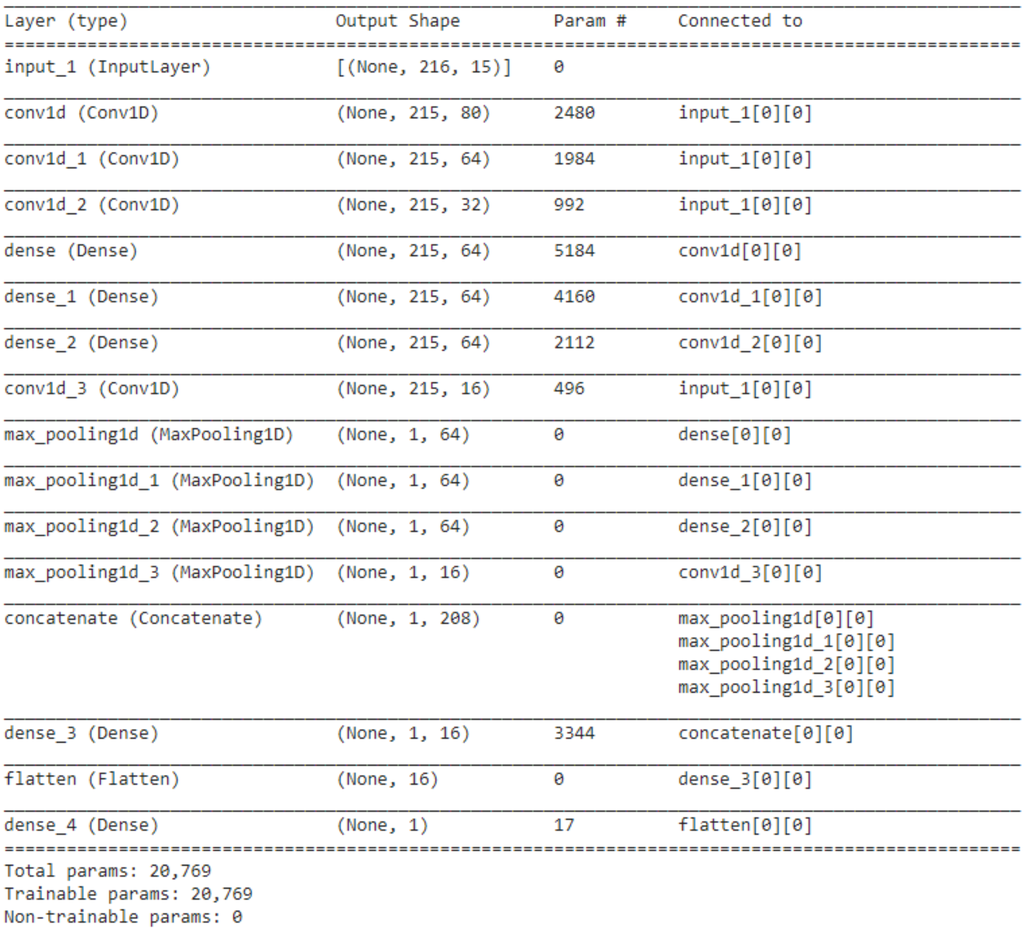
7 Network Structure

To predict the yield quantities, we developed a Convolutional Neural Network (CNN) using TensorFlow and Keras. The network utilized four 1-D Convolution Layers run in parallel to process the time series dataset. Convolution was chosen as the ideal network structure because of its ability to effectively learn from the raw representation of features and deduce the spatial and temporal dependencies of the features. The convolution filter size was utilized in order to capture different filtered features of varying time lengths. Each convolution layer was specified with a different filter size, 80, 64, 32, and 16 respectively, so as to capture the best mixture of feature representations in the data for each of these windows. The output of each convolution layer is fed through a densely connected layer with 64 units to reduce the dimensionality of the data. The network is then passed through a max pooling layer which downsamples the output into its most pertinent components. The outputs of these four pipelines are concatenated together, and reduced once more using a densely connected layer. Finally, it is flattened and passed through a dense with a single unit with a linear activation to represent the final yield prediction. Table 3 shows the network layers.

Khaki et al. (2020) uses a similar architecture, utilizing four convolutional layers, however the outputs of these are then passed through an LSTM layer before finally outputting a yield prediction [1]. We found our network performed on par with this architecture choice, and was robust enough to handle potential outliers in the data and also performed well on additional crop types such as wheat. Furthermore, our model proved to give equivalent and even better results while utilizing a smaller set of features that are more easily accessible to researchers.

The training took about 40 seconds utilizing a Tesla P100 GPU, with a batch size of 32 and 200 epochs. Each of the Convolutional layers weights were initialized using the Xavier method and ReLU activation with Adam optimizer.

Table 3: CNN Structure



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 216, 15)]    0

__________________________________________________________________________________________________
conv1d (Conv1D)                 (None, 215, 80)      2480        input_1[0][0]
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 215, 64)      1984        input_1[0][0]
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 215, 32)      992         input_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 215, 64)      5184        conv1d[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 215, 64)      4160        conv1d_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 215, 64)      2112        conv1d_2[0][0]
__________________________________________________________________________________________________
conv1d_3 (Conv1D)               (None, 215, 16)      496         input_1[0][0]
__________________________________________________________________________________________________
max_pooling1d (MaxPooling1D)    (None, 1, 64)        0           dense[0][0]
__________________________________________________________________________________________________
max_pooling1d_1 (MaxPooling1D)  (None, 1, 64)        0           dense_1[0][0]
__________________________________________________________________________________________________
max_pooling1d_2 (MaxPooling1D)  (None, 1, 64)        0           dense_2[0][0]
__________________________________________________________________________________________________
max_pooling1d_3 (MaxPooling1D)  (None, 1, 16)        0           conv1d_3[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 1, 208)       0           max_pooling1d[0][0]
                                                                 max_pooling1d_1[0][0]
                                                                 max_pooling1d_2[0][0]
                                                                 max_pooling1d_3[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1, 16)        3344        concatenate[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 16)           0           dense_3[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 1)            17          flatten[0][0]
==================================================================================================
Total params: 20,769
Trainable params: 20,769
Non-trainable params: 0
```

8 Results

Our network was able to predict the yield for corn, wheat, and soy within a reasonable margin of error. Corn had a MAPE of 34.1% and a RMSE of 23. Wheat had a MAPE of 15.15% and a RMSE of 7. Soy had a MAPE of 23.53% and an RMSE of 9. This performance was in line with the benchmark network performance from previous work in this field, where a CNN predicted a combined crop yield for corn and soy with an RMSE of 16 [1]. Figure 3 shows the results from our test and training datasets for corn, wheat, and soy.

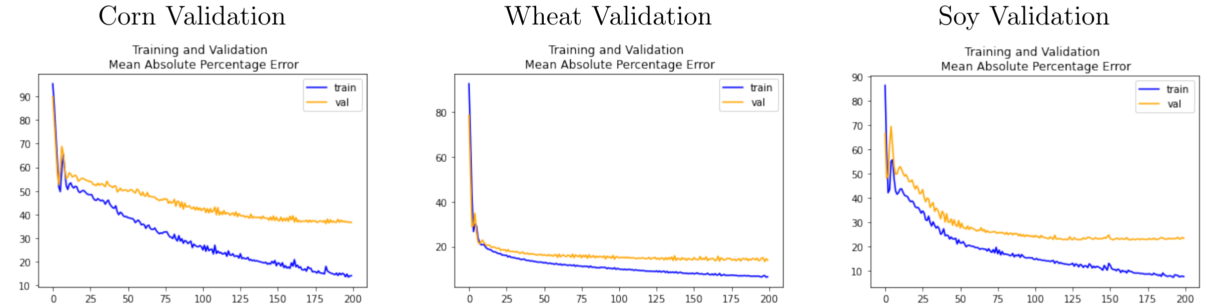


Figure 3: Training and Validation MAPE Metrics for corn, wheat, and soy

Since we had a relatively small dataset, we wanted to make sure that our results were statistically significant. We compared the resulting MAPE against the relative standard deviation (RSD) for each crop to make sure we were not overfitting a model to a dataset with a minimal variance. Figure 4 shows the histograms for each of our datasets. Corn had a RSD of 59%, and a MAPE of 34.1%. Wheat had a RSD of 35%, and a MAPE of 15.15%. Soy had a RSD of 56%, and a MAPE of 23.53% (Table 4). The MAPE values are smaller than the RSD for all of the crops, suggesting that our model provides reasonable predictions for the dataset.

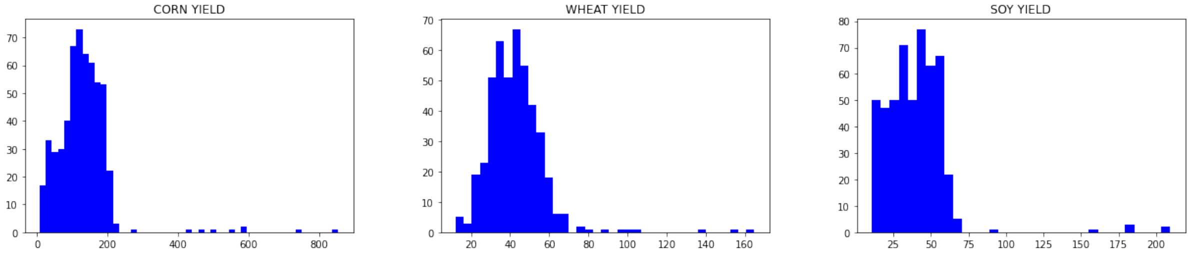


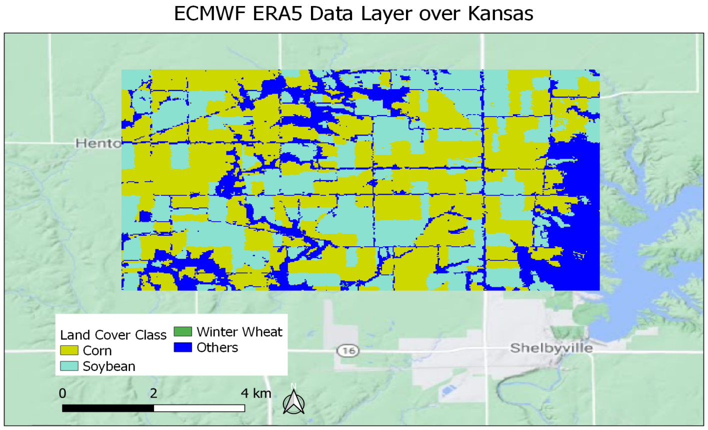
Figure 4: Histogram showing the yield distribution for corn, wheat, and soy

Table 4: Crop Dataset Statistics and Network Performance

	CORN	**WHEAT**	**SOY**
Mean Yield (BU/ACRE)	130	43	39
Standard Deviation	77	15	22
Relative Standard Deviation (RSD)	59%	35%	56%
Mean Absolute Percentage Error (MAPE)	34.1%	15.15%	23.53%
Root Mean Squared Error (RMSE)	23	7	9

9 Future Work

While our network performed well against the benchmark, a few changes could be made to improve the performance of the network. The first change to consider would be to increase the spatial resolution of the correlation between the input feature and label sets. For our analysis, we binned the yield and feature data at the county level. However, the satellite data provides information at a higher resolution. We could use the cropland layers associated with the satellite dataset to examine the input features at a field by field level instead of a county by county level. We can use a Kriging style analysis to fill in gaps within the dataset. For example, yield data is available on a county by county basis while satellite soil moisture data is available across a lat/long grid. To solve the issue of not having a large enough output dataset we can slice the Kriking-style yield heatmap to get sub-county level crop yield that can be correlated with the satellite data. Similar techniques can be applied to irrigation and weather data. This would create more inputs for training and validation sets.



Basemap: Google Maps (Terrain) Data Source: ECMWF ERA5-Land hosted in Google Earth Engine Author: Yuhua Situ

Figure 5: Sample map image showing the cropland layers in the satellite dataset for Illinois

Next, additional datasets could be added to the feature sets. Our analysis only fused two datasets, but there are additional features that could be added that include environmental features (soil surveys) and the impact of man-made irrigation tools. While we looked at some of these datasets, we were unable to incorporate them due to time constraints. With these additional datasets and higher resolution features, the network structure could be tailored to better utilize temporal information to make yield predictions during the growth cycle, which could be extremely useful to growers.

10 Sources References

10.1 Articles and Projects

Reference	Title	Source
1	A CNN-RNN Framework for Crop Yield Prediction	https://www.frontiersin.org/articles/10.3389/fpls.2019.01750/full
2	Crop yield prediction using machine learning: A systematic literature review	https://www.sciencedirect.com/science/article/pii/S0168169920302301?dgcid=rss_sd_all
3	Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt	https://www.nature.com/articles/s41598-020-80820-1
4	Combining Remote Sensing Data and Machine Learning to Yield Predict Crop Yield	http://sustain.stanford.edu/crop-yield-analysis

10.2 Utilized Datasets

These datasets were utilized in the MVP.

Reference	Data	Period	Source Link
5	Crop yield by county	2010-now	https://quickstats.nass.usda.gov/
6	Soil Reanalysis (ECMWF)	1981-now	https://developers.google.com/earth-engine/datasets/catalog/ECMWF_ERA5_LAND_HOURLY

11 Appendix

11.1 Link to Notebook

For UCR email addresses only:
https://colab.research.google.com/drive/1p-KMoYk9H1VD8kh121VJk-wt76jfgn5K?usp=sharing

11.2 Future Datasets

These datasets were not incorporated in our project, but would be useful for future work.

Data	Period	Source Link	Comment
Precipitation (NOAA)	2000-now	`https://www.ncdc.noaa.gov/cdo-web/datatools/selectlocation`	This link is for station-wise weather information
Temperature (NOAA)	2000-now	`https://www.ncdc.noaa.gov/cdo-web/datatools/selectlocation`	This link is for station-wise weather information
Sunlight (Days of sunshine/growing season)	unknown	`https://openweathermap.org/history`	May not intersect well with NOAA/USDA data
Photosynthetically active radiation (PAR)	2001-now	`https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/science-domain/photosynthetically-active-radiation/` `https://oceancolor.gsfc.nasa.gov/cgi/l3`	There is version 6 Level 3 (global coverage) product produced daily at 5 kilometer pixel resolution with estimates of PAR every 3 hours, but it only covers water surfaces
Precipitation index (NCAR, SPI)	2003-now	`https://climatedataguide.ucar.edu/climate-data/standardized-precipitation-index-spi`	The index may be different depending on how long the period (e.g. 12 months) is used for the calculation.
Drought (PDSI)	1850-2018	`https://rda.ucar.edu/datasets/ds299.0/`	Palmer Drought Severity Index 2.5°x2.5°
Soil Type	unknown	`https://websoilsurvey.sc.egov.usda.gov/App/WebSoilSurvey.aspx`	Soil composition and bedrock foundation for the continental United States, assumed to be invariant in our timescale
Soil Moisture (SMAP)	2015-now	`https://developers.google.com/earth-engine/datasets/catalog/NASA_USDA_HSL_SMAP10KM_soil_moisture`	10km resolution
Soil Moisture (ASTER L1)	2000-now	`https://developers.google.com/earth-engine/datasets/catalog/ASTER_AST_L1T_003#description`	Resolution is 30m in short-wave IR, 90m in thermal IR. Infrared band contain information on leaf water content (crop health)
Soil radar (Sentinel-1)	2014-now	`https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S1_GRD`	Resolution is 10m. Available in HH, HV, VV, and VH bands
Cropland data layers	1997-2020	`https://developers.google.com/earth-engine/datasets/catalog/USDA_NASS_CDL`	254 classes One crop class map each year

11.3 Declaration of No Project Overlap

While Merrick Campbell was enrolled in both 228 and 243 this quarter, there was no overlap between final projects. The project for EE228 predicted crop yield using a CNN to process satellite data while the final project for EE243 used classical computer vision approaches (filtering and morphology) to count the number of oranges on a tree. The EE228 codebase is in python/javascript while the EE243 codebase is entirely in C/C++.