JOURNAL OF
EMERGING INVESTIGATORS

Article

Time-Efficient and Low-Cost Neural Network to detect plant disease on leaves and reduce food loss and waste

Nehal Singh[1], Switi Singh[2]
[1]Coppell High School, Irving, Texas
[2]Senior Mentor

SUMMARY

About 25% of the food grown never reaches consumers due to spoilage, and 11.5 billion pounds of produce from gardens are wasted every year. Diseases are a major cause of food loss as Asia loses 14.2% of its crops due to disease, which is equivalent to 43.8 billion dollars as of 1988-1990. Current solutions involve farmers manually looking for and treating diseased crops. These methods of tending crops are neither time-efficient nor feasible. I used a convolutional neural network to identify signs of plant disease on leaves for garden owners and farmers. The overall accuracy goal of 90% or higher was set to develop the most accurate models. The learning rates of 0.1, 0.01, and 0.001 were tested for the network. I hypothesized that the step size at each iteration or the learning rate of 0.001 would result in the highest accuracy, which was supported through testing. By using machine learning Python libraries, the solution met the standard accuracy goal and was evaluated by several performance metrics, including precision, recall, f-score, specificity, and overall accuracy rates. The model had an accuracy of 95%. By uploading a picture of the crops into the highly-accurate neural network, farmers and gardeners can receive results in seconds on whether or not their crops have a disease, and if they do, which ailment specifically.

INTRODUCTION

About ¼ of the food grown never reaches consumers due to some form of spoilage which results in an economic loss, and about 40% of American-grown crops never reach consumers (1). Produce loss occurs due to physical injuries, disease, and pests. Additionally, about 11.5 billion pounds of crops produced in gardens, which is enough to feed 28 million people, contribute to food loss every year (1).

Worldwide, per capita availability of food is projected to increase, yet global demand for certain foods such as cereals is increasing, indicating the significance of saving food to meet the growing necessity. In addition, about 3 billion people have moderate food insecurity (2). More than 820 million people were hungry worldwide in 2018 (2). Specifically, in Eastern Africa, almost ⅓ of the population is malnourished (2). Due to having less food, problems in child growth are common (2). As recorded in 2018, approximately 20.5 million babies are born underweight (2). About 148.9 million children under five are stunted in height, and almost 49.5 million children under the age of five have a low weight relative to their height (2).

Regions across the world lost billions of dollars and a large percentage of agricultural production due to crop disease. In 2019, the Food and Agriculture Organization of the United Nations estimated that between 20%-40% of crops produced worldwide are lost due to pests (3). Additionally, $220 billion worth of produce is lost to plant diseases (3). One such bacteria includes the Xylella fastidiosa, which infects crops like olives, citrus fruits, and grapevines, and since 2015, it has been spreading globally in the Americas, Europe, and Asia (3). This one bacteria is the cause of a $104 million loss in wine in California annually and in 180,000 hectares of olive groves (3).

Crop diseases can be transmitted before the shipping, packaging, and consuming stage and instead at the stage of growing crops on a farm or garden. Once the disease has spread, little can be done to control it (4). Infections caused by bacteria, fungus, or pests can be spread by contact between diseased and healthy leaves (4). The disease can also spread within host tissues by grafting or transplanting branches or buds from diseased crops to healthy ones (4). Grafting can also be done by root grafts and with parasitic dodder (4). 50-60 viruses are dispersed by seed as well (4). For disease control, the infected crop must sometimes be destroyed and frequently quarantined or separated (4). Other measures include using insecticides to keep away pests and virus carriers (4).

Pests cause bacterial spots, fungal infections, or specks on leaves. These spots on the leaves interrupt photosynthesis and cause weakness within the plant (5). Over time, this condition causes leaf loss and death unless treated (5). This kind of condition is common, but in backyard gardens and large farms, identification is often missed or left untreated (5).

To solve limitations regarding time, convenience, and practicality of the traditional method of watching over the crops to make sure they grow, deep learning and machine learning (ML) solutions have been developed. Current deep learning models either prevent food from rotting and being thrown out or aim to reduce food loss at the transportation, shipping, or retail stage rather than at a farm or garden (6). Some deployed models are able to tell users when the fruit is ripe or weigh how much food is being thrown out (6). For example, Winnow Solutions is a smart trash for commercial kitchens and can use Artificial Intelligence to predict the cost of the food being thrown out in kitchens (6). This allows chefs to be more aware of the cost of their food waste to encourage less loss (6).

I developed a convolutional neural network that learns to identify signs of common diseases on different types of leaves, which would reduce the time needed and increase the accuracy of a diagnosis. The objective is to create the most efficient neural network using TensorFlow in Python with an overall accuracy of 90% or higher.

Figure 1: Classes in the Plant Village dataset. Figure showing the categories of healthy leaves and leaves with various diseases on bell pepper, potato, and tomato leaves.

RESULTS

The experiment was conducted using the Plant Village dataset available online on Kaggle (7). The Plant Village dataset contains images of healthy leaves and diseases on bell pepper, potato, and tomato leaves. (**Figure 1**).

I determined the most efficient network by comparing the learning rate, accuracy, time taken to converge, and confusion matrices. Once the best model was chosen, it was evaluated on several performance metrics (precision, recall, f-score, specificity, and accuracy rates) to meet the accuracy standard of 90% or above. I used the accuracy rate to evaluate this model because the precision, recall, and specificity rates only explain specific abilities of the model such as predicting true images as true or false images as false, and the f-score places more emphasis on false negatives, which is not a serious downside in this study.

Learning Rate	Accuracy (%)	Time Taken to Converge (hr)
.1	15%	30.29
.01	15%	21.52
.001	56%	28.2

Table 1: The effect of learning rate of the neural network on the accuracy and time taken to converge. Table comparing the learning rate used to the accuracy and time taken to converge. The learning rates of 0.1, 0.01. and 0.001 were tested to determine the rate that would allow for the most efficient and accurate model.

The 0.001 learning rate performed the best on the model. The 0.001 learning model had about a 273% increase in accuracy compared to the 0.01 or 0.1 model. The 0.001 model had an accuracy of 56% by the computer while the 0.1 and 0.01 models were at a 15% accuracy rate. The time taken to converge is not correlated with the learning rate. Although the model that used a 0.001 learning rate took more time to process, the accuracy rate is higher. Therefore, it can be concluded that the 0.001 learning rate was the best fit for this neural network as it resulted in the highest accuracy without sacrificing time-efficiency (**Table 1**).

The model with the learning rate of 0.1 predicted all images as Tomato Yellow Leaf Curl Virus. The high inaccuracy rate would not allow the model to be beneficial in the real world for farmers and other potential users. Therefore, this could not be the final model (**Figure 2**).

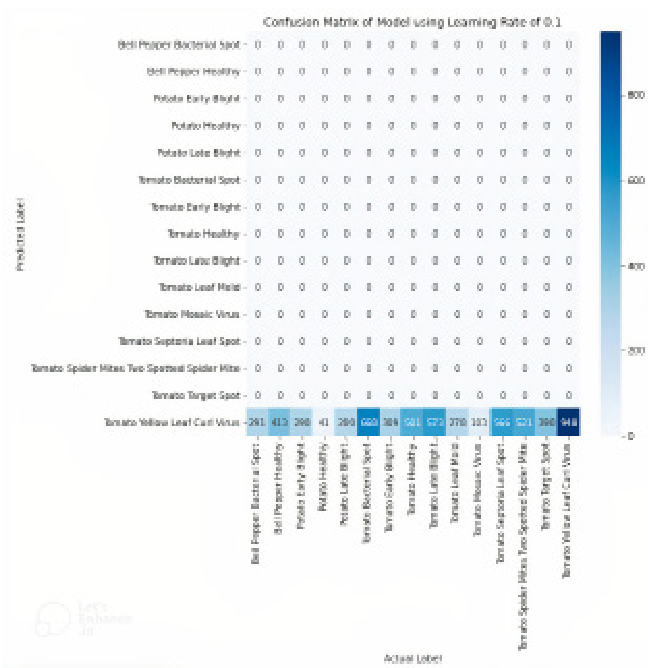

Figure 2: Confusion matrix of model using learning rate of 0.1. Confusion matrix showing the predicted and actual class of each image in the testing data set. The learning rate of .1 was tested to determine the rate that would allow for the most efficient and accurate model.

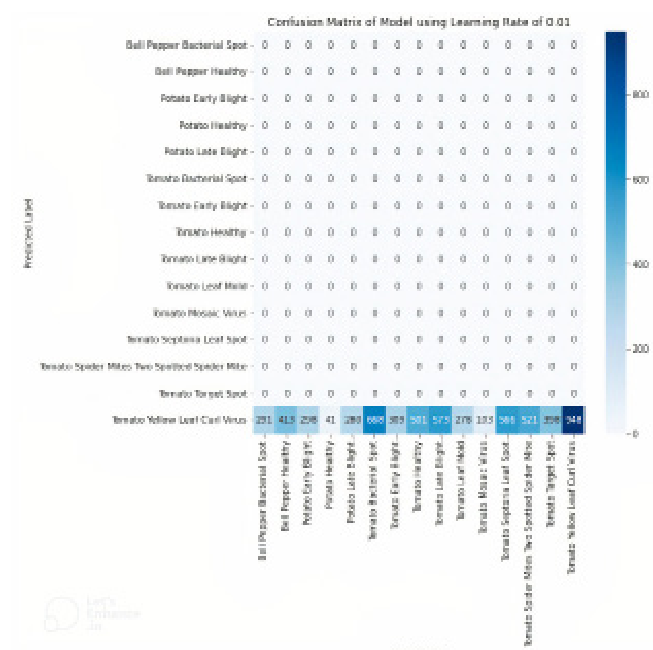

Figure 3: Confusion matrix of model using learning rate of 0.01. Confusion matrix showing the predicted and actual class of each image in the testing data set. The learning rate of .01 was tested to determine the rate that would allow for the most efficient and accurate model.

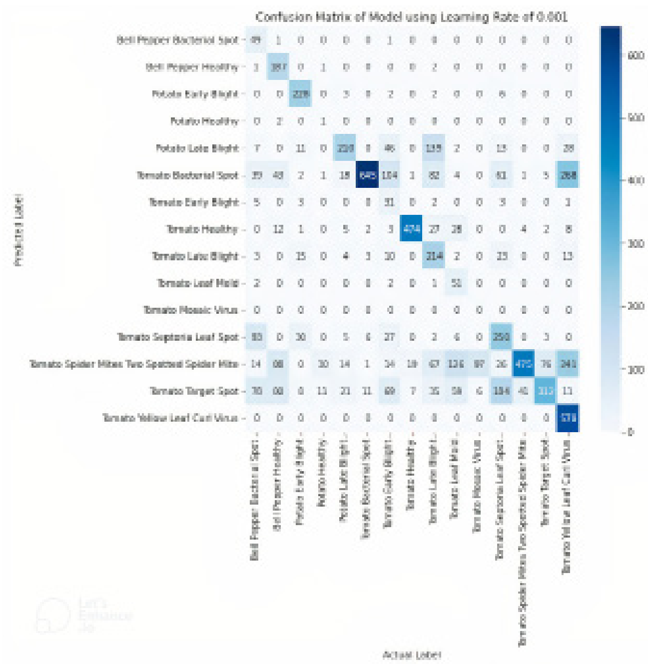

Figure 4: Confusion matrix of model using learning rate of 0.001. Confusion matrix showing the predicted and actual class of each image in the testing data set. The learning rate of .001 was tested to determine the rate that would allow for the most efficient and accurate model.

The model using the learning rate of 0.01 also predicted all images as Tomato Yellow Leaf Curl Virus. This not only gave the model a low accuracy of 15%, but it also made the model insufficient in the real world where every disease is not the Tomato Yellow Leaf Curl Virus. It could not be the final model (**Figure 3**).

Comparatively, the model with the lowest learning rate of 0.001 was able to be trained for the highest accuracy. The model did not predict any images as having the Tomato Mosaic Virus and predicted few images as healthy potato leaves. This model could be improved in specific classes like the Potato Healthy, Tomato Mosaic Virus, and other classes with a high false negative and false positive rates (**Figure 4**).

DISCUSSION

Most classes had accuracy rates in the high 90 percentages, and the specificity scores were also relatively high, ranging from the high 80s to the high 90s. These rates could be due to the high number of images that do not belong to the class being analyzed. These are referred to as negative or false images. In addition, the classes had either a low precision or recall rate. The classes with the lowest precision rates include the Tomato Mosaic Virus, Tomato Spider Mites Two Spotted Spider Mite, Tomato Septoria Leaf Spot, and the Potato Late Blight classes. These classes all had precision rates below 50%, indicating a high number of false positives or that a small amount of the images predicted as that class are actually true and positive. However, the Bell Pepper Bacterial Spot, Bell Pepper Healthy, Potato Healthy, Tomato Early Blight, Tomato Late Blight, Tomato Leaf Mold, Tomato Mosaic Virus, Tomato Septoria Leaf Spot, and Tomato Yellow Leaf Curl all had recall rates below 50%. These classes were weak in predicting true images as positive. Overall, the final model

Class	True Positive	True Negative	False Positive	False Positive
Bell Pepper Bacterial Spot	49	5898	2	242
Bell Pepper Healthy	187	5774	4	225
Potato Early Blight	228	5880	13	70
Potato Healthy	1	6145	2	43
Potato Late Blight	210	5665	246	70
Tomato Bacterial Spot	645	4894	629	23
Tomato Early Blight	31	5868	14	278
Tomato Healthy	474	5598	92	27
Tomato Late Blight	214	5920	73	359
Tomato Leaf Mold	51	5908	5	227
Tomato Mosaic Virus	0	6088	0	103
Tomato Septoria Leaf Spot	250	5453	172	316
Tomato Two Spotted Spider Mite	475	4857	813	46
Tomato Target Spot	312	5181	612	86
Tomato Yellow Leaf Curl Virus	378	5243	0	570

Table 2. Class statistics on diseases on leaves neural network. Table displaying the true positive, true negative, false positive, and false negative values of each class in the model. Once the best model was chosen, the true positive, false positive, true negative, and false negative of each class was recorded.

with the learning rate of 0.001 had a high overall accuracy and specificity rates, the weakness lies in the precision and recall of classes described above (**Tables 2-3**).

Class	Precision	Recall	F-Score	Specificity	Accuracy
Bell Pepper Bacterial Spot	96%	17%	.28	100%	96%
Bell Pepper Healthy	98%	45%	.62	100%	96%
Potato Early Blight	95%	77%	.85	100%	99%
Potato Healthy	65%	2%	.04	100%	99%
Potato Late Blight	46%	75%	.57	96%	95%
Tomato Bacterial Spot	51%	97%	.66	89%	89%
Tomato Early Blight	69%	10%	.18	100%	95%
Tomato Healthy	84%	95%	.89	98%	98%
Tomato Late Blight	75%	27%	.40	99%	93%
Tomato Leaf Mold	91%	21%	.34	100%	96%
Tomato Mosaic Virus	0%	0%	0	100%	98%
Tomato Septoria Leaf Spot	59%	44%	.51	97%	92%
Tomato Two Spotted Spider Mite	37%	91%	.53	86%	86%
Tomato Target Spot	34%	78%	.47	89%	89%
Tomato Yellow Leaf Curl Virus	100%	40%	.57	100%	91%

Table 3. Class evaluation metrics on diseases on leaves neural network. Table displaying the precision, recall, f-score, specificity, and accuracy rates for each class in the model. Using the true positive, false positive, true negative, and false negative of each class to calculate the evaluation metrics.

Precision	67%
Recall	48%
F-Score	.46
Specificity	97%
Overall Accuracy	95%

Table 4. Overall evaluation metrics of diseases on leaves neural network. Table displaying the precision, recall, f-score, specificity, and accuracy rates of the model overall. The mean or macro-average of the precision, recall, f-score, specificity, and accuracy rates among the classes was calculated.

I concluded that the model accurately predicts 95% percent of images, indicated by the overall accuracy. The model precisely evaluated 67% of predicted positive images and only 48% of true testing data is predicted true. The low precision and recall rates resulted in a low f-score of .46 as the false negative rate or false positive rate was low for each class. However, the model proved that it can predict 97% of false images as negative (**Table 4**).

In the neural network, there was little data and images available. This small amount of data does not accurately represent the potential of the model and resulted in inaccurate precision, recall, f-score, specificity, and accuracy rates. The dataset is unbalanced, meaning it does not have about the same number of testing and training images between classes, which also returned imprecise evaluation metrics for each class. In the dataset, some classes like the Tomato Mosaic Virus folder had 400 images while others such as the Tomato Yellow Leaf Curl Virus folder had 3000 images. Additionally, the learning rate test was done on one kind of model with a set number and types of hidden layers. The current layers used may not be the right layers to help the model learn best.

Any future research on this innovation would require more data. Insufficient data do not train the model to its full potential. Using more images would reduce significant data size unbalance and would create a more efficient neural network. The current data is unbalanced when regarding images in each class, which resulted in imprecise evaluation metric scores. These inaccuracies can be improved through data augmentation by applying a rotation or crop, to create new data images. This augmentation would create more data that is also more balanced and will prevent overfitting the network as well. Furthermore, to find the model and layers that help the neural network perform at its best, testing can be done with the number and types of hidden layers such as Dense, MaxPooling2D, Conv2D, and Dropout layers.

Lastly, this innovation can be expanded to detect diseases on a larger variety of leaves through more data, and the same technology can be applied to signs of infection on fruits. The farmers and gardeners can simply take a picture of the plants and upload the pictures into the models for quick results in less than a second.

Knowing if a certain plant is diseased quicker and earlier could save surrounding crops and fresh produce. Additionally, it would reduce the amount of food and fresh produce disposed of due to bacteria, fungi, and pests, potentially helping feed the increasing population and minimizing the money lost at farms. This procedure would not require any unnecessary manual work as it would normally. The solution is time-efficient, feasible, and cost-efficient. The innovation can be utilized to ultimately reduce food loss in a faster and more functional way.

MATERIALS AND METHODS

After downloading the Plant Village dataset on Kaggle, an online machine learning and data science community (7), Python libraries needed were downloaded using pip install in the terminal for os, Numpy, Matplotlib, Pandas, Tensorflow, Keras, cv2, Seaborn, Math, and Glob in Jupyter Notebook. Next, a bar graph was coded to depict the total number of images in each class or category of potential classification. The training images were prepared by resizing them to 256 by 256 pixels and allowing for rotations, shifts, and zooms to randomly occur on data. The TensorFlow model had 12 hidden layers. The first two layers were Conv2D layers with 32 filters each. The third layer was a MaxPool2D layer followed by a Dropout layer. The next 4 layers consisted of 2 Conv2D layers of 64 filters each, one MaxPool2D layer, and another Dropout layer. The last four layers were 2 Conv2D layers of 128 filters each, one MaxPool2D layer, and another Dropout layer. Next, the training and testing data were split into a ratio of 7:3, and the model used the Root Square Mean Propagation optimizer.

After preparing for training, the code was copied into three different notebooks with different learning rates. Learning rate is a hyperparameter that controls how much to change the model in response to the error each time the weights change. This determines the step size at each iteration while moving towards less loss. By testing for the learning rate that results in the highest overall accuracy, the model can be engineered to be more accurate. The number of hidden layers, epochs, and other testable variables can stay the same. Then there will only be one testing variable, and the learning rate will be tested for the model to accommodate the predetermined layers and other variables. Low learning rates result in longer times to converge. High learning rates result in divergence or in the model skipping the best configurations. One notebook has a learning rate of .1, another has .01, and the last one has .001. Training was done in 30 epochs with 14,445 steps each, and testing was completed with testing dataset.

The most efficient network was determined by comparing the learning rate, accuracy, and time taken to converge and by analyzing the confusion matrices illustrating the predicted and actual class of each testing sample. The time taken to converge in hours is calculated by adding the time taken to complete each epoch. Once the best model was chosen, the true positive, false positive, true negative, and false negative of each class was recorded, and the precision, recall, f-score, specificity, and accuracy rates for each class were calculated. By taking the average of each metric, a table showing the overall evaluation metrics was created. If the goal for the accuracy of 90% or above was not met, the network would have been modified.

ACKNOWLEDGEMENTS

The author would like to thank their parents for their support throughout this study.

Received: March 11, 2022
Accepted: May 22, 2022
Published: April 24, 2023

REFERENCES

1. "Garden Food Waste." *AmpleHarvest.org*, 4 Aug. 2021, ampleharvest.org/2016-nationwide-study-of-garden-food-waste/.
2. "World Hunger Is Still Not Going down after Three Years and Obesity Is Still Growing – UN Report." *World Health Organization*, World Health Organization, www.who.int/news-room/detail/15-07-2019-world-hunger-is-still-not-going-down-after-three-years-and-obesity-is-still-growing-un-report.
3. "New Standards to Curb the Global Spread of Plant Pests and Diseases." *FAO*, www.fao.org/news/story/en/item/1187738/icode/.
4. "Transmission." *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., www.britannica.com/science/plant-disease/Transmission.
5. Grabowski, Michelle. "Leaf Spot Diseases of Trees and Shrubs." *UMN Extension*, extension.umn.edu/plant-diseases/leaf-spot-diseases-trees-and-shrubs.
6. Rejcek, Peter. "Food Waste Is a Serious Problem. Ai Is Trying to Solve It." *Singularity Hub*, 31 Oct. 2019, singularityhub.com/2019/11/03/food-waste-is-a-serious-problem-ai-is-trying-to-solve-it/.
7. Emmanuel, Tairu Oluwafemi. "Plantvillage Dataset." *Kaggle*, 30 Oct. 2018, www.kaggle.com/emmarex/plantdisease.