IN AND END OF SEASON SOYBEAN YIELD PREDICTION WITH HISTOGRAM BASED DEEP LEARNING

E. Erik[1], M. Durmaz[2] *, A. Ö. Ok[3]

[1]HAVELSAN A.Ş., Ankara, Türkiye - eerik@havelsan.com.tr
[2]Dept. of Geomatics Engineering, Hacettepe University, Ankara, Türkiye - muratdurmaz@hacettepe.edu.tr
[3]Dept. of Geomatics Engineering, Hacettepe University, Ankara, Türkiye - ozgunok@hacettepe.edu.tr

KEY WORDS: Crop Yield Prediction, QGIS Plugin, Google Earth Engine, Histogram, Deep Learning

ABSTRACT:

One sector that feels the effects of global warming and climate change on all levels is agriculture. In order to prepare for possible yield loss, as well as market, storage, and import planning challenges brought on by climate change, businesses can utilise agricultural decision support applications. Within the scope of this study, a crop yield prediction module has been developed that can provide in and end of season estimation of crop yields to be obtained from the determined regions. The Python programming language was used in the creation of the module as a QGIS plugin. The area for which crop yield predictions are to be made is covered by retrieving MODIS SR, MODIS LST, and Daymet data from the Google Earth Engine data catalogue. Histograms obtained from remotely sensed images are used as input data to two deep learning methods (CNN-LSTM and HistCNN). As a result, the HistCNN model outperformed CNN-LSTM for in season soybean yield prediction, with an R^2 of 0.72, while the CNN-LSTM model outperformed it for in end of season soybean yield prediction, with an R^2 of 0.67.

1. INTRODUCTION

The Earth's surface temperature is gradually rising as greenhouse gas emissions from the usage of fossil fuels expand, resulting in long-term local, regional, and global climate change. The effects of climate change include the melting of ice, the change of precipitation patterns, and the migration of wildlife to different regions. In particular, agricultural production is the activity most affected by climate change, and due to unexpected climate conditions, agricultural yields can be reduced. Therefore, it must be carefully monitored and managed.

Apparently, the quantity of yield is particularly vital for developing countries and any decrease has also the impact on farmers, government agencies, and agricultural insurance companies (Basso et al., 2013). In addition, governments require yield estimates prior to harvest for purposes such as arranging for imports and exports, estimating market prices, and determining storage requirements (Cunha and Silva, 2020). Furthermore, post-harvest yield estimates are required to determine the agricultural insurance and governmental support.

Predictions of crop yield can be used to generate pre- and post-harvest forecasts. With the help of remote sensing data, we can extract the factors affecting agricultural yield and then use deep learning techniques to analyse the link between these features and the crop yield. As a first step, the data sources that provide the attributes that have the potential to affect crop yield should be identified. For crop yield analysis, it is essential to analyse the crop's growth status indicators and the environmental conditions where the crop is located.

With the advancement of technology, the availability of remote sensing data and the volume of data that may be utilised have considerably grown (Weiss et al., 2020). Besides, the characteristics that determine crop yield can be obtained from various public sources. Nonetheless, the temporal and spatial integrity of the obtained information is critical. On top of all, a substantial quantity of data is required for the development of effective training models in deep learning approaches. All of these point to the necessity of collecting long-term data for remote sensing from a reliable data source. Besides, such a vast quantity of collected data must be stored and efficiently processed. Because the decision support module to be built is an end-user package, it must be capable of responding without storing and processing data on local computers, allowing the application to operate on as few hardware resources as possible. Therefore, Google Earth Engine (GEE) was chosen as the base platform for this study since it met all of the above-given criteria.

GEE is a cloud computing platform for scientific geospatial data analysis and visualisation (Kennedy et al., 2018). In addition, this Google cloud platform service enables the processing of massive amounts of data with high-performance resources. GEE is therefore one of the most popular platforms for processing the largest geodata sets available. It is highly beneficial for mapping and monitoring crops, phenology-based classification, agricultural yield estimation, and other studies (Kennedy et al., 2018). In this study, GEE's services were used both to access remote sensing data and to analyse the acquired information.

Deep learning algorithms, a subfield of machine learning, accelerate analytical learning by simulating human neural networks with artificial neural networks. In crop yield prediction studies, the Convolutional Neural Network (CNN), the Long Short-Term Memory (LSTM), and the Deep Neural Network (DNN) are the most often adopted deep learning algorithms. Hybrid techniques that combine the two algorithms are also popular (van Klompenburg et al., 2020). CNN-LSTM has been utilised in studies on crop yield estimation and forecasting to extract important characteristics from historical information. It is appropriate to use LSTM networks to make a scalar estimate of crop yield from n temporal images of a year (Khaki et al.,

* Corresponding author

2020). It has been also reported that the CNN-LSTM model is more accurate than CNN-only and LSTM-only models (Sun et al., 2019). Besides, in histogram-based CNN (HistCNN), by using a multi-layer CNN, a hierarchical learning model can be created in which close associations with the input occur in the lower layer and distant relations occur in the downstream CNN layers (Gehring et al., 2017).

As is well known, it’s possible to browse, edit, print, and analyse geospatial data with Quantum GIS (QGIS), a free and open-source cross-platform desktop GIS application. Therefore, in this work, a QGIS plugin is developed using Python for crop yield prediction. Besides, the GEE Python API is utilised to access and process the remotely sensed data. Meanwhile, CNN-LSTM and HistCNN deep learning models are tested from the utilised data, which is then used to predict the crop yield. We preferred the 15 states in the United States’ CONUS region where soybeans are the most widely produced crop for our study.

This paper is organised as follows: We discussed the study area and the datasets utilised in Section 2. Section 3 goes over the methodology and QGIS plugin that were developed. Section 4 discusses the evaluation strategies for the proposed methodologies as well as the achieved accuracy. Section 5 provides a conclusion with some final remarks.

2. STUDY AREA AND DATASET

This study is conducted in the 15 U.S. CONUS states where soybeans are cultivated the most (Figure 1). Ground truth information of soybean yield was obtained from United States Department of Agriculture (USDA) National Agricultural Statistics Service (NASS) ((dataset) USDA National Agricultural Statistics Service (2017)”, n.d.a). As is public knowledge, the USDA’s National Agricultural Statistics Service (NASS) conducts hundreds of surveys annually and produces reports covering practically every aspect of U.S. agriculture, and this service provides county-level soybean yield information for the years 2006–2021. As required by deep learning models, ground-truth yield data serves as the necessary labelled information.

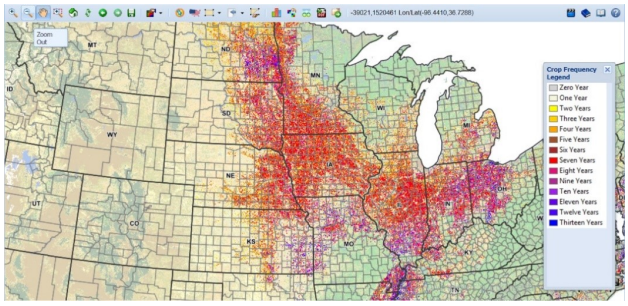

Figure 1. Density map of soybean of the U.S. CONUS region.

In this study, the shapefile vector file can be used to gather information about the borders of the study area. The county boundaries of the United States are collected from the GEE DataCatalog TIGER/2018/Counties dataset. It is used to gather the related geometric information for the region being processed. Moreover, it is used to filter both the input data and the crop data based on the region shape that was derived from the collected information.

Cropland Data Layers (CDL) is a land-cover data layer tailored to crops. Using satellite and agricultural ground truth data, it is produced annually for the United States ((dataset) USDA National Agricultural Statistics Service (2017)”, n.d.b). This information is also freely accessible through GEE’s USDA/NASS/CDL data catalogue. A crop mask is applied to remotely sensed images to remove areas where the crop is not grown. The code block provided in Appendix 1 demonstrates the generation of a CDL soybean mask using region and year data. Using the produced mask, non-soybean pixels in remote sensing data are filtered out.

The seven band values in the MODIS surface reflectance (SR) data can provide information such as crop growth status indicators. MODIS land surface temperature (LST) data provide daily and nightly ground surface temperature information, whereas Daymet weather data include information on vapour and precipitation (Table 1). These data are analysed to determine the crop’s environmental conditions. All of this input data can be accessed using the GEE data catalogue by providing the necessary study area information (Table 2).

Data Name	Catalog Name	Resolution
MODIS SR	MOD09A1	500 m - 8 days
MODIS LST	MOD11A2	1 km - 8 days
DAYMET V4	DAYMET/_V4	1 km -1 day

Table 1. Image dataset utilized in this study.

Data Catalog Name	Bands
MODIS/006/MOD09A1	sur_refl_01- sur_refl_07
MODIS/006/MOD11A2	LST_Day, LST_Night
NASA/ORNL/DAYMET_V4	prcp, vp

Table 2. Data catalog names and the bands utilized.

3. METHODOLOGY

In this section, we detail our strategy in four sub-sections: (i) preparation of the input dataset, (ii) tensor generation, (iii) prediction strategy, and (iv) QGIS plugin development.

3.1 Preparation of the Input Dataset

In the first step, the data from MODIS SR, MODIS LST, and Daymet are required to be put together to generate a total of eleven bands. Note that the theoretical upper and lower bounds for these bands are too large; therefore, in this study, we used the min-max pixel values within the study area as our upper and lower bounds, respectively. In Table 3, the Old Min-Max column shows the real band limits, whereas the New Min-Max column displays the values valid for our study region. During implementation, the related remote sensing data are collected using county geometry and year information. Note that, unlike other data, the cloud removal process is applied to MODIS SR data.

Feature	Old Min-Max	New Min-Max
MOD09A1	-100 - 16000	1 - 5000
MOD11A2	7500 - 65535	12400 - 15600
Precipitation (mm)	0 - 544	0 - 35
Vapor pressure (Pa)	0 - 8230	0 - 3200

Table 3. Theoretical and calculated ranges for input features.

It is necessary to collect historical data features and crop yield information in order to obtain the training data necessary for the deep learning model. In this study, a histogram-based approach is used to collect information on the yield of all band values and to gain more information with fewer data. This approach assumes that meaningful information may be derived from the number of different pixels in images acquired for the study area, disregarding the region's geographical features (You et al., 2017).

3.2 Tensor Generation

Using the histogram method, it is possible to determine the characteristics of annual yield data by combining histograms sequentially. Obtaining histograms of standard sizes and associating them with the annual yield delivers a tensor. Figure 2 illustrates the generation steps for tensors. While creating the tensors, the extent of the study area and the starting and ending dates of the training data are given.

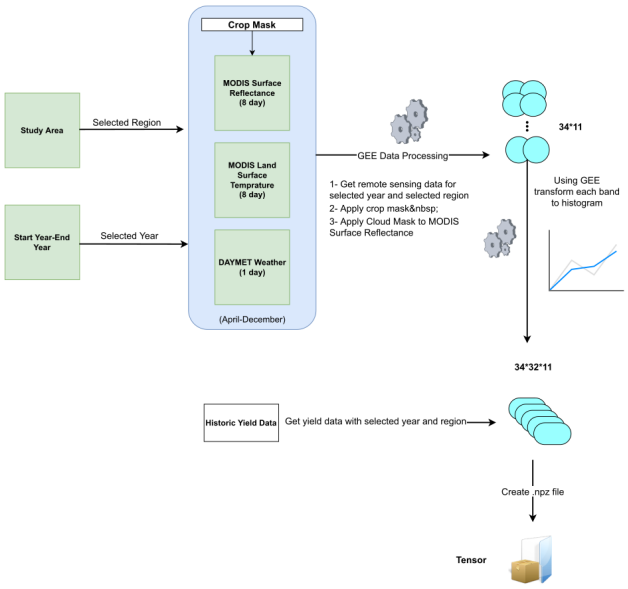

Figure 2. General workflow of tensor generation.

This study follows the soybean plant growth period from April to December. During this period, 34 different remote sensing images are obtained at 8-day intervals. With the help of GEE, we are capable of collecting MODIS SR, MODIS LST, and Daymet weather data for our study area and applying crop masking. To ensure that Daymet weather data has the same periodicity as MODIS data, 8-day average values are calculated. The 34 images that result, each with 11 bands, are saved as a GEE ImageCollection. In this case, filtering, masking, and band merging are all accomplished with the assistance of GEE's capabilities, and the GEE functions. GEE functions used for filtering and masking MODIS LST data are provided in Appendix 2. Notice that all image data is analysed directly on Google Cloud servers, without the datasets being downloaded to the local computer. Tensors are created for each county in the study region for the years 2006–2019. If there is no information regarding soybean production during a year, the tensor for that year is ignored.

In this study, CNN layers are used to extract the properties affecting crop yield using surface reflection values, day-night temperature values and weather data. The 3D sequential histograms obtained in $time \times bin \times bands$ format are used in the 2D CNN to extract important features that affect the crop yield. As mentioned before, the created ImageCollection has 34 images, each with 11 bands. Each image band is divided into 32 bins during the histogram generation step. Hence, an npz-formatted tensor is formed utilising the $34 \times 32 \times 11$ ImageCollection and the yield information.

The CNN-LSTM and HistCNN models are selected as prediction models in this study. In addition, 5-fold cross validation is carried out to achieve an appropriate deep learning model configuration. During implementation, the tensors for the years 2006–2019 are used as training data, while the tensor data for the year 2020 is reserved for testing purposes.

3.3 Prediction Strategy

Crop yield prediction is important both for in season and end of season. In season prediction is important and preferred to supply support for agricultural decisions. Farmers, decision makers and other users can take related precautions and build plans. Besides, end of season prediction is important for assessing farmer statements for agricultural insurance and government agency support payments.

In this study, we performed crop yield prediction for both in season and end of season types. Only data from April through the period determined in the season are used in this work for in season predictions. When predicting the end of season out, all relevant information from April to December is considered.

3.4 Development of QGIS Plugin

QGIS Crop Yield Prediction module has three different actors; user, GEE and QGIS. Figure 3 depicts "The Prepare Training Dataset," the first activity diagram for which a user interface was also developed. This interface requires the user to select the application working directory, study region, crop type, and training years. The application's workspace is called the "Working Directory." It stores study area, trained models, prediction feature, trained features, yield label data, crop data layers. All of the information is kept in different subfolders consisting of StudyAreas, TrainedModels, TrainingData, YieldLabelData and Auxiliary-Data. It should be selected by the user at the start of the application. Additionally, the "Prepare Training Dataset" UI receives input data from GEE. After that when with "Create Training Data" action tensor creation process is started. When training data is ready to use, it saves under the working directory to use when training phase.

"Train Prediction Model" is another UI tab inside the developed plugin. In this step, the plugin expects the user to select the prediction type and training model. The activity diagram of this UI tab is displayed in Figure 4. "Predict Yield" is the final tab in the plugin's interface (Figure 5). Both user and GEE inputs are received. The prediction process begins with the selection of a region and a model. When the prediction is completed, results is shown as temporary shape file on the map. In Figure 6, predictions that in the prepared temporary shape file is shown. Predictions are given in kilogrammes per hectare.

4. RESULTS AND DISCUSSION

This section presents the study's findings and addresses the issues experienced during the data analysis. In total, six different models were created for HistCNN and seven different models

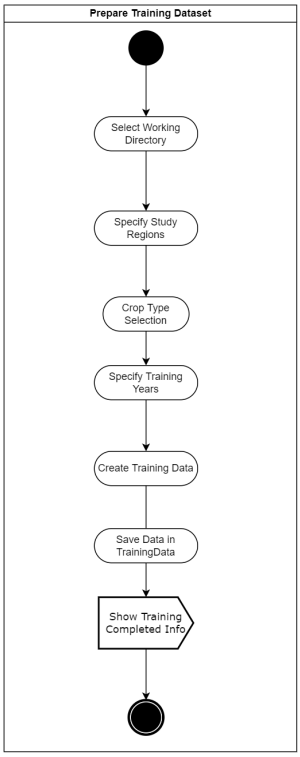

Figure 3. Prepare Training Dataset

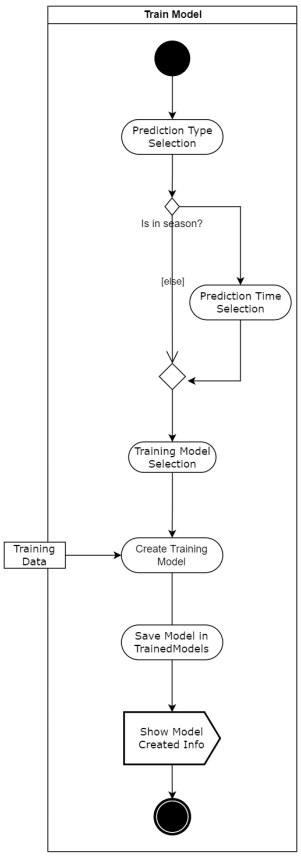

Figure 4. Train model.

were created for CNN-LSTM. Different model combinations

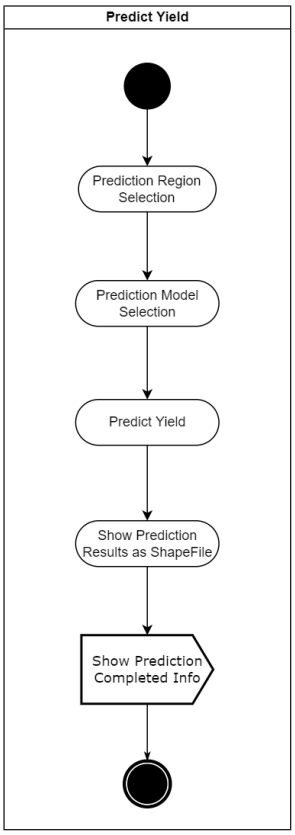

Figure 5. Predict yield.

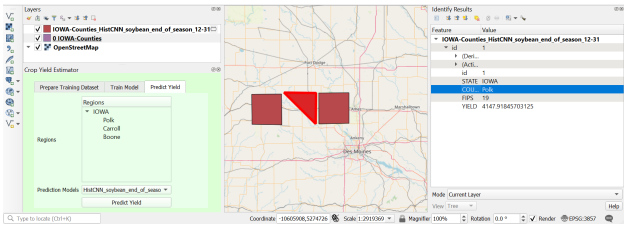

Figure 6. Preparing training dataset is completed interface.

were created by differentiating the number of convolution layers, the number of filters, and the number of dense layer neurons compared to the base model. Each model was separately evaluated for in season and end of season predictions. For in season yield predictions, the 18th time is selected.

The effects of different parameters and their combinations on the accuracy of the crop yield prediction are investigated. The training and test data are split randomly to compare all model configurations. Thus, a 5-fold cross validation is used to compare each model objectively. Finally, model selection is performed using training data. Table 4 and Table 5 show the created model configurations.

Table 6 presents the 5-fold cross validation results of the HistCNN models. The 2th and 4th models provide the worst results for in season and end of season predictions. This result proves that the convolution layer and the filters are important for the performance of the HistCNN model. Table 7 presents the model comparisons of the CNN-LSTM models after the cross validation. Amongst the models, 3th model configuration is the most successful model for both prediction types. 7th model configur-

HistCNN Models	Conv Filter Sizes	Conv Count	Dense Count	Dense Neuron Count
1.Model	128, 256, 512	3	1	64
2.Model	32, 64, 128	3	1	64
3.Model	64, 128, 256, 512	4	1	64
4.Model	128, 256	2	1	64
5.Model	128, 256, 512	3	2	64,32
6.Model	128, 256, 512	3	1	32

Table 4. HistCNN model configuration matrix.

CNN-LSTM Models	Conv. Filters	Conv. Count	LSTM Neurons	Dense Count	Dense Neurons
1.Model	32,64	2	256	1	64
2.Model	64,128	2	256	1	64
3.Model	128,256	2	256	1	64
4.Model	32,64,128	3	256	1	64
5.Model	32,64	2	256	2	64,32
6.Model	32,64	2	256	1	128
7.Model	32,64	2	512	1	64

Table 5. CNN-LSTM model configuration matrix.

ation has the worst error rate among the in season predictions. Therefore, it can be said that the increase in the number of dense layer neurons negatively affected the deep learning output in our case.

Model Name	In Season (18th)		End of Season	
	RMSE	R^2	RMSE	R^2
1.Model	330.807	0.77	309.950	0.80
2.Model	353.455	0.74	331.638	0.77
3.Model	308.418	0.80	317.138	0.79
4.Model	331.427	0.77	336.802	0.76
5.Model	323.097	0.79	333.207	0.77
6.Model	334.020	0.77	324.690	0.78

Table 6. 5-fold cross validation results of HistCNN model.

CNN-LSTM and HistCNN model were tested with test data consisting of 128 tensors that were not included in the training data. According to CNN-LSTM test results, the best end-season yield estimates were obtained with the 2nd model configuration with R^2 of 0.67. The best in season yield estimation results were obtained with the base model with R^2 of 0.56.

In the HistCNN test results, while the 3rd model is the most successful model in predicting soybean yield in season with R^2 of 0.72, the 2nd model is the most successful model in end of season predictions with R^2 of 0.62. Besides, the lowest prediction results were obtained with 4th model configuration for both prediction types with R^2 of 0.3.

The R^2 values derived from the prediction data are lower than those derived from the 5-fold cross validation data. This could be due to the fact that the only data used for testing is for the year 2020. Using 128 tensors from 15 randomly selected states was also incapable of producing reasonable R^2 results. In addition, the prediction results indicate that HistCNN models perform better than CNN-LSTM models for both in season and end of season predictions.

Model Name	In Season (18th)		End of Season	
	RMSE	R^2	RMSE	R^2
1.Model	364.152	0.73	333.333	0.77
2.Model	350.693	0.75	338.191	0.76
3.Model	337.279	0.77	314.482	0.79
4.Model	363.638	0.73	320.531	0.79
5.Model	347.959	0.75	315.476	0.79
6.Model	363.347	0.73	331.090	0.77
7.Model	409.001	0.66	325.191	0.78

Table 7. 5-fold cross validation results of CNN-LSTM model.

The module for predicting crop yields that has been developed relies on surface reflectance data for determining the current state of the crop and on analyses of environmental factors such as temperature, precipitation, and vapour to better better predict crop yields. However, in reality, the prediction module needs to account for farmers' spraying, irrigation, and fertilisation practises. Examining how the soil's properties have changed over time is also crucial for estimating yield.

5. CONCLUSION

This study focuses on developing a QGIS plugin for crop yield prediction. It delivers in season and end of season crop yield predictions by using remote sensing data and deep learning methods. A key component is the use of GEE, a cloud computing platform for the collection and processing of remote sensing data.

During the methodology, images were standardised using a histogram-based strategy so that deep learning models could learn more information with less input. HistCNN and CNN-LSTM models were constructed using the Tensorflow library. Several model configurations were tested, and RMSE, MSE, and R^2 metrics were used to evaluate the results. 5-fold cross validation was applied to determine the optimum deep learning model configuration. As a result, the HistCNN model outperformed CNN-LSTM for in season soybean yield prediction, with an R^2 of 0.72, while the CNN-LSTM model outperformed it for in end of season soybean yield prediction, with an R^2 of 0.67.

In future studies, in addition to remote sensing data, farmers' agricultural practises and soil characteristics can be used to obtain more reliable and accurate yield predictions. In addition, it is thought that the GEE "Batch Environment" method can decrease the training data duration. Moreover, the plugin user interface can be effective when using the batch processing method. However, in future studies, the performance of the plugin developed in this study on different crop types, such as corn and cotton, can be addressed.

ACKNOWLEDGEMENT

This study was carried out as the first author's Master of Science thesis titled "In and End of Season Soybean Yield Prediction with Histogram Based Deep Learning."

REFERENCES

Basso, B., Cammarano, D., Carfagna, E., 2013. Review of Crop Yield Forecasting Methods and Early Warning Systems. *The*

First Meeting of the Scientific Advisory Committee of the Global Strategy to Improve Agricultural and Rural Statistics, 1–56.

Cunha, R. L. D. F., Silva, B., 2020. Estimating Crop Yields with Remote Sensing and Deep Learning. *2020 IEEE Latin American GRSS and ISPRS Remote Sensing Conference, LAGIRS 2020 - Proceedings*, 273–278.

(dataset) USDA National Agricultural Statistics Service (2017)”, year=2021, m. n., n.d.a. NASS - Quick Stats. USDA National Agricultural Statistics Service. https://data.nal.usda.gov/dataset/nass-quick-stats.

(dataset) USDA National Agricultural Statistics Service (2017)”, year=2021, m. n., n.d.b. Usda national agricultural statistics service cropland data layer. https://nassgeodata.gmu.edu/CropScape.

Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y. N., 2017. Convolutional Sequence to Sequence Learning. http://arxiv.org/abs/1705.03122.

Kennedy, R. E., Yang, Z., Gorelick, N., Braaten, J., Cavalcante, L., Cohen, W. B., Healey, S., 2018. Implementation of the LandTrendr algorithm on Google Earth Engine. *Remote Sensing*, 10(5), 2019–2021.

Khaki, S., Wang, L., Archontoulis, S. V., 2020. A CNN-RNN Framework for Crop Yield Prediction. *Frontiers in Plant Science*, 10(January), 1–14.

Sun, J., Di, L., Sun, Z., Shen, Y., Lai, Z., 2019. County-level soybean yield prediction using deep CNN-LSTM model. *Sensors (Switzerland)*, 19(20), 1–21.

van Klompenburg, T., Kassahun, A., Catal, C., 2020. Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, 177(August), 105709. https://doi.org/10.1016/j.compag.2020.105709.

Weiss, M., Jacob, F., Duveiller, G., 2020. Remote sensing for agricultural applications: A meta-review. *Remote Sensing of Environment*, 236(December 2018), 111402. https://doi.org/10.1016/j.rse.2019.111402.

You, J., Li, X., Low, M., Lobell, D., Ermon, S., 2017. Deep gaussian process for crop yield prediction based on remote sensing data.

A. APPENDIX

Appendix 1- The function of generating soybean mask using CDL data.

```
def create_soybean_mask(self):
# Eliminating non soybean pixels
 cdl = ee.ImageCollection('USDA/NASS/CDL')
 .filter(ee.Filter.date
 (self.historic_start_date,
 self.historic_end_date))
 .filterBounds(self.region)

  def createMask(cdl):
    # Soybean mask for selected year
    imageCdl = ee.Image(cdl.toList(1)
    .get(0))
    soybeanMask = createMask(imageCdl)
    return soybeanMask
```

Appendix 2- Function to retrieve MODIS LST data for the selected year and region.

```
def getMODLST(self):
 MOD_lst = self.create_image_collection
 ('MODIS/006/MOD11A2',
 self.start_date,
 self.end_date, self.region)
 MOD_lst = MOD_lst.select(self.LST_bands)

 def updateMasking(image):
  return image.updateMask(self.soybeanMask)

MOD_lst = MOD_lst.map(updateMasking)
```