HIGH SCHOOL EDITION
Journal of Student Research

Volume 10 Issue 4 (2021)

# IoT and AI Based Integrated System to Detect Crop Diseases and Deficiency of Nutrients in a Large Farm

Shraman Kar[1] and Jeffrey Willert[#]

[1]duPont Manual High School, MI, USA
[#]Advisor

## ABSTRACT

The world population is expected to increase by 2 billion by 2050. A person dies of hunger every 3.6 seconds. The UN has come up with a list of 17 goals to make the world a better place and ending hunger comes at second. A 119% increase in yield is required by 2050 to sustain life. As the amount of land cannot be increased, the only way is by increasing crop yield from the same land. Research has shown that disease and nutrient deficiency related yield losses have cost the world 60% of global agricultural productivity. Diagnosis of the disease and nutrient deficiency in a large farm is very difficult because of the number of crops. This project, "Crop-Mates", aims to identify crop diseases in a vast area of farmland by first surveying the acreage using a GPS-enabled drone. The camera-fitted drone captures video and pictures of the farm while flying over the field. A deep learning algorithm is created to identify leaves from these pictures. Another deep learning-based algorithm using AlexNet architecture is trained with 38 different classes of diseases on 87,000 images of leaves to identify diseases from these leaves. An Artificial Neural Network is developed and is trained on 20,000 data points to identify nutrient deficiencies in the soil from IoT-enabled sensors put across the farm. An app is built to show the results and to recommend the type and amount of pesticide and fertilizer for optimum crop yield.

## Background / Purpose of Work

The world population is expected to increase by 35% over the next 30 years[1]. If society continues on this dietary trajectory, a 119% increase in edible crops grown will be required by 2050 to meet dietary demands [2]. Both increases in planted area and increases in yield are likely needed to meet global demands for grain. However, the cultivation of new acreage requires land clearing and subsequent tillage that results in significant greenhouse gas emissions which also has negative impacts on biodiversity and water quality. Thus, yielding more crops from the same land is of paramount importance for society to function properly. To maximize yield from the same acreage, the crops need to be healthy, free of diseases and be grown with optimal nutrients. Crop diseases and nutrient deficiencies have always been a challenge to plant growth and crop production in several parts of the world. They can affect plants by interfering with several processes such as the absorbance and translocation of water and nutrients, photosynthesis, flower and fruit development, plant growth and development, cell division, and enlargement. Farmers cannot maximize their yields primarily due to plant disease and nutrient deficiencies. Nutrient deficiencies and Crop Disease cause losses of as much as 60% of the most important crops [3]. Our solution to this problem is to identify the diseases and deficiencies in the plant quickly and at scale and suggest the appropriate treatment so that farmers can apply them easily and efficiently.

Various efforts have been developed to prevent crop loss due to diseases and deficiencies. One possible approach is to apply widespread pesticides to plants. However, it causes a significant amount of waste and pollution and is also harmful to crops as nutrients can become toxic when given in excess. Another approach is to identify the plant disease and give the appropriate treatment based on the disease. Plant diseases are often classified by their symptoms. Many diseases, however, produce practically identical symptoms but are caused by very different microorganisms or agents, thus requiring completely different control methods. Identifying plant disease with a naked eye is thus extremely difficult so leaf samples have to be taken to a lab to investigate the types of diseases in the crops. The whole process is very laborious, takes a lot of time, and is expensive and oftentimes impossible for a large farm. Our project proposes to identify disease in plants in a fast and efficient way by using state-of-the-art technologies, such as a Convolutional Neural Network and YOLO architecture for detection and GPS-enabled and camera-fitted drone to survey and fly over a farm to quickly capture the images of the farm.

Other than the disease, crops go through nutrient deficiency. Nutrient deficiency occurs when an essential nutrient is not available in sufficient quantity to meet the requirements of a growing plant and the plant is toxic when excess nutrients are given to the plant (refer figure 1).

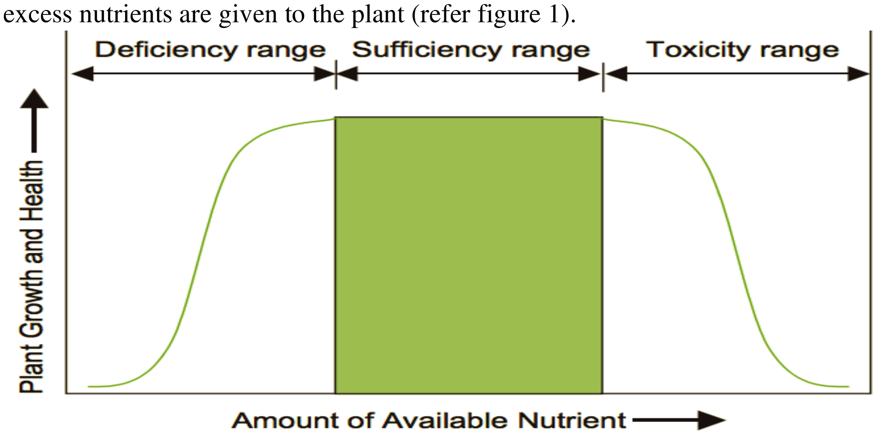



**Figure 1.** Relationship about the relationship between plant growth and health and the number of nutrients available [4]. (Brady and Weil, 1999)

The most common nutrient deficiencies are Nitrogen (N), phosphorus (P), and potassium (K). The three common ways for diagnosing nutrient deficiencies are soil testing, plant analysis, and visual observations in the field [5]. But all of these processes are laborious, time-consuming, and expensive. Our solution proposes to detect nutrient deficiencies using data from nutrient sensors from across the farmland. Then based on the data from the sensors, an appropriate amount of nutrients will be suggested for the plant so that the plant is not deficient or toxic.

## Methodology

CropMates consists of a Camera fitted drone, soil sensors, a mobile app, and a few algorithms to connect all together. In this flowchart I'll describe what CropMates does. CropMates detects crop diseases from individual leaves in a vast area of farmland by surveying the acreage using a GPS-enabled drone and it detects nutrient deficiencies from soil nutrient sensors and recommends the appropriate pesticide, fertilizer and water based on the detection. Following is a flowchart showing the workflow of CropMates.

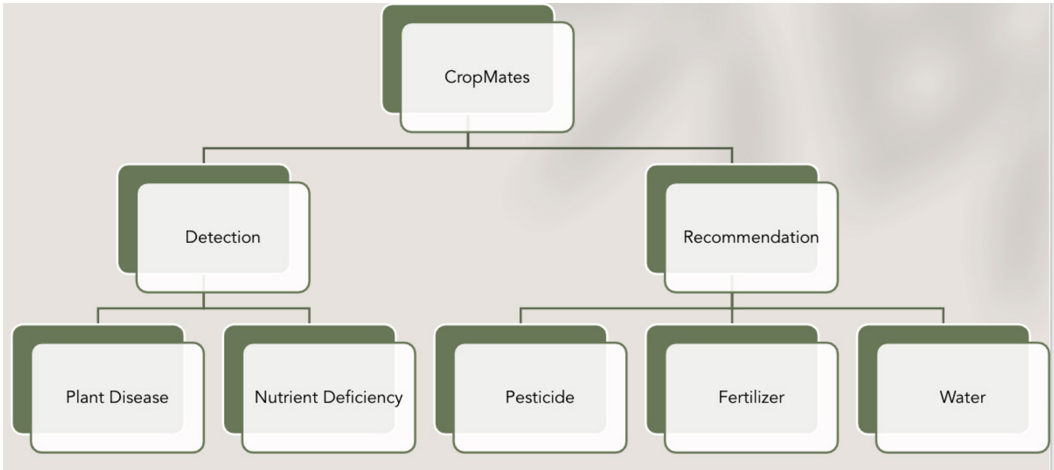



**Figure 2.** Flowchart of CropMates

The purpose of this project, CropMates, is to increase crop yield by detecting diseases and nutrients deficiencies and give proper recommendations via mobile app to correct them.
To achieve this, a system is developed to capture images of leaves of the plants via GPS enabled and camera fitted drone and get data from nutrient sensors placed in the soil. The images of the leaves are passed through a machine learning-based algorithm, created and trained for this project, to detect any disease in the plants. The sensor data is also passed to another algorithm to detect deficiencies of nutrients in the soil. Based on the results from the algorithms, appropriate amounts of fertilizers and pesticides are suggested to the farmers to be given to the plants via CropMates mobile app. In this section, both crop disease detection and nutrient deficiency detection methodologies are discussed.

## 3.1 Crop Disease Detection

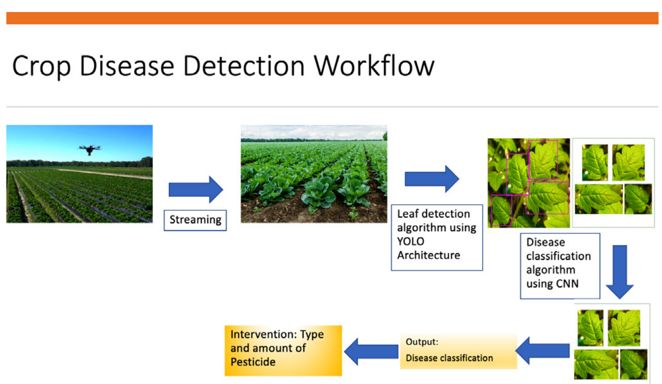



**Figure 3.** The methodology workflow for disease detection of the crops.

Disease detection from a drone image of farmland happens in two steps. In the first step a leaf is isolated from a top-view image of each part of the farm. In the second step, another algorithm takes these leaves as inputs and detects possible diseases in the crop.

An object detection algorithm is created to detect a leaf from a video or image of farmland taken by a drone flying over it. The object detection algorithm will output bounding boxes around the leaves and provide the coordinates respective to the bounding box. The object detection algorithm is trained with labeled images of leaves and is written in python using the OpenCV library and the YOLO (You Only Look Once) architecture using Google cloud's GPU. To make the data prepared for the object detection algorithm, each training image is run through an open-source labeling algorithm available on GitHub [6]. The labeling algorithm is a tool to label the bounding boxes of all the leaves for the object detection algorithm.

The output of the above algorithm is a collection of leaves. The images of these leaves are run through an object detection algorithm using Convolutional Neural Network (CNN) to see if there are any diseases in the plants. A CNN is a specialized deep learning model which is mainly used in the field of classifying images [7]. The machine learning model is created and trained with 87,000 images of 38 types of diseases and is written in python using the Scikit-learn library. The data is obtained from Plant Village, a non-profit organization with the goal of helping African farmers to manage their farms [8]. Scikit-learn is an open-source machine learning library for the Python programming language and features various classification, regression, and clustering algorithms, and contains the famous AlexNet architecture [9]. Alex-Net is a Convolutional Neural Network architecture that is known for its speed and accuracy [10]. Based on the diseases detected by the algorithm, appropriate pesticides will be given to the plants. This will be done by having a dataset that contains the 38 types of disease and the appropriate pesticides to fix each disease.

## 3.2 Nutrient Deficiency Detection

The following figure (Figure 4) shows the workflow of nutrient detection.

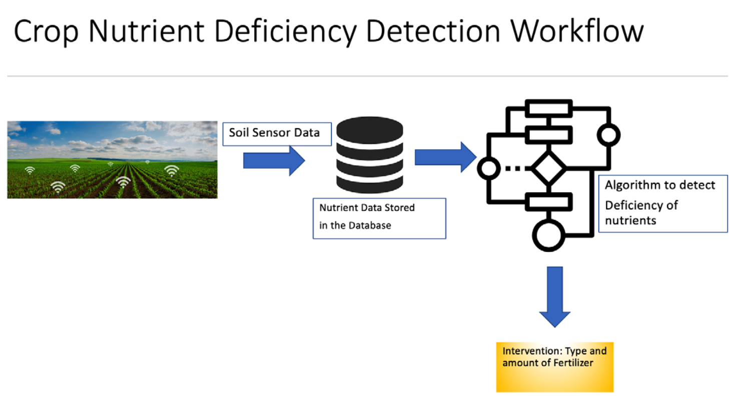



**Figure 4.** The methodology workflow for detecting deficiencies of the crops.

To detect deficiencies in plants, IoT-based nutrient sensors are placed in the soil all around the farm and then the outputs of the sensors are captured and analyzed by a machine learning model using an Artificial Neural Network to detect any deficiencies in the plants. The Artificial Neural Network (ANN) is trained with 20,000 cases of different micronutrients and macronutrients and moisture levels recorded in a farm from researchers and the corresponding fertilizer and quantity in pounds/hectare that should be given to crops for each case. Figure 5 shows the architecture overview of the ANN-based recommendation engine.

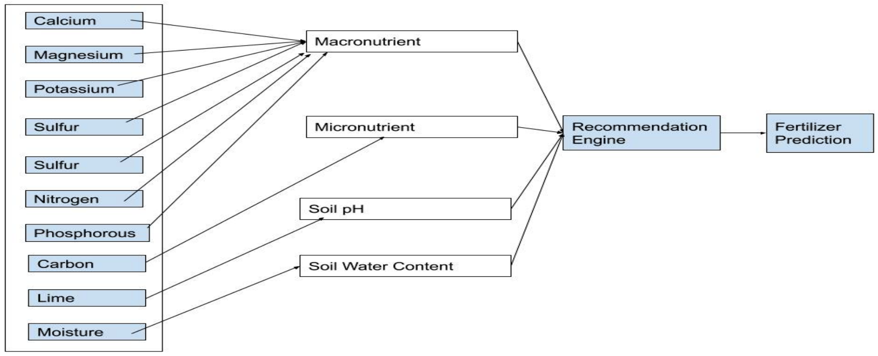



**Figure 5.** Architecture overview of prediction of fertilizer from nutrients in the soil.

### 3.2 Success Criteria

For the object detection algorithm to be successful crops. The object detection algorithm will be considered successful if it has an IoU >55% for 20 images. The IoU is calculated by taking the intersection of the overlapping area between the ground truth bounding box and the predicted bounding box divided by the area of the union [11]. For the CNN model for detection of disease from an individual leaf to be successful, the model has to have an accuracy score>85% for 17,400 testing images. The model for detection of fertilizer class has to have an accuracy score>80% for 4,000 testing nutrient and moisture level sensor data. The model for detection of fertilizer quantity has to have an R-2 score>.83 for 4,000 testing nutrient and moisture level sensor data.

## Results / Analysis of Results

### 4.1 Leaf detection from the drone-captured image of farm

The first part of the project is to detect leaves. To do this, an object detection algorithm for getting the images of the distinct leaves for the images or videos from the drone. The algorithm had an IoU of 77%. Figure 6 will show the python code and figure 7 will show the IoU score for each of the 20 images.

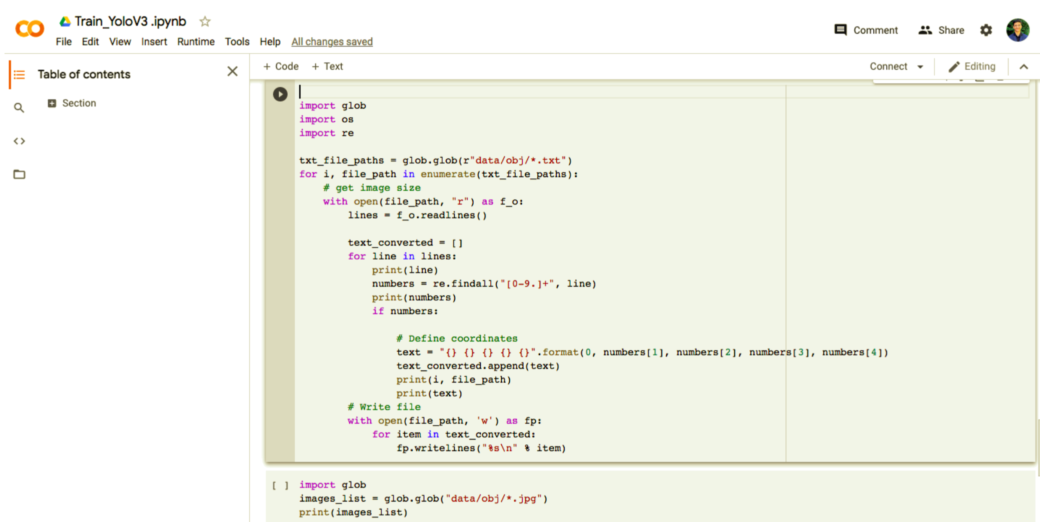



**Figure 6.** A Python code snippet for the object detection algorithm.

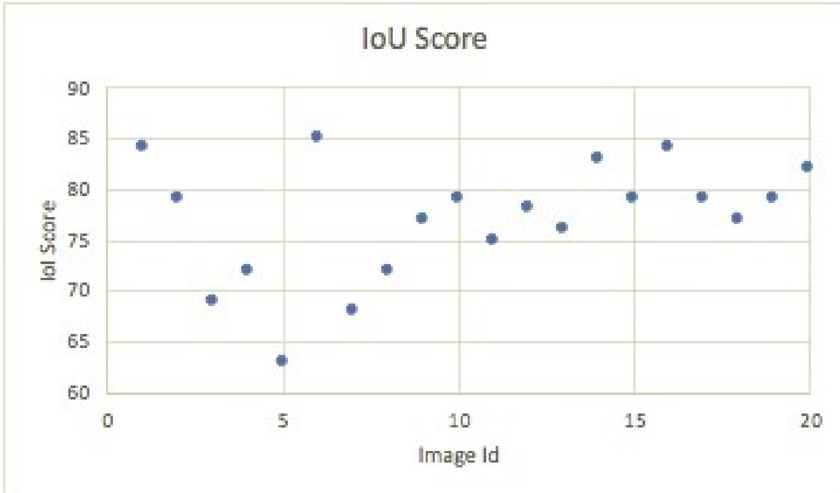



**Figure 7.** IoU score for each of the 20 images.

A drone captured the image of farmland every second. The image below is one case of an image running through the object detection algorithm. Figure 8 shows the input that will go into the object detection algorithm. Figure 9 shows the image with a leaf with the highest probability in a bounding box. Figure 10 shows the screenshot of the text file with the coordinates of that leaf. The figure below shows sample input and output for the object detection algorithm.


**Figure 8.** Drone image of tomato farmland. **Figure 9.** Bounding box around the leaf detected.

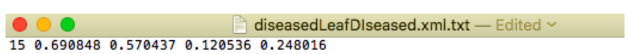



**Figure 10.** Coordinates of the bounding box shown in figure 6

## 4.2 Plant disease detection from the detected leaf

The plant disease detection model is built using Convolutional Neural Network and trained with 87,000 different images of leaves with or without diseases of 38 types. The model reached a testing/validation accuracy of 96.5% after 25 epochs of being tested on 17,400 images of leaves with diseases. Figure 11 is the snapshot of the Python code of the algorithm and Figure 12 shows the Python code to output a prediction. Figure 13 shows the accuracy of this model for each of the 25 epochs.

```python
from keras.layers.normalization import BatchNormalization

# Initializing the CNN
classifier = Sequential()

# Convolution Step 1
classifier.add(Convolution2D(96, 11, strides = (4, 4), padding = 'valid', input_shape=(224, 224, 3), activation = 'relu'

# Max Pooling Step 1
classifier.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2), padding = 'valid'))
classifier.add(BatchNormalization())

# Convolution Step 2
classifier.add(Convolution2D(256, 11, strides = (1, 1), padding='valid', activation = 'relu'))

# Max Pooling Step 2
classifier.add(MaxPooling2D(pool_size = (2, 2), strides = (2, 2), padding='valid'))
classifier.add(BatchNormalization())

# Convolution Step 3
classifier.add(Convolution2D(384, 3, strides = (1, 1), padding='valid', activation = 'relu'))
classifier.add(BatchNormalization())

# Convolution Step 4
classifier.add(Convolution2D(384, 3, strides = (1, 1), padding='valid', activation = 'relu'))
classifier.add(BatchNormalization())

# Convolution Step 5
classifier.add(Convolution2D(256, 3, strides=(1,1), padding='valid', activation = 'relu'))
```

**Figure 11.** Code to train the ml model for plant disease

```python
model = joblib.load('CnnPlantDiseaseModel5epoch_joblib')
history = joblib.load('historyCnnPlantDiseaseModel_joblib')

batch_size = 128

base_dir = "new-plant-diseases-dataset/New Plant Diseases Dataset(Augmented)"

train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   fill_mode='nearest')

training_set = train_datagen.flow_from_directory(base_dir+'/train',
                                                 target_size=(224, 224),
                                                 batch_size=batch_size,
                                                 class_mode='categorical')

class_dict = training_set.class_indices
#print(class_dict)
print('Class Dict:',class_dict)

li = list(class_dict.keys())
print('li:',li)
o
valid_datagen = ImageDataGenerator(rescale=1./255)

valid_set = valid_datagen.flow_from_directory(base_dir+'/valid',
                                              target_size=(224, 224),
```

**Figure 12**. Code to make a prediction

 Volume 10 Issue 4 (2021)

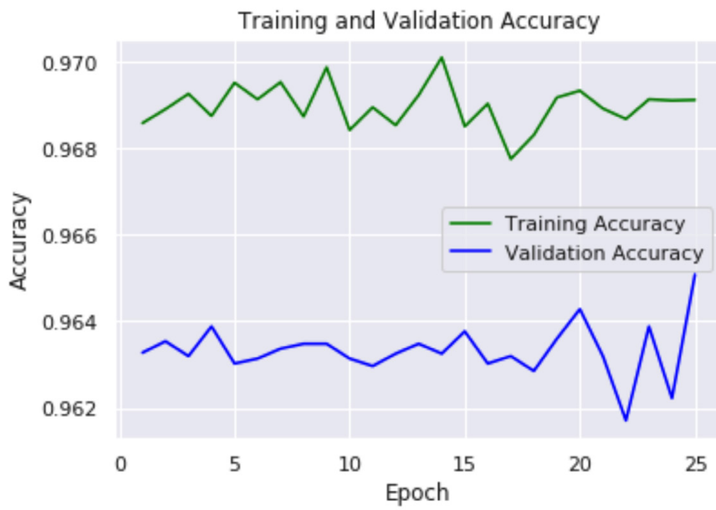


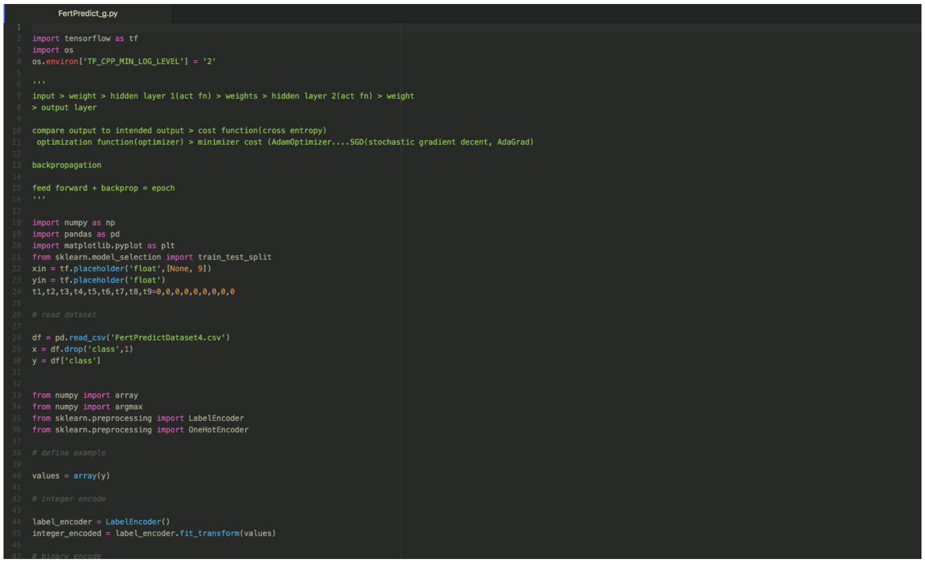

**Figure 13.** Accuracy of the model for each of the 25 epochs

The leaf is run through the plant disease detection model. Figure 14 shows a sample input of a leaf with a disease which is run through the plant disease model and figure 15 shows the output of the model.


**Figure 14**. The input of the plant disease model Figure 15: The output of the plant disease model.

## 4.3 Soil nutrient detection

The second part of the project is to detect nutrient deficiencies in the soil. A recommender engine is developed using the Artificial Neural Network and was trained with 20,000 training data points consisting of micronutrients, macronutrients, and moisture to recommend the appropriate fertilizer class and quantity. The model has an accuracy of 93% and an R-2 score of .95 after 25 epochs. Following is a snapshot of the Python program (Figure 16). Figure 17 and Figure 18 accuracy and R2 score vs epochs respectively.



```python
import tensorflow as tf
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

'''
input > weight > hidden layer 1(act fn) > weights > hidden layer 2(act fn) > weight
> output layer

compare output to intended output > cost function(cross entropy)
 optimization function(optimizer) > minimizer cost (AdamOptimizer....SGD(stochastic gradient decent, AdaGrad)

backpropagation

feed forward + backprop = epoch
'''

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
xin = tf.placeholder('float',[None, 9])
yin = tf.placeholder('float')
t1,t2,t3,t4,t5,t6,t7,t8,t9=0,0,0,0,0,0,0,0,0

# read dataset

df = pd.read_csv('FertPredictDataset4.csv')
x = df.drop('class',1)
y = df['class']


from numpy import array
from numpy import argmax
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

# define example

values = array(y)

# integer encode

label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(values)
```

**Figure 16.** Python code to train the neural network for predicting the appropriate fertilizer

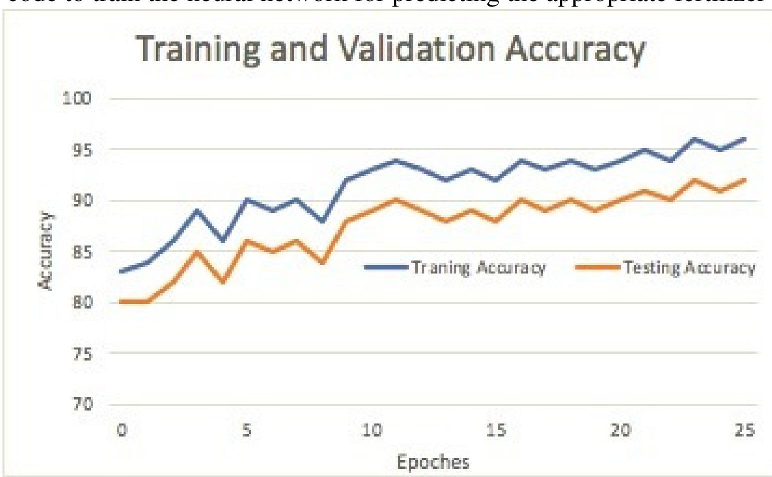



**Figure 17.** Accuracy of the model for each of the 25 epochs

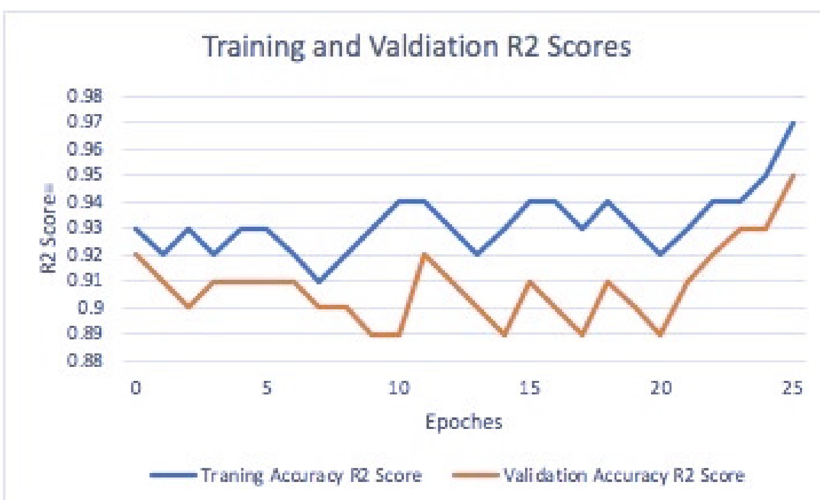



**Figure 18.** R-2 score of the model for each of the 25 epochs

Soil nutrients data are obtained from IoT-based sensors fitted in the soil. Following sensors are used:
3 in 1 Soil PH, Light and Water sensor (shown in the figure below)
Rapitest 4 in 1 npk sensor (shown in the figure below)

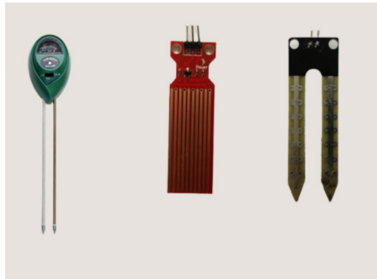


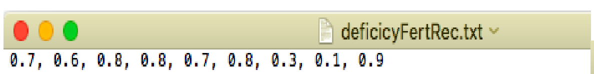
**Figure 19.** Soil Sensors

The IoT data is obtained by writing code in Arduino IDE. Figure 20 shows a snapshot of the program.

```
void setup() {
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");

}
void loop() {
 int soilmoisture = map(analogRead(A0),0,1023,100,0);
 Serial.println(soilmoisture);
 int ph = map(analogRead(A1),290,405,400,700)/100;


 String getData = "GET /update?api_key="+ API +"&field1="+String(soilmoisture)+"&field2="+String(ph);
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
}

void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
```

**Figure 20.** Arduino code for the soil sensors

Once the IoT data is obtained, these are passed to the machine learning model in a format depicted in Figure 21 Figure 22 shows the output of the model telling which fertilizer class to be used.





.

**Figure 21.** Input of the nutrient deficiency algorithm. **Figure 22.** Output of the nutrient deficiency algorithm

## Conclusion

The goal of the project was the development of a scalable and accurate IoT and AI-based integrated system for the detection and recommendation of plant diseases and deficiencies of nutrients in the soil so that the quality and quantity of crops are maximized. This also helps in effectively decreasing pesticide runoff and water usage. This project has achieved an accuracy of 96.5% for the detection of diseases in a plant and an accuracy of 93% and an R-2 Score of .95 for recommending the correct fertilizer class and quantity. This integrated system can be used by farmers to monitor the crops and remediate before any large-scale diseases happen to the farm. It also helps the farmers to monitor nutrients in the soil and help deliver the right fertilizer to the farm. Other researchers [12-13] have attempted to detect disease by analyzing leaves using image processing techniques.

Researchers [14] evaluated sensing technologies for nutrients of soils. However, this research is to develop an integrated system to detect plant diseases and detect deficiencies of nutrients in the soil at the same time to increase crop yields holistically and maximize crop output. This research has the potential to revolutionize farming and aims to reduce food scarcity in the world.

## References

1. United Nations. (2017, June 21). *World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100*. United Nations. https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html.
2. Berners-Lee, M., Kennelly, C., Watson, R., & Hewitt, C. N. (2018). Current global food production is sufficient to meet human nutritional needs in 2050 provided there is radical societal adaptation. Elementa: Science of the Anthropocene, 6. doi:10.1525/elementa.310
3. Leonard KJ, Szabo LJ. 2005. Stem rust of small grains and grasses caused by Puccinia graminis. Mol. Plant Pathol. 6, 99–111. (10.1111/j.1364-3703.2005.00273.x)
4. Frossard, E., Bucher, M., Mächler, F., Mozafar, A., & Hurrell, R. (2000). Potential for increasing the content and bioavailability of Fe, Zn, and Ca in plants for human nutrition. Journal of the Science of Food and Agriculture, 80(7), 861-879.
5. Römheld, V. (2012). Diagnosis of deficiency and toxicity of nutrients. In Marschner's Mineral Nutrition of Higher Plants (pp. 299-312). Academic Press.
6. Tzutalin, J. L., JL. (2000, April 09). Tzutalin/labelImg. Retrieved December 04, 2020, from https://github.com/tzutalin/labelImg
7. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). Ieee.
8. (2018, October 30). *PlantVillage Dataset*. https://www.kaggle.com/plantdisease.
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.
10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.
11. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 658-666).
12. M. A. Jasim and J. M. AL-Tuwaijari, "Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques," 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2020, pp. 259-265, doi: 10.1109/CSASE48920.2020.9142097.
13. Singh, V., & Misra, A. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture, 4*(1), 41-49. doi:10.1016/j.inpa.2016.10.005
14. Sinfield, J. V., Fagerman, D., & Colic, O. (2010). Evaluation of sensing technologies for on-the-go detection of macro-nutrients in cultivated soils. Computers and Electronics in Agriculture, 70(1), 1-18. doi:10.1016/j.compag.2009.09.017