Spring 2022

Remote Crop Disease Detection using Deep Learning with IoT

Ivy Chung

Anoushka Gupta

SANTA CLARA UNIVERSITY

Department of Electrical and Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Ivy Chung, Anoushka Gupta

ENTITLED

REMOTE CROP DIESEASE DETECTION USING DEEP LEARNING WITH IOT

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL AND COMPUTER ENGINEERING

June 3, 2022

Thesis Advisor(s) date

June 7th, 2022

Department Chair(s) date

REMOTE CROP DISEASE DETECTION USING DEEP LEARNING WITH IOT

By

Ivy Chung, Anoushka Gupta

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical and Computer Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical and Computer Engineering

Santa Clara, California

Spring 2022

Remote Crop Disease Detection using Deep Learning with IoT

Ivy Chung, Anoushka Gupta

Department of Electrical and Computer Engineering
Santa Clara University
2022

ABSTRACT

Agriculture is such a vital part of our society, and according to the United Nations' Food and Agricultural Organization (FAO), plant diseases are considered one of the two main causes of decreasing food availability. This paper explores not only the methods and findings of building a CNN-based disease detection model, but that of building a deployable remote crop disease detection system incorporating IoT technology. By using transfer learning with AlexNet, we were able to predict with 89.8% accuracy tomato plant images into one of the ten pre-defined disease classes. Our proposed system tracks plant health throughout the day by using a microprocessor and a camera to automatically capture images, diagnose the plant, and report results. The system is a proof of concept of a technology that can significantly help increase crop yield, reduce food waste, and automate the tasks of detecting and caring for diseased crops.

Keywords: Deep Learning, Agriculture, Plant Disease, CNN, IoT, Disease Identification, Machine Learning

Acknowledgments

We would like to express our appreciation and gratitude to our advisor, Dr. Tokunbo Ogunfunmi, for all the support and guidance he has given us. He has helped us on many fronts: from proof-reading our work to pushing us to strive towards bigger project goals with his encouragement.

We would also like to thank the Electrical and Computer Engineering Department, Dr. Shoba Krishnan, Dr. Andrew Wolfe, and Dr. Cary Yang, for providing us with the necessary tools and steppingstones that enabled us to make steady progress towards our final senior design goal.

We are grateful to Santa Clara University School of Engineering for funding this project and facilitating the Senior Design Conference so that we can showcase our project and gain valuable feedback from industry professionals.

Special thanks go to all the interviewees- Samuel Bertram from OnePointOne, SCU's Forge Garden, and Ariana Reyes and Aparna Gazula from UCANR - who gave us great insights and perspectives into the current agricultural environment during our research phase of our project.

Finally, we owe our thanks to our friends and family for all their support and enthusiasm for our project.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

1.1 Technology and Agriculture

Smart Farming is a fast-growing and evolving concept that involves the creation and development of novel methods to meet the challenges of the modern agricultural sector. We must preface by defining what exactly constitutes Smart Farming or Farming Automation in agriculture- it is the process by which various technological innovations are utilized to upgrade and automate the many processes in agriculture that pose challenges to farmers around the world. Using Smart Farming methodologies, farmers can acquire more time and resources to manage and redirect into their farms for better overall growth that would greatly benefit food production.

With how our society is dependent upon agriculture today, protecting and improving the practices of agriculture would produce significant effects globally. This is the reason why smart farming is taking a leap in America and why it is the solution our society is moving towards (see Figure 2). By integrating deep learning into agricultural practices, our society can benefit greatly from farms that have more food security and climate sustainability [1].

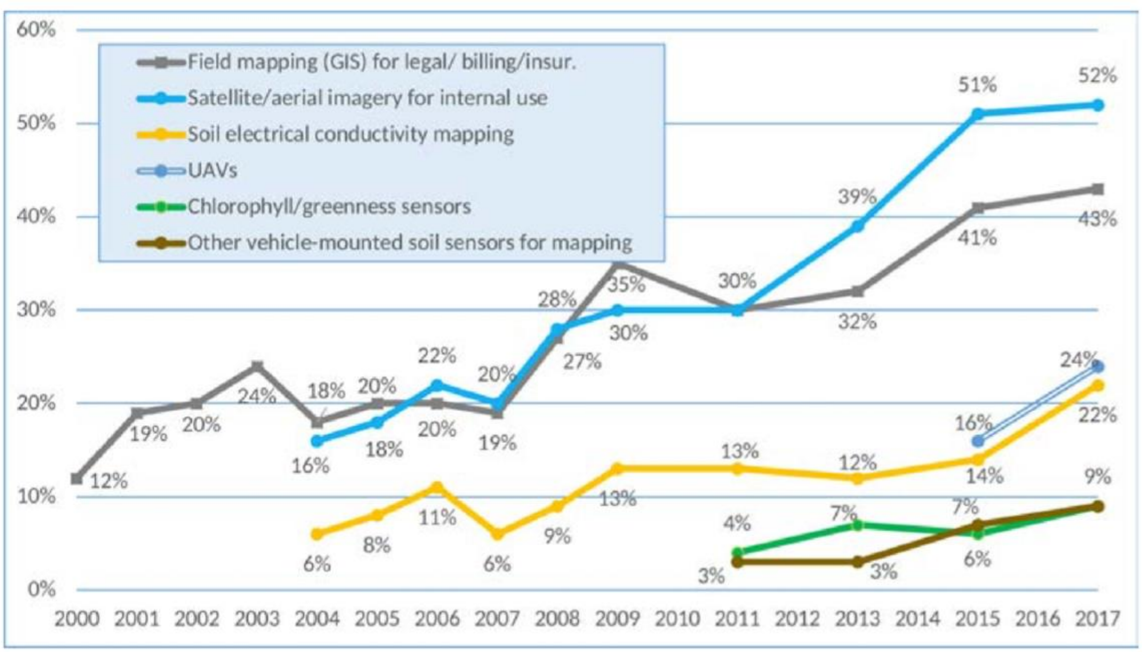


Figure 1. Usage of digital technology in agriculture in the United States of America over time [2]

One such instance in which farmers have found themselves devoting several hours of hard labor into is the planting of crops on large plots of land. This process has been made easier by applying robotics and drone technology to plant and water seeds from both the ground and the air. An example for such an innovation is DroneSeed [3], which is a planting drone that is used to help reforest large spaces that have been affected by forest fires. This drone is capable of locating and targeting the precise areas on the ground that are ideal for seeding. Then, they use compressed air to shoot the seeds onto the ground without allowing them to scatter because of the wind.

One such challenge faced by farmers that requires them to spend enormous amounts of time is the monitoring and analyzing the conditions for their crop manually. A proposed solution to such an issue is seen in the application of machine learning, more specifically, deep learning-based diagnosis that has the potential to help farmers monitor their crops after the planting and growing phase has been completed. Deep Learning is a subset of Machine Learning that is a rapidly growing field, forming itself to be an integral tool to aid image-based classifications of various input objects. Our goal with our project endeavor is to adapt a Deep Learning model with our custom image database to classify input leaf photographs taken by our camera integrated IoT system, resulting in plant health diagnosis and thereby, automating the crop monitoring process through this integration of software and hardware processes.

1.2 Motivation

Agriculture is one of the most important segments of our society, providing us with food, clothing, shelter, and is “identified as one of the pathways to achieve the Zero hunger goal of the United Nations Sustainable Development Goals" [4]. According to the UN, the current world population of 7.6 billion is expected to reach 8.6 billion in 2030, 9.8 billion in 2050 and 11.2 billion in 2100. Therefore, it is extremely necessary to find ways to reduce costs and increase production and product quality. To gain a better, more comprehensive understanding of the current challenges faced in agriculture, let us examine the existing agricultural environment in both developed and developing nations.

The biggest trials faced by the current global agricultural sector is food insecurity. An interesting fact as brought up by the World Bank [5] is that the impact of rising food prices is greater on people in low- and middle-income countries than on people in high-income countries since the former groups spend a larger share of their income on food compared to the latter.

A specific and significant issue stemming is acute food insecurity, which is defined as a person's life or livelihood being in immediate danger because of the lack of food. Recent data from the Global Network Against Food Crises [6] estimates that a whopping 161 million people experienced "crisis" levels of acute food insecurity in 2021, which is an approximate increase of 4% over the previous year. Furthermore, 227 million people were estimated to be experiencing "stressed" acute food insecurity in 2021, which is a 7% increase from the prior year.

Numerous nations are undergoing high food price inflation on a retail level, which reflects labor shortages, currency devaluations, a steep spike in fertilizer costs and other such factors. Focusing specifically on fertilizer prices, these surged in March of this year, up nearly 20% since January 2022 and almost three times higher than in 2021 [5]. High fertilizer costs pose a serious problem, adversely affecting the next food production cycle. A potential solution seen to this issue would involve periodic, early-stage diagnosis of the plant that would allow farmers to plan ahead in terms of fertilizer purchasing and crop maintenance to maximize their resource usage by minimizing how much they would spend on fertilizers, water, and other irrigation aids. Our project tackles this application of predictive health diagnosis, with our deep learning model focusing specifically on tomato plants. However, the training database can be adapted to work with leaf images of other kinds of food crops.

1.3 Tomato Crop Statistics

Tomatoes are one of the world's most cultivated and consumed crops and the most popular vegetable in the world. The annual global production of tomatoes is 177.04 million metric tons and the largest producers are China, India, and the United States [7]. Today, over 200 pests and diseases have been identified in tomato plants, which lead to significant production losses both directly and indirectly. Illnesses caused by fungi, viruses, bacteria, and nematodes have severe implications on the crop's nutritional value, as well as the overall health of consumers and the economy. Some examples of such diseases seen in tomato plants are Late Blight, Bacterial Leaf Spot, Yellow Leaf Curl Virus, and Pepino Mosaic Virus.

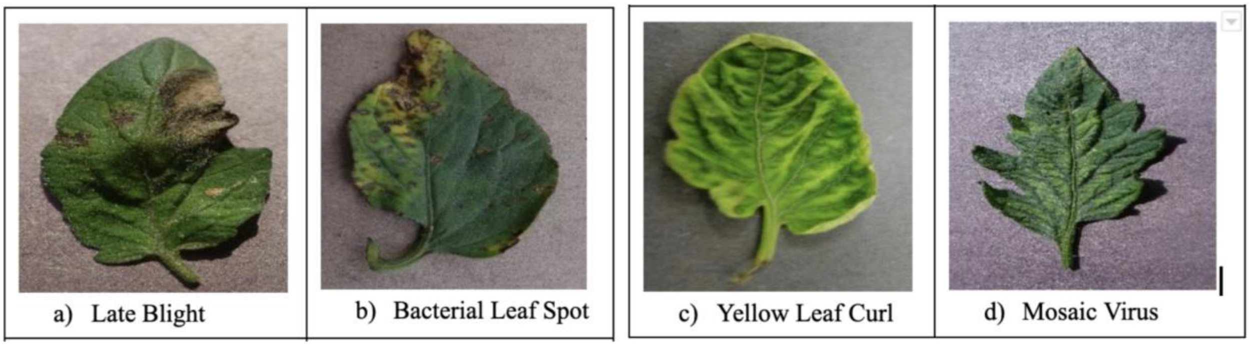


Figure 2. Four examples of disease classes from training dataset

Late Blight is caused by the Phytophthora infectants and is one of the most destructive diseases in tomatoes, resulting in major economic loss from 20% to 70% [8]. Bacterial Leaf Spot is a common bacterial disease caused by Xanthomonas campestris. It is highly damaging in both greenhouse and field environments, resulting in a 10-15% loss [9]. In India, tomato production loss has an estimated range of 10-80% whereas annual production loss due to Bacterial Leaf Spot is 10-20% which might rise to 80% in some drastic cases [10]. Two major viral diseases are the Yellow Leaf Curl Virus and the Pepino Mosaic Virus. The former disease is seen in tropical and subtropical regions worldwide and losses up to a full 100% are most frequent, making the Yellow Leaf Curl one of the most dangerous illnesses that hinders and limits tomato production. The Mosaic Virus is a rapidly emerging viral illness that has now been establishing itself as a significant disease towards tomato crops [11].

Our tomato image database comprises 10 classes, with one class being the “Healthy” class and the rest of the 9 classes being a type of tomato crop illness, 4 of which are the Tomato Late Blight, Bacterial Leaf Spot, Yellow Leaf Curl Virus and Mosaic Virus. Our disease detection model can distinguish between the various diseases since it has been trained using leaf images being fed into our custom image classifying AlexNet implementation. This enables the user to load an image of a leaf on their tomato crop into the system, which would output the disease classification result, thereby informing the user of the illness their plant likely has.

Chapter 2 – Background

2.1 Deep Learning and CNN

First, what exactly is machine learning? Simply put, machine learning is a subset of Artificial intelligence that is “based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention” [12]. This is extremely helpful in situations that allow you to use historic data to improve prediction and assessments, scenarios that we can’t program by hand such as autonomous driving, and times when we want customizations that are tailored to each user [13].

Deep learning builds on top of this technology. It is not as widely known but is the technology behind many inventions that people credit as the work of machine learning. The difference between deep learning and machine learning is that with deep learning, the learning algorithm is made up of “a hierarchy of concepts. Each concept is defined in terms of its relation to simpler concepts” [14]. Deep learning utilizes machine learning algorithms to solve simple problems so that the computer can learn complex concepts.

Dechter introduced the concept of using deep learning in machine learning in 1986, and Aizenberg et al. introduced deep learning to artificial neural networks later in 2000 [15]. This introduction of neural networks into the architecture of machine learning algorithms has been proven to be impactful as currently, “convolutional Neural Networks (CNNs) are the most widely used architectures to classify images” [4]. A simple CNN is a sequence of layers (convolutional, pooling and fully connected layers) stacked to form a full CNN.

To allow for effective classification by our Deep Learning model that uses CNN’s, we decided to populate our image database with photographs of tomato leaves. This can also be expanded to using images of fruits and the whole plant itself, as shown in the image indicator figure below, which displays the various metrics and parameters we are looking at for each component of the tomato plant.

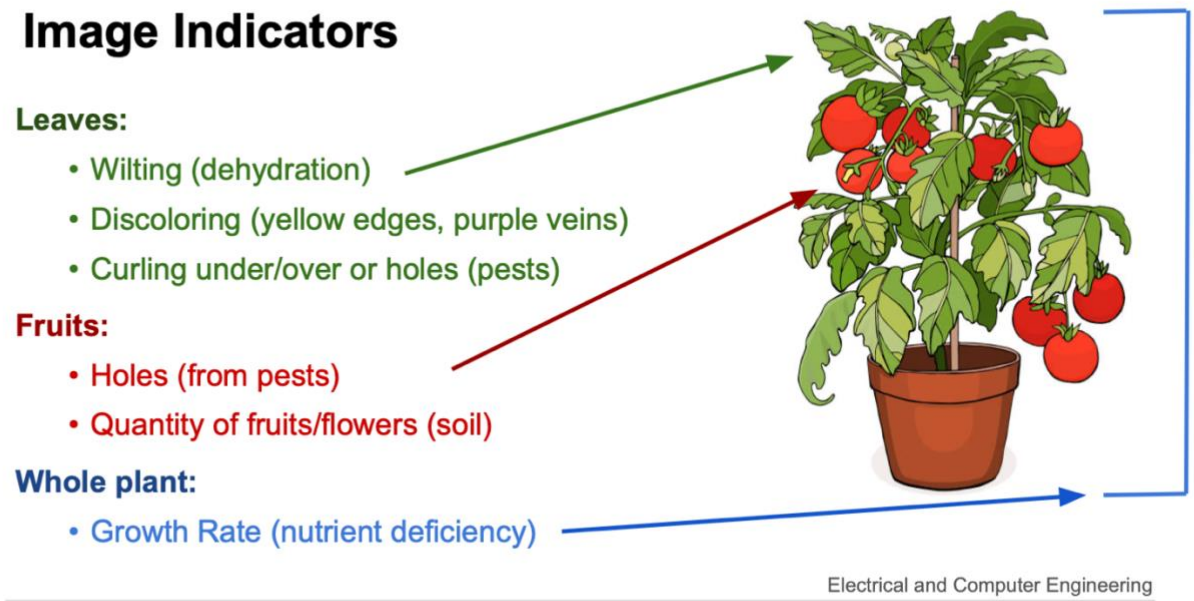


Figure 3. Image indicators to identify the state of the plant

2.2 Supervised and Transfer Learning

Supervised learning is done by making predictions based on previous evidence from experimental procedures outside of the machine itself. It involves training with labeled data and making predictions on unlabeled data and consists of classification and regression. Classification is done when there exists discrete output classes and regression is done when the output classes are continuous. Our project uses supervised learning as it relies on giving a network labeled data and characteristics to classify unlabeled data.

Transfer learning is a process by which knowledge is transferred from one supervised learning task to another and this process is commonly used in deep learning applications. Transfer learning enables us to take a pretrained network and using that as a starting point, allows the machine to learn a new task. Fine tuning such a network with transfer learning is much faster and easier to do than training a network with weights that are randomly initialized from scratch.

The pretrained network we would be utilizing and repurposing for our project is the AlexNet CNN, which can perform classification on new collections of images, or in our case, a custom dataset of images. AlexNet has been trained over a million of images, thus allowing it to learn rich representations of a wide range of images. The AlexNet network takes an image as an input and obtains an output as a label for the object in the image together with probabilities for each object class category [16].

2.3 Literature Review

Identifying crop diseases using CNN image classification models is an emerging research area. This paper [17] reviews more than 200 literature published on deep learning in agriculture, and found most of the research successful, with the livestock branch showing “most of the works achieved an accuracy above 95%” [17].

Building on the success of these work, our proposed project incorporates hardware to a deep learning model to further the research on deployable systems. This paper [14] explores state-of-the-art machine learning methods that are used for plant disease detection. It finds the main formats of data acquisition to be IoT ground imaging, unmanned aerial vehicle imaging, and satellite imaging, with IoT ground imaging to have “guarantee high detection accuracy thanks to the close level leaf imaging” [14]. Given IoT ground devices’ ability to add other sensors such as humidity and pH sensor, we decided to pursue our research on an IoT ground system to collect image data.

Chapter 3 – Design and Methodology

3.1 General Overview

With the goal to create an effective, affordable, and easily deployable solution, we designed an automated system which includes taking images of the crop with a Raspberry Pi HQ camera attached to a NVIDIA Jetson Nano GPU processor, processing the images through a trained AlexNet model, and recording the results on Cloud.

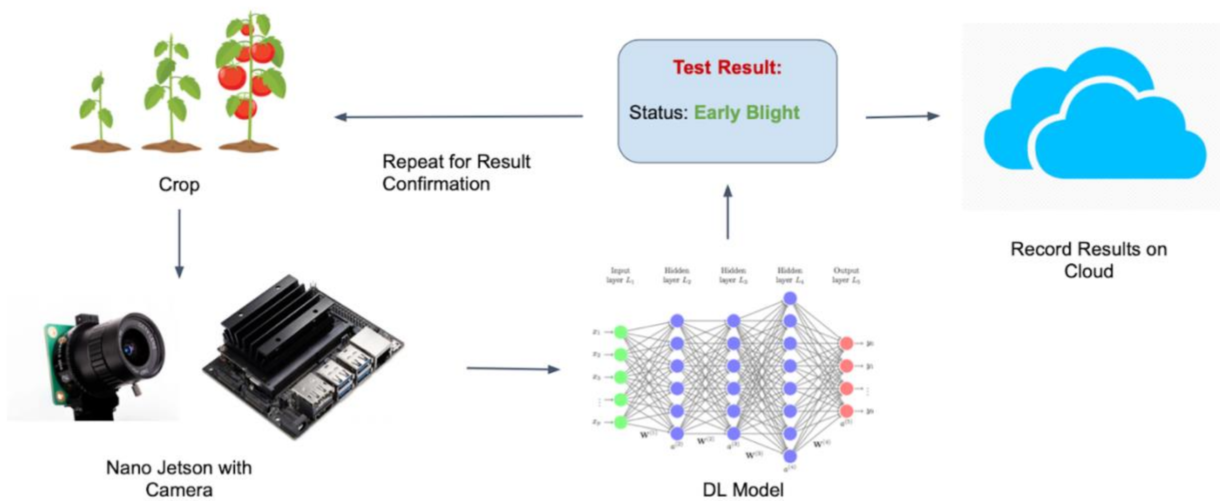


Figure 4. System design block diagram

3.2 Dataset Collection

One of the most crucial steps in training an effective deep learning model is to acquire a quality dataset. The proposed research work uses a dataset from PlantVillage, which contains leaf images of 4 different Tomato plant species. These images are divided into 10 subcategories: healthy, spider mites, Septoria leaf spot, leaf mold, late blight, bacterial spot, mosaic virus, yellow leaf curl virus, early blight, and target spot. The diseases present in the dataset are some of the most common Tomato plant diseases.

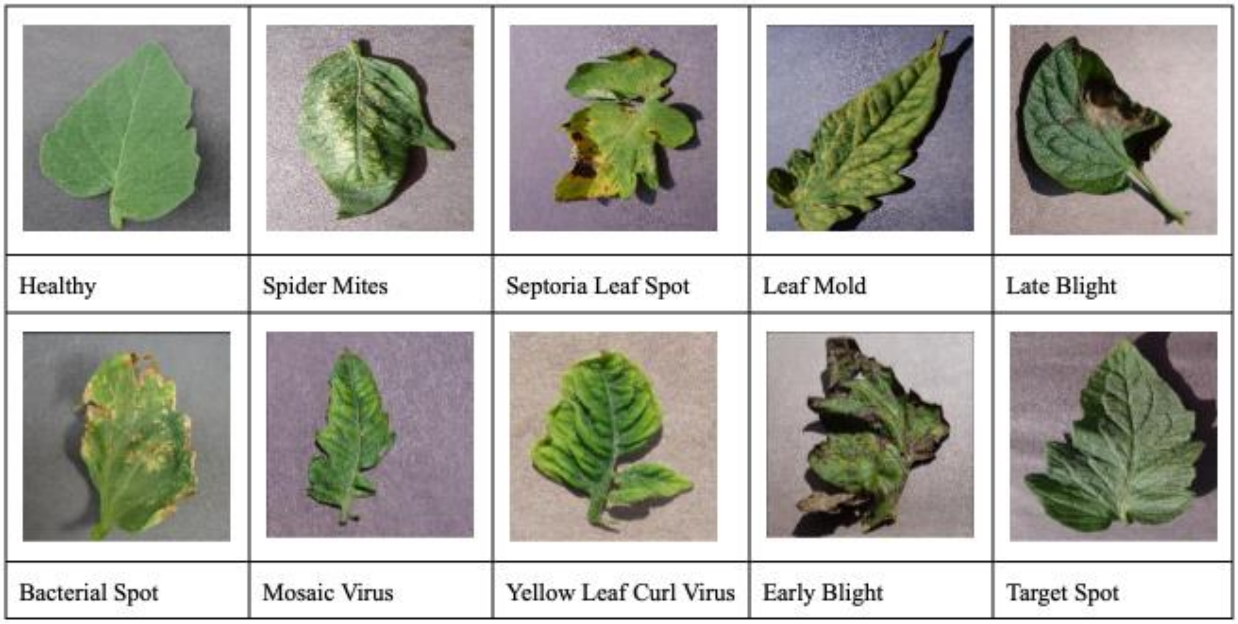


Figure 5. Different tomato disease classes in our dataset

3.3 Data Pre-Processing

Before the images are fed into the AlexNet model to train, they are pre-processed by color standardization and resizing. The images are first resized to 64 x 64 pixels to fit the transfer learning model. They are then pixel-wise standardized using Tensorflow's per_image_standardization function, which linearly scales the pixels in each image to have mean 0 and standard deviation 1, and drastically reduces training time.

$$For\ each\ x\ in\ image, compute: \frac{x\ - mean}{\max(\ stddev, \frac{1.0}{sqrt(N)})}$$

$$Where\ N\ is\ the\ number\ of\ elements\ in\ x\ and\ stddev\ is\ the\ standard\ deviation\ of\ all\ values\ in\ x \qquad \text{(eq. 1)}$$

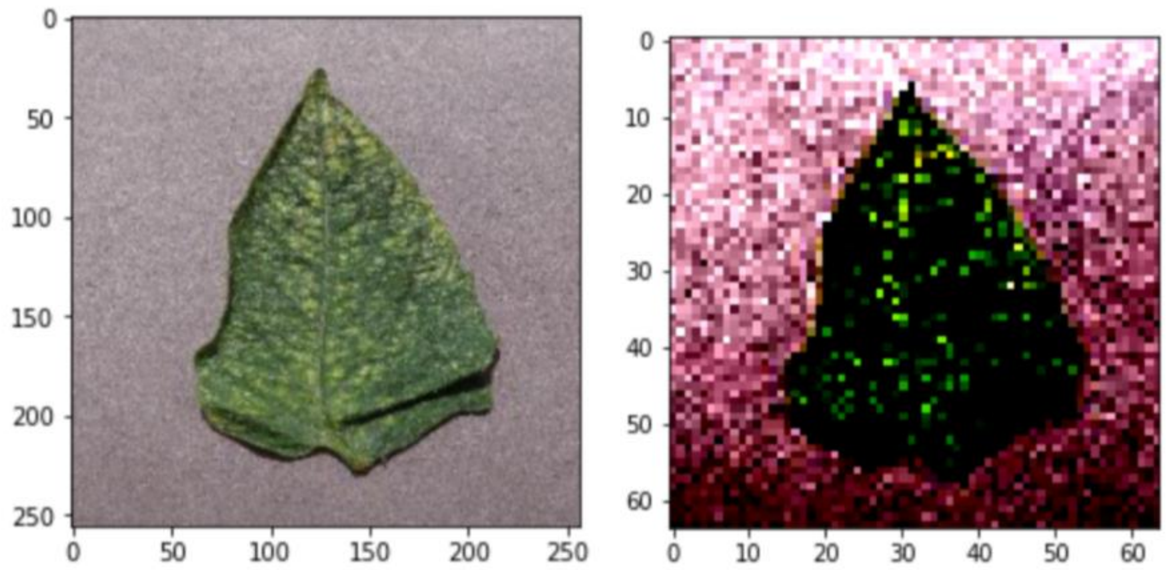


Figure 6. Image before and after standardization

After resizing and standardization, the images are split so that 75% becomes the training set and 25% becomes the testing set. For each class we have between 1702 to 1961 images to train, and between 425 to 490 images to test.

	Healthy	Spider Mites	Septoria Leaf Spot	Leaf Mold	Late Blight	Bacterial Spot	Mosaic Virus	Yellow Leaf Curl Virus	Early Blight	Target Spot
Train	1926	1741	1745	1882	1851	1702	1790	1961	1920	1827
Test	481	435	436	470	463	425	448	490	480	457

Table 1. Number of Images Used to Train and Test each Disease Class

3.4 Training the AlexNet

To create the system, we performed transfer learning on a trained AlexNet model, which has 5 Convolutional Neural Networks, 3 Fully Connected Layers, and a Softmax Layer. This AlexNet model is trained on Google Colab and serves as our supervised classification model. We compiled the AlexNet model with an optimizer set to a learning rate of 0.001. To fit the model with the optimal epoch count, we used Tensorflow Keras' callback EarlyStopping function, which tracks the validation accuracy until it reaches the maximum value, then restores the model to the best weights. Using this method, we have found that the most optimal epoch count is around 30. By testing out different batch sizes, we have settled on using the batch size of 32 to train our implementation of AlexNet.

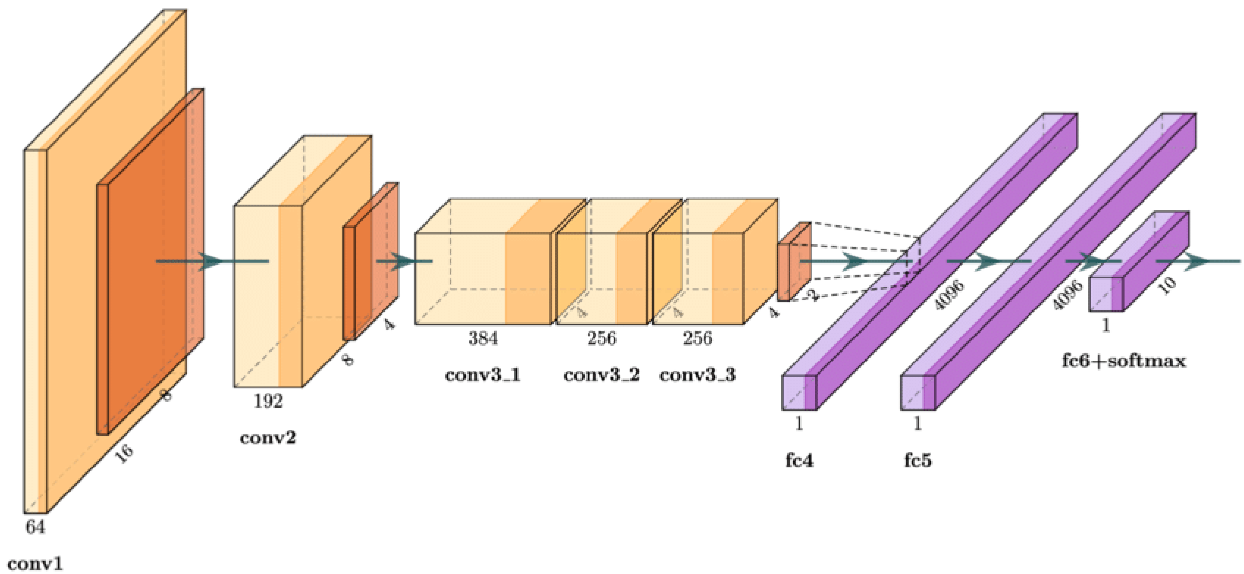


Figure 7. AlexNet Model [18]

3.5 IoT System

All of the hardware in our IoT system are off-the-shelf and easily accessible products. We used NVIDIA Jetson Nano 4GB as our processor and the center of the proposed IoT system. The small computer has a 128-core NVIDIA Maxwell™ architecture-based GPU, a Quad-core ARM® A57, and a 4 GB 64-bit LPDDR4 RAM [19]. We added connectivity to the microprocessor by installing an Intel Dual Band Wireless-Ac 8265 card, which allows the system to communicate through Wifi and Bluetooth. We have also added 2 RP style antennas to provide a 6dbi of signal gain. On the software side, we installed various drivers and changed configurations on the Jetson Nano in order to allow for Wifi connectivity and communication through the secure shell protocol. These changes make the IoT system fully configurable and controllable remotely. To back up data and improve usability, we connected certain Google Drive folders to the system for images and classification results to be automatically transferred to Cloud.

Another critical component in our IoT system is a Raspberry Pi HQ camera, which takes high quality images even with lower ambient lighting. With one resistor removed, the camera is fully compatible with the Jetson Nano. In our testing, we used JOBY GorillaPod as a flexible camera tripod.

Figure 8. IoT system setup

Chapter 4 – Results

4.1 Our First CNN Model

The first implementation of CNN in our proposed system was one that we implemented from scratch using Tensorflow and Keras. It has 3 total layers: 2 convolution and max pooling layers and one fully connected and dropout layer as shown in the figure below. For this model, we used Adam optimizer, and achieved a validation accuracy of 98.5% in classifying whether the leaf shows signs of a disease. Using Tensorflow Keras' callback EarlyStopping function, we have found the optimal amount of epoch to be around 12. The figure below shows the performance of this model through different epochs.

We have ultimately found this CNN model to be insufficient when we expanded the classification to more than the 2 classes of classifying an image to either healthy or unhealthy. Using this model to classify tomato leaves into one of the ten classes we have set up, we only achieved a validation accuracy of 53.6%. We then implemented K-fold cross validation with 10 folds in hopes to improve the result. With the K-fold cross validation implemented, we improved the validation accuracy to 74.4% with the cost of a significantly increased training time.

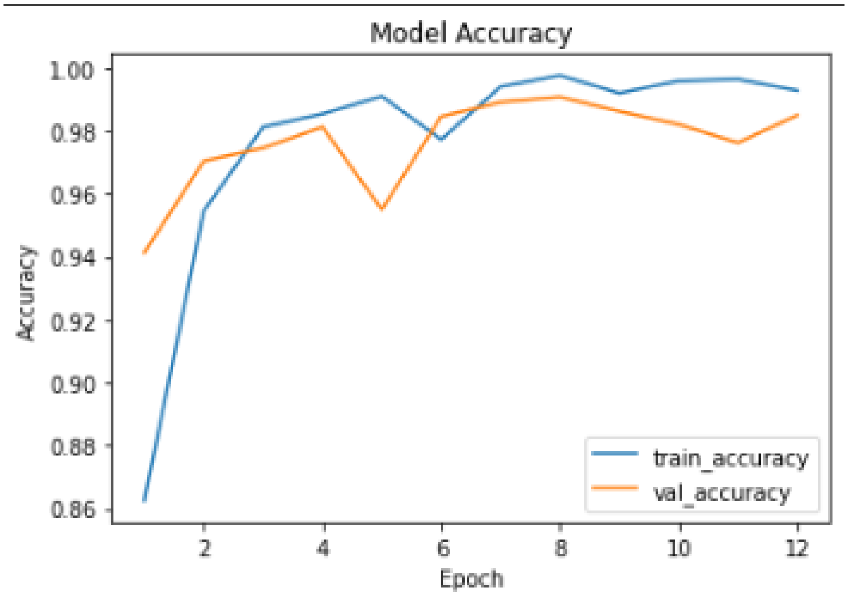


Figure 9. Two-class classification performance with our first CNN model

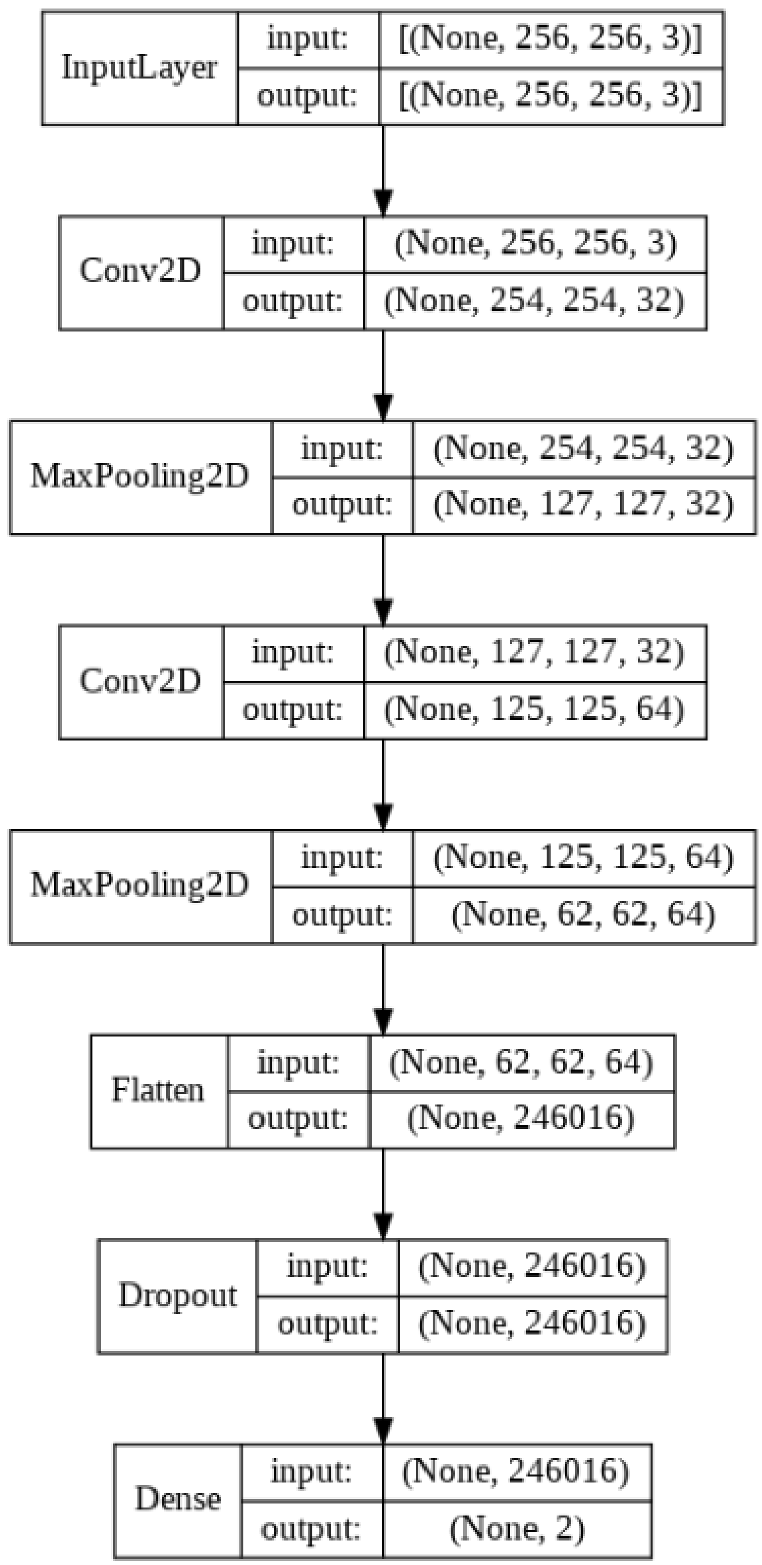
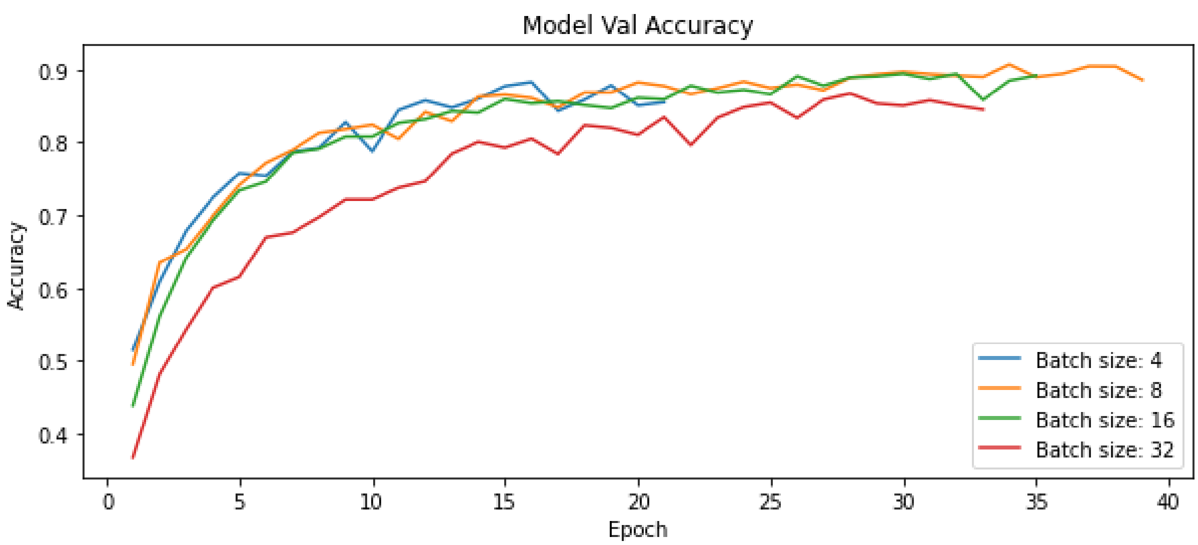


Figure 10. Our own CNN implementation

4.2 AlexNet Implementation

Our final CNN implementation uses transfer learning with a trained AlexNet model. This implementation was able to achieve a training accuracy of 98.1% and a validation accuracy of 89.8%. We experimented our model with batch sizes of 4, 8, 16, 32, 64, 128, and 256. We have found the most optimal batch size to be 8. Testing our system with unseen images from our dataset, we were able to consistently get high accuracy predictions.



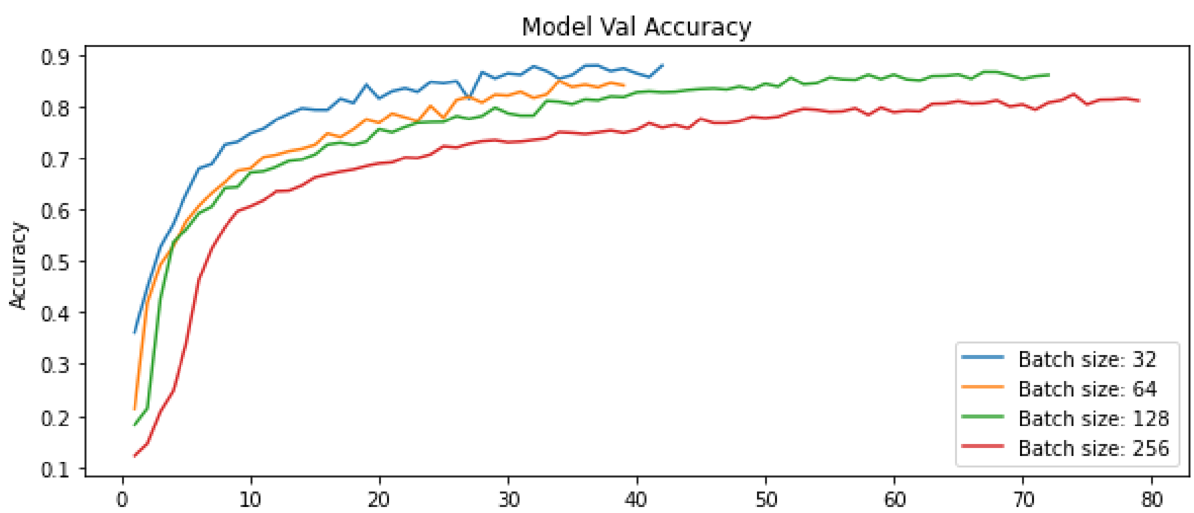


Figure 11. Experimenting with various batch sizes (each graph stopping at the optimal epoch count)

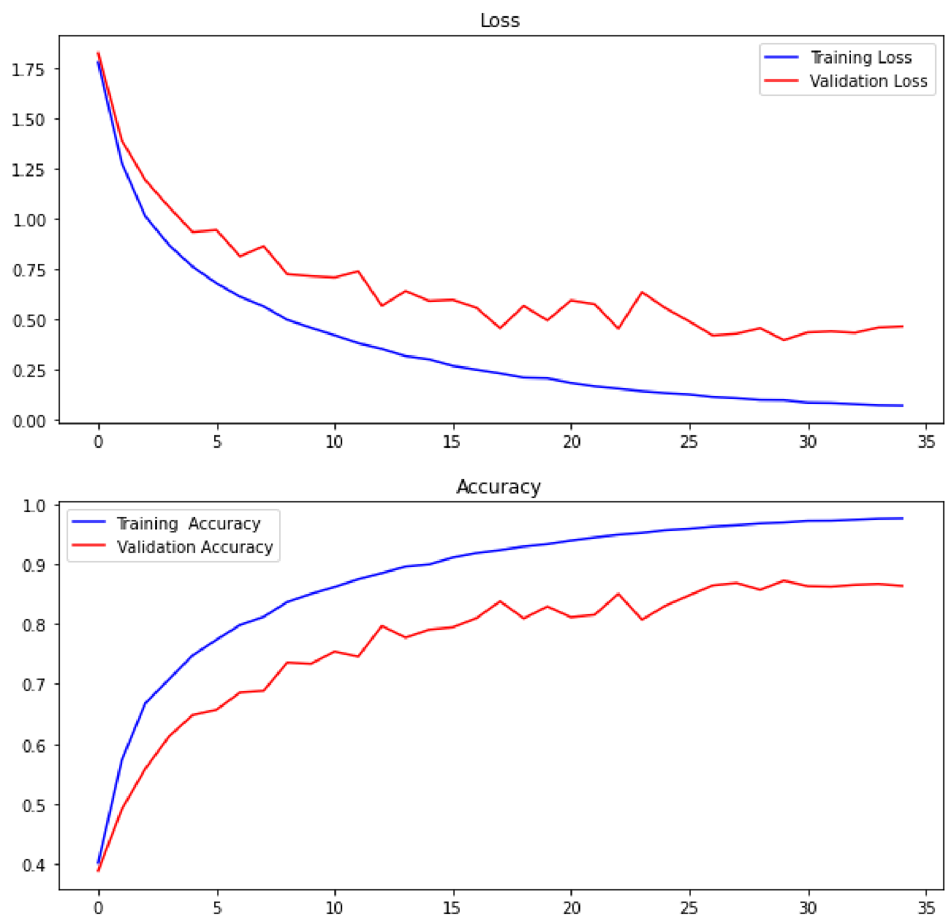


Figure 12. Final model performance in specifying disease

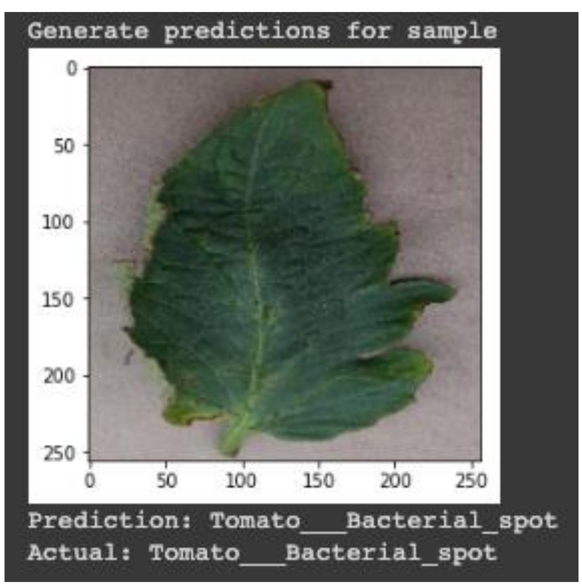


Figure 13. Sample image prediction

4.3 Testing with Real Images

To test the validity of our system, we re-trained and tested our AlexNet model with images captured through our embedded system. We performed image data augmentation by randomly flipping and rotating the images to populate our dataset with more images than we were able to collect. For each original image we captured, we resulted in 9 augmented images.

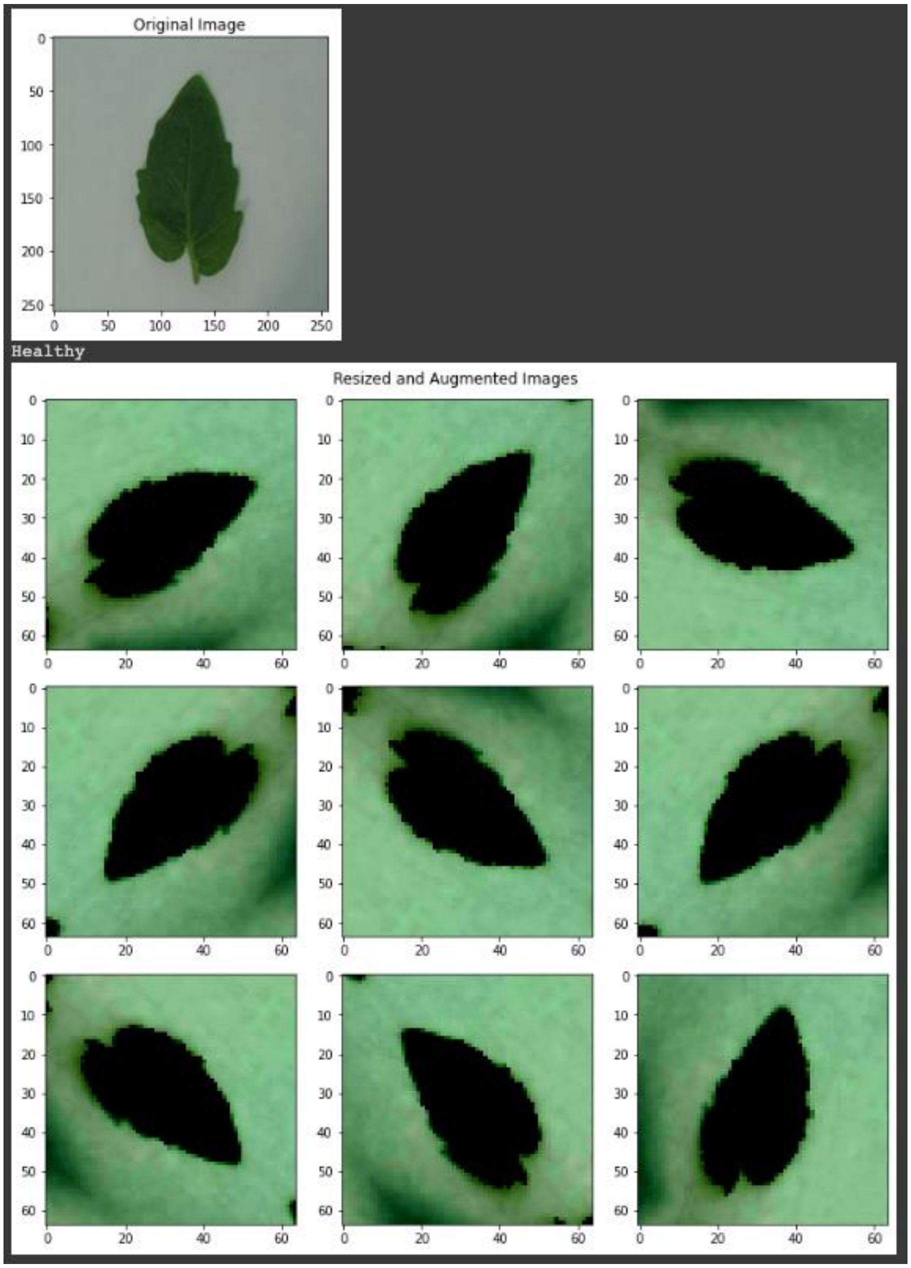


Figure 14. Nine images are produced from each original image

We found that our model was able to correctly classify the leaves as healthy or unhealthy with a 76.2% test accuracy and a 68.2% test loss. The higher-than-expected test loss and the lower-than-expected test accuracy is to do with the images in our training dataset not being diverse enough with lightings, leaf positions, and background. We are building a larger dataset to improve real-life system implementation results.

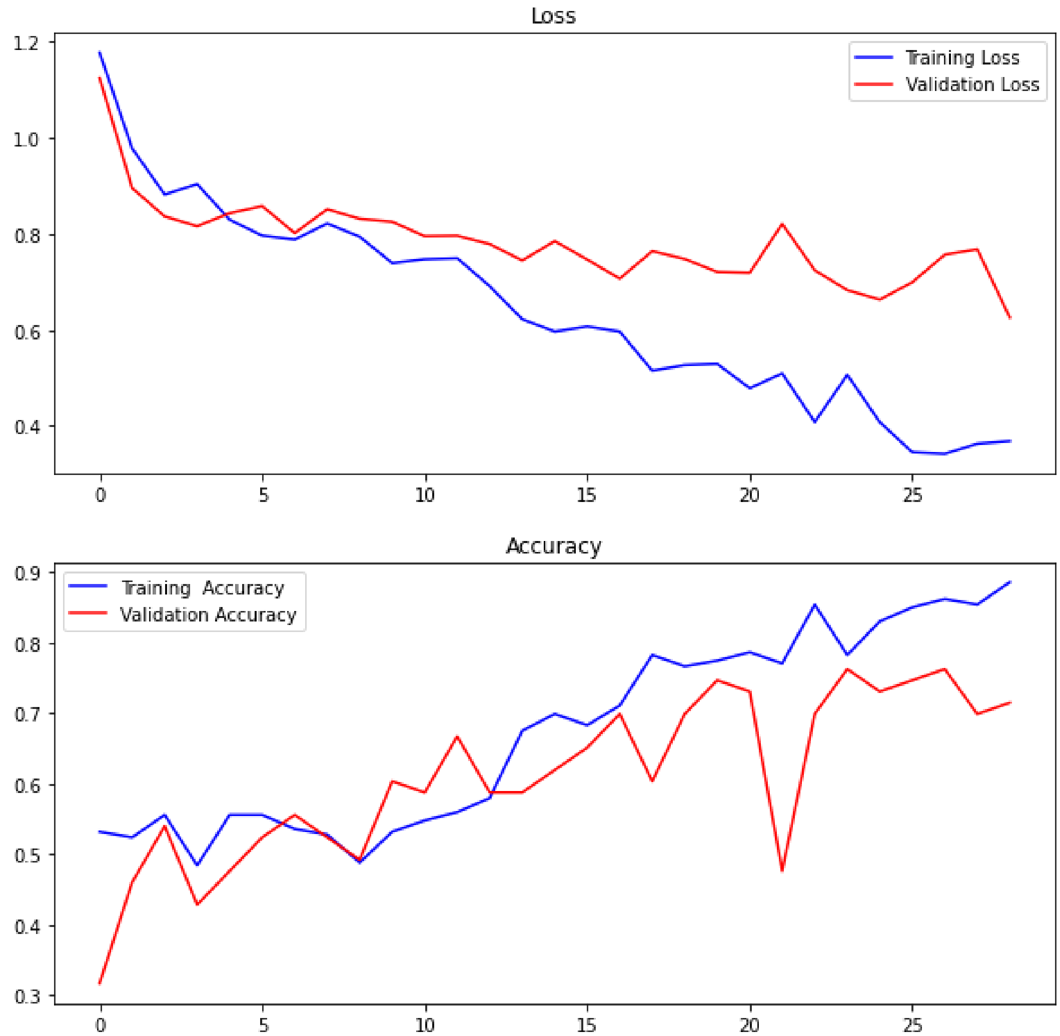


Figure 15. Test accuracy and loss of our AlexNet model trained with images captured through our IoT system

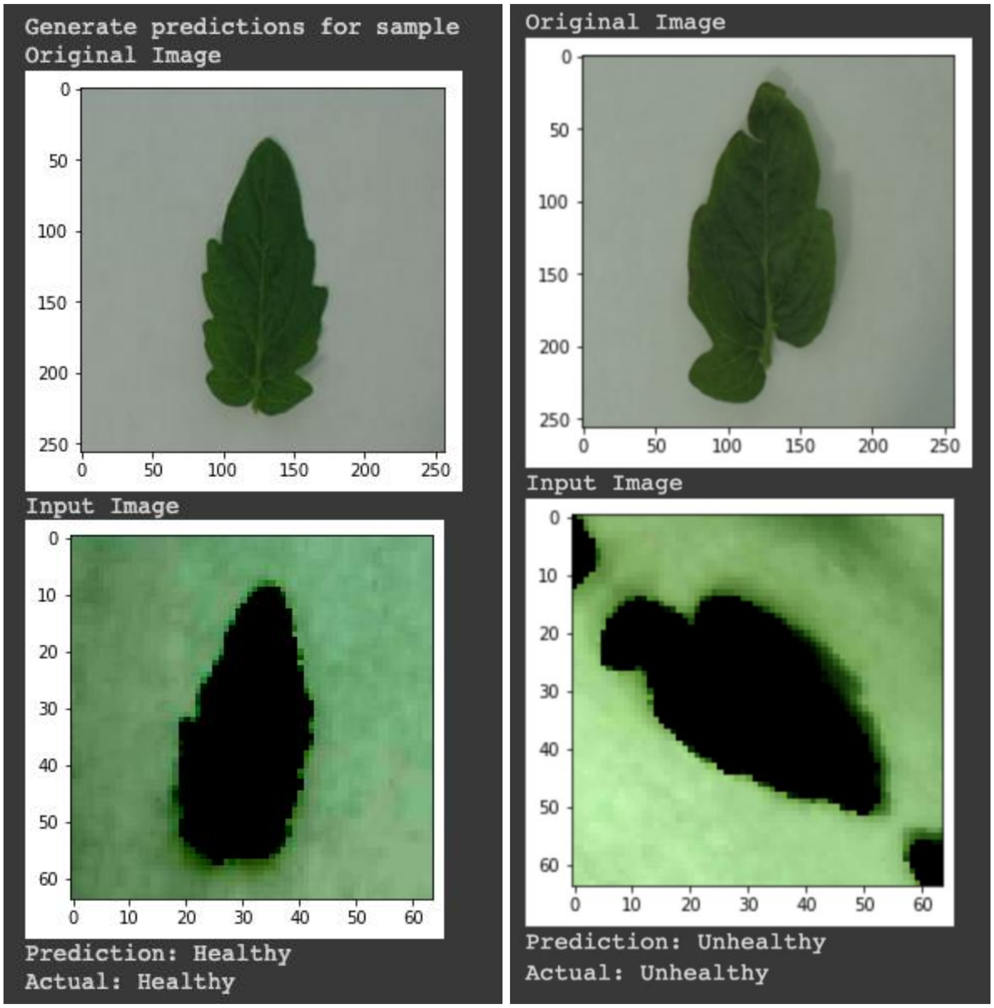


Figure 16. Health predictions using an image captured with our IoT system

Chapter 5 – Professional Issues and Constraints

5.1 Ethics - Science, Technology, and Society

As much as we would like to mitigate ethical risks to do with our project, there are still a couple of ethical concerns that we cannot eliminate. Just like any platform or product to do with collecting data, our project has a component of data ownership and privacy concerns. Under external threats, our product could leak plant and farm information a farmer may not want to disclose. There is also a concern of who owns the intellectual property to the data collected from a farm. The potential ethical problems to do with data ownership and privacy must be addressed, understood, and taken precautions before a project such as ours enters the market.

In addition, inaccurate recommendations provided by our deep learning model may lead to loss harvest or earnings for a farm. The likelihood of the risk of misidentification and system damage is low and our mitigation strategy for the former is to conduct frequent testing of the ML algorithms, and regularly update databases with a new disease, if detected. To make changes/updates to the model easier, without being physically near the set-up, we would be creating an independent environment on our microprocessor board which we can control remotely. A general mechanism we would like to incorporate into our model is a double-checking feature that would involve re-running the algorithm, that is, the checking is done twice totally, and results are compared to detect discrepancies.

Our research's goal is to gain knowledge that could help with reducing food waste, protecting our agriculture industry, and relieving farmers' stress. It is our hope that our project only be used in tools that supplement and better the lives of farm workers, instead of tools that overtake human jobs.

5.2 Civic Engagement

There are many ways in which our product may be implemented in our community. Our system is meant to be adaptable to settings ranging from large mono-crop farms to indoor vertical farms to home gardens. The hardware and software setup in each situation would need minimal change. So long as we can acquire a quality dataset from the intended setting, we can deploy our project.

5.3 Economic

The cost of carrying out our research was relatively low, at under $400. The hardware equipment we had to purchase are listed below in Appendix B. We also purchased a 3-months subscription of Google Colab to train and develop our deep learning mode.

5.4 Health and Safety

Our project has no serious health and safety concerns since most of the research was done in software on our laptops. The hardware components of our project run on low current and voltage levels.

5.5 Manufacturability

We wanted to focus on the feasibility of our ideas instead of designing our own hardware so everything we have used in our project can be bought off the shelf.

5.6 Usability

Since the setup of our project is very simple, and we are keeping the software portion of our research on our GitHub, we expect our project to be easily accessible to anyone who wishes to further the research.

5.7 Sustainability

Deep learning is finding its way into more and more applications, and agriculture is one of the ways in which it can make the greatest humanitarian impact. IoT technology is also becoming more mature with lower power consumption, more powerful processors, and more robust networking capabilities. We are sure that there will be further research and development in deep learning IoT agricultural technologies, and we hope that our research's novel findings will help guide future work in this area.

5.8 Environmental Impact

The environmental impact of our research was minimal as we did not consume much electricity and we did not waste any material. However, if our product was to be deployed in a real-world farm, we would see a huge potential for a positive environmental impact in that our system would detect problems caused by over/under watering, improper sun exposure, and imbalances in the pH level of the soil. By detecting and solving these issues early on, our product would help to create a more effective and environmentally friendly system to care for crops.

Chapter 6 – Conclusion and Future Work

6.1 Conclusion

In furthering the research on deployable systems using deep learning and IoT technology, we have implemented a NVIDIA Jetson Nano based system that can remotely monitor the health of agricultural crops. We have conducted our research using tomato plants as proof of concept of a technology that can be deployed in any agricultural farm to help with the humanitarian cause of securing food production and reducing global hunger.

6.2 Further Improvements to Our Existing Model

One way we are aiming to improve our Deep Learning model is by training it with a more diverse dataset, that is, including images taken directly from the plant. Our current dataset is populated mostly with images of singular leaves on a plain background. To acquire a more realistic use case environment, populating our images with leaves on the plant itself without removing the leaf would be ideal, since this would allow us to fully automate the process of periodically checking the plant for diseases. Through our research into other existing work in this project field, we have found that the ResNet34 model displays the highest accuracy rates of about 99.67% [14], so we would aim to implement this network in the same way we did for AlexNet and compare our results. We can also have another camera and model running that would calculate, measure, and compare the overall growth rate of the tomato plant to display overarching trends that would aid the farmer in making productive conclusions and decisions of the impact certain weather conditions and trends have on their crops.

6.3 Addition of Motion Sensor to System

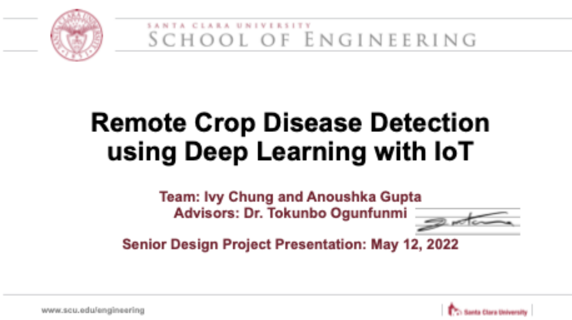
Adding a model working with moisture sensing data would also be an intuitive additional indicator for the farmer. By collaborating with a system that delivers equal amounts of water to all crops in a plot of farmland, helps further automate the process of crop maintenance and monitoring. A company that is working on this endeavor of harnessing moisture sensor data and technology is Verdi [20], which is based out of Vancouver, Canada. Their innovation consists of swarms of tiny smart valves that deliver water to crops that display low moisture statistics and reduce water delivery to those crops that exhibit high moisture levels. This system is currently being tested on grape plants, but a prospective application for it could be working in conjunction with our disease detected system, furthering the ability of farmers to regain control over their crop growth, in the face of a diverse type of soil, climatic/weather conditions and terrain.

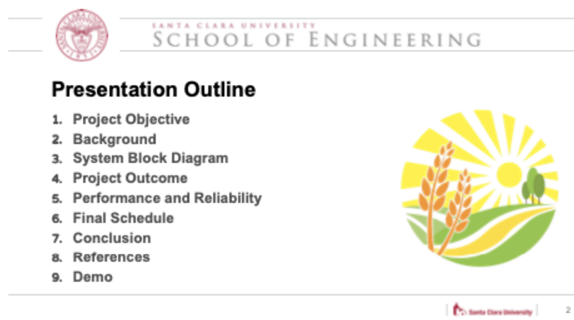
Chapter 7 – References

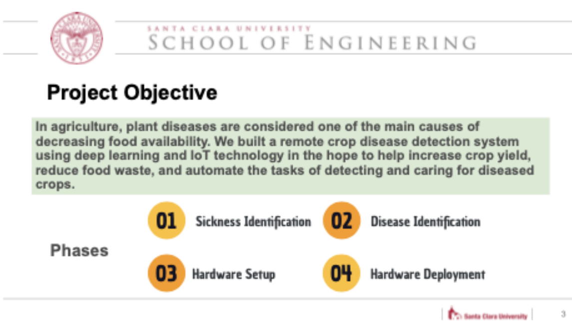
1. T. Gondwe, N. Nhamo, and D. Chikoye, “Chapter 1 - Smart Agriculture: Scope, Relevance, and Important Milestones,” in Smart Technologies for Sustainable Smallholder Agriculture: Upscaling in Developing Countries, Academic Press, 2018.
2. B. Erickson and J. Lowenberg-DeBoer, Purdue University, West Lafayette, IN, 2020.
3. “Rapid reforestation,” DroneSeed. [Online]. Available: https://droneseed.com/. (accessed: Apr. 15, 2022).
4. V. Panchbhaiyye and T. Ogunfunmi, “Experimental Results on Using Deep Learning to Identify Agricultural Pests,” 2018 IEEE Global Humanitarian Technology Conference (GHTC), 2018.
5. World Bank Group, “Food security update,” World Bank, 08-Apr-2022. [Online]. Available: https://www.worldbank.org/en/topic/agriculture/brief/food-security-update. (accessed: Apr. 15, 2022).
6. “Www.fightfoodcrises.net,” fightfoodcrises.net. [Online]. Available: http://www.fightfoodcrises.net/. (accessed: Apr. 15, 2022).
7. S. Omondi, “The most popular vegetables in the world,” WorldAtlas, 19-Jun-2018. [Online]. Available: https://www.worldatlas.com/articles/the-most-popular-vegetables-in-the-world.html. (accessed: Apr. 15, 2022).
8. Foolad MR, Merk HL, Ashrafi H. Genetics, genomics and breeding of late blight and early blight resistance in tomato. Crit Rev Plant Sci. 2008;27:75–107. doi: 10.1080/07352680802147353.
9. Nowicki M, Foolad MR, Nowakowska M, Kozik EU. Potato and tomato late blight caused by Phytophthora infestans: an overview of pathology and resistance breeding. Plant Dis. 2012;96:4–17. doi: 10.1094/PDIS-05-11-0458.
10. Reddy SA, Bagyaraj DJ, Kale RD. Management of tomato bacterial spot caused by Xanthomonas campestris using vermin compost. J Biopest. 2012;5(1):10–13.
11. V. K. Singh, A. K. Singh, and A. Kumar, “Disease management of Tomato through PGPB: Current trends and future perspective,” 3 Biotech, Aug-2017.[Online].Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5519495/.

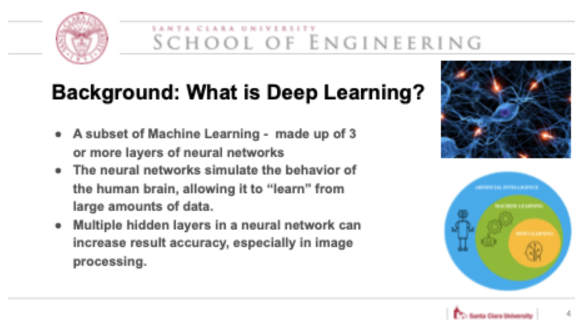
12. “Machine Learning: What it is and why it matters,” SAS, 2021. [Online]. Available: https://www.sas.com/en_us/insights/analytics/machine-learning.html. (accessed: Apr. 17, 2022).
13. T. M. Mitchell, Machine learning. New York: McGraw Hill, 2017
14. M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, “Computer Vision, IoT and Data Fusion for Crop Disease Detection Using Machine Learning: A Survey and Ongoing Research,” Remote Sensing, vol. 13, no. 13, p. 2486, Jun. 2021, doi: 10.3390/rs13132486.
15. J. Schmidhuber, “Deep Learning,” Scholarpedia, 2015. [Online]. Available: http://www.scholarpedia.org/article/Deep_Learning.
16. “Transfer learning using AlexNet,” MATLAB & Simulink. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/transfer-learning-using-alexnet.html. (accessed: Apr. 19, 2022).
17. Scheer, Claudio & Guder, Larissa & Griebler, Dalvan. (2019). Deep Learning in Agriculture: A Systematic Literature Review. 10.13140/RG.2.2.17367.19363.
18. Strisciuglio, Nicola & Lopez Antequera, Manuel & Petkov, Nicolai. (2020). Enhanced robustness of convolutional networks with a push–pull inhibition layer. Neural Computing and Applications. 32. 1-15. 10.1007/s00521-020-04751-8.
19. “Jetson Nano,” NVIDIA Developer. https://developer.nvidia.com/embedded/jetson-nano (accessed: Apr. 15, 2022).
20. “Climate-proofing agriculture,” Verdi. [Online]. Available: https://verdiag.com/. (accessed: Apr. 20, 2022).

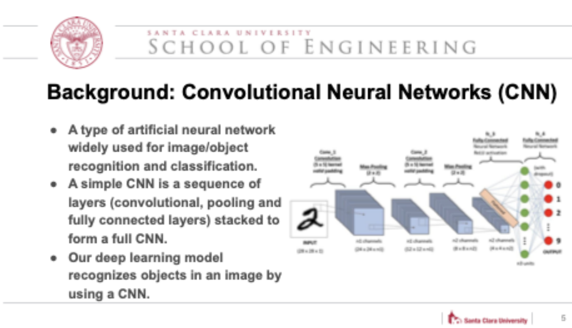
Appendix A: Conference Slides

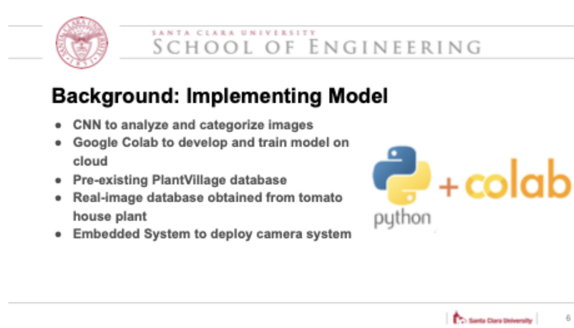












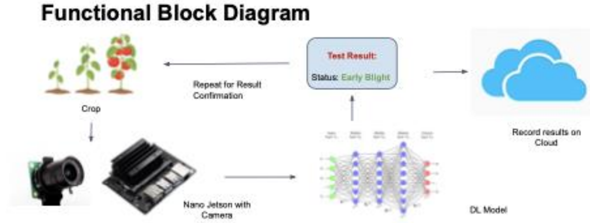
Background: Image Indicators

Leaves:
- Wilting (dehydration)
- Discoloring (yellow edges, purple veins)
- Curling under/over or holes (pests)

Fruits:
- Holes (from pests)
- Quantity of fruits/flowers (soil)

Whole plant:
- Growth Rate (nutrient deficiency)

Functional Block Diagram



Project Outcome: Data Collection

- Dataset: PlantVillage
  - 4 different Tomato species
  - 10 subcategories with 9 disease classes which are are some of the most common Tomato plant diseases.

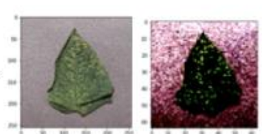
Project Outcome: Training and Validating Data

- 75% of dataset used for training 25% of dataset used for validation
- For each class we have between 1702 to 1961 images to train, and between 425 to 490 images to test

	Healthy	Spider Mites	Septoria Leaf Spot	Leaf Mold	Late Blight	Bacterial Spot	Mosaic Virus	Yellow Leaf Curl Virus	Early Blight	Target Spot
Train	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1702	[illegible]	1961	[illegible]	[illegible]
Test	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	425	[illegible]	490	[illegible]	[illegible]

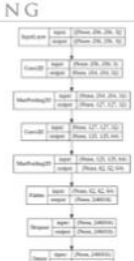
Project Outcome: Data Pre-Processing

- Resized to 64 by 64 pixels
- Pixel-wise standardized
  - using Tensorflow *per_image_standardization* function



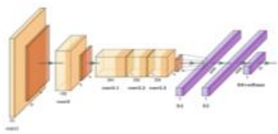
Project Outcome: Our Own CNN Model

- 3 total layers
  - 2 convolution and max pooling layers
  - 1 fully connected and dropout layer
- 98.5% - 2 class classification test accuracy
- 53.6% - 10 class classification
  - K-fold cross validation improved result to 74.4%



Project Outcome: Training the AlexNet

- Transfer Learning with AlexNet
- 10 class classification model - 89.8% test accuracy



SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

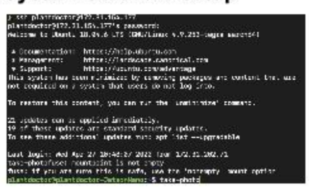
Project Outcome: IoT System Hardware Setup

1. Nvidia Jetson Nano 4GB
2. Intel Dual Band Wireless-Ac 8265 card
3. 2 RP style antennas
4. Raspberry Pi HQ camera
5. JOBY GorillaPod

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

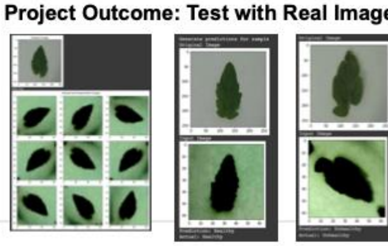
Project Outcome: IoT System Software setup

1. Wifi Connectivity
2. Secure Shell
3. Cloud Connection using *google-drive-ocamlfuse*
4. Global command of *take-photo* which uses the *nvgstcapture* library



SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Project Outcome: Test with Real Images

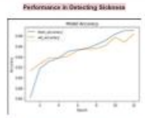


SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

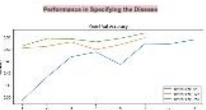
Performance and Reliability: Our Own CNN Model

- In detecting whether or not the plant is sick, our CNN model was able to achieve a validation accuracy of 98.50%
- For 10 class classification, our CNN model was able to achieve a validation accuracy of 74.4%

Performance in Detecting Sickness



Performance in Specifying the Disease

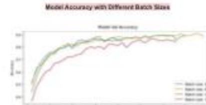


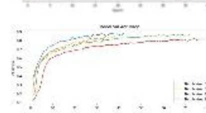
SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Performance and Reliability: AlexNet Model

- By using the optimized epoch count and batch size, we were able to attain higher accuracy in our test results
- For 10 class classification, our AlexNet model was able to achieve a validation accuracy of 89.8%

Model Accuracy with Different Batch Sizes





Final Schedule

Tasks										
Research	X	X	X	X	X					
Development environment			X	X	X					
Building database						X	X			
Building and training DL model							X	X		
Testing DL model								X		
Hardware									X	X
System Testing									X	X
	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	March	April

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Conclusion

- **It is extremely difficult and labor intensive for farmers to detect and identify plant illnesses on time for treatment, resulting in unnecessary food waste that we hope to eliminate.**
- **To combat this issue, we have utilized deep learning in conjunction with IoT to automate the monitoring of plant health remotely.**

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

References

Scheer, Claudio & Guder, Larissa & Griebler, Dalvan. (2019). Deep Learning in Agriculture: A Systematic Literature Review. 10.13140/RG.2.2.17367.19363.

V. Panchbhaiyye and T. Ogunfunmi, "Experimental Results on Using Deep Learning to Identify Agricultural Pests," 2018 IEEE Global Humanitarian Technology Conference (GHTC), 2018, pp. 1-2, doi: 10.1109/GHTC.2018.8601896.

Mark, Ryan, “Ethics of Using AI and Big Data in Agriculture: The Case of a Large Agriculture Multinational”, The ORBIT Journal, Volume 2, Issue 2, 2019, Pages 1-27, ISSN 2515-8562, https://doi.org/10.29297/orbit.v2i2.109.

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Demo 1

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Demo 2

Appendix B: Hardware and Software

- NVIDIA Jetson Nano 4GB
- Intel Dual Band Wireless-Ac 8265 card
- 2 RP style 6dbi gain antennas
- Raspberry Pi HQ camera
- JOBY GorillaPod
- Tomato Houseplant
- Macbook Pro
- Google Colab
- Google-drive-ocamlfuse
- Nvgstcapture library
- Python with relevant libraries

Appendix C: GitHub

Our GitHub repository with all our final models and a Wiki outlining some of our work can be found here: https://github.com/ivychung/plant-ML