Ben-Gurion University of the Negev
The Faculty Engineering
The Department of **Industrial Engineering and Management**

AI Algorithms and Machine Learning for Plant Disease Detection

Eyal Ginosar

Complete assignment report.
Literature Review and Project

Contents

List of Figures

List of Tables

1 LITERATURE REVIEW

1.1 Introduction

Diseases and pests in plants cause a slowdown in the growth of crops. The diseases affect the quantity and quality of agricultural products, thus affecting the agricultural industry ecologically and economically. Farmers are faced with the difficult task of reducing and eradicating pests in plants as well as identifying disease outbreaks. In most countries today, pests and diseases are located manually by specialists, which entails the need for regular crop management [20]. The diseased part of a leaf can cause an entire growing plot of the crop to become infected, so it is essential to detect these diseases as soon as they appear. Aside from posing a threat to food security globally, plant diseases can be devastating to smallholder farmers whose livelihoods rely on healthy crops. Therefore, identifying those diseased leaves in a timely and optimal manner is crucial to prevent greater losses to crops.

Plant disease detection can be automated using computer vision technology, thereby reducing the amount of time and effort required to detect diseases. These systems can replace the need for expert advice and guidance [18]. Therefore, an automated disease detection system is essential. There are already companies and applications that offer smart solutions using machine learning to support farmers in this regard [13]. A variety of machine learning methods have been used to classify and identify plant diseases over the years. In addition to more traditional machine learning algorithms like KNN, SVM, and RF, artificial intelligence has become more popular in recent years [14]. In particular, CNNs (Convolutional Neural Networks) use deep learning methods to detect diseases more effectively [5]. One of the benefits of deep learning is the ability to extract features automatically from images as the network learns how to extract them in training [20]. This eliminates the need for manual selection of the desired features which makes the process easier and possibly more accurate.

1.2 Related Work

The detection of plant diseases is usually carried out through the analysis of images of the leaves. The leaves are analyzed because they are considered the most likely area where symptoms of most plant diseases will appear [2]. The following five steps [5] (see Figure 1) are involved in the processing process:

- **Data collection:** The most common method of collecting data is through high-quality images taken by humans, robots, and drones.
- **Database Annotation:** This is usually done by an expert whose job is to categorize healthy plants and categorize unhealthy plants according to different diseases.
- **Image Processing:** The image is first pre-processed to identify the most important features for further processing. An image segmentation process is used to

parse the plant images into different parts. In image segmentation, each meaningful or focused part of the digital image is partitioned into various parts required within the domain of interest. It is used to differentiate and isolate the foreground from the background of an image. It is useful for identifying diseased areas on a plant.

- **Feature extraction:** Extracting features of color, shape, and texture of diseased parts of a plant. Features are quantifiable properties obtained from the various targeted parts of an image. It is an important step in computer vision-based machine learning tasks.
- **Classification:** Machine learning algorithms are employed to classify plant diseases.

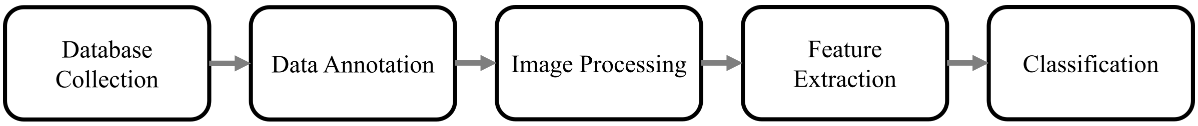

Figure 1: Disease detection work flow.

There are many ways by which the last three steps of the plant disease detection process can be carried out. They can generally be divided into two categories: traditional machine learning and deep learning [14]. Next, I will review the uses of these categories for the rest of the survey.

1.2.1 Traditional machine learning techniques

KNN Classifier

K-Nearest Neighbors (KNN) is a method used for finding patterns, statistical estimation, and classification in machine learning. This method is considered simple, fast, and its results are easy to understand [21].

Kaushal and Bala [12] used this method in their work as an alternative to the SVM algorithm for the classification step and were able to improve the disease classification results by about 10%. Singh and Kaur [21] also used this algorithm for identifying and classifying potato owner diseases that were taken from the "Plant Village" image database. Their pre-processing involved converting the image to a grayscale image and then using the Gray-Level Co-occurrence Matrix (GLCM) method for feature extraction of the texture. Within an image, GLCM describes the distribution of pixel values (grayscale or color values) at given offsets. The method is used for analyzing textures in a variety of applications. By analyzing the intensity or values of the grayscale or different color dimensions of an image, the common occurrence matrix can measure the texture of the image. Following feature extraction, the leaves were segmented using the

K-means algorithm to identify infected areas. Lastly, the image generated by the K-means algorithm is entered as an input into the KNN algorithm when k = 1. The model was evaluated on a small set of data (24 pictures for training, 16 pictures for testing) and achieved an average accuracy, precision, recall, and F1-score of 97% across all classes.

SVM Classifier

Support Vector Machine (SVM) is fundamentally a binary classifier that constructs a linear separating hyper-plane to classify data instances. SVM can also be enhanced by using the "kernel trick" to transform the original feature space into a higher-dimensional feature space. Clustering, regression, and classification have all been done using SVM [14]. This algorithm has several advantages over other algorithms, including efficiency when dealing with high dimensional spaces, effectiveness even in cases where the number of dimensions is greater than the number of samples, and memory efficiency [9].

Islam et al. [11] classified potato diseases using SVM based on images of the potato leaves. A process of segmentation was first performed on the leaf to separate it from the background of the image, and then a second segmentation was performed only on the affected areas (if there were any). A set of masks was generated by analyzing the color and luminosity components of different areas in the image using the L*a*b color space. Following that, they extracted 10 features based on color and texture. An analysis of the texture used GLCM to extract contrast, correlation, energy, and homogeneity features. In addition, they used numerical indicators such as mean, standard deviation, entropy, skew, and energy. Lastly, a linear SVM model was trained on 300 images (180 for training, 120 for examination), two-thirds of which were of diseased potato leaves. On average, the model's precision, recall, and F1-score were 95% across all classes. Hossain et al. [9] classified tea plant disease using the SVM algorithm. Their image processing process involved uniformizing each image's size and converting it to gray-scale. Feature extraction was then performed on 10 features they manually selected (Contrast, Correlation, Energy, Mean, Standard deviation, Entropy, RMS, Variance, Kurtosis, Skewness). For the disease classification, they trained a model on a set of 200 images (150 for training, 50 for examination) and achieved a 93% accuracy rate across all classes.

Random Forest Classifier

Random Forest (RF) is a machine learning algorithm used for several types of classification and regression tasks. It is an ensemble of tree-structured classifiers. Every tree of the forest gives a vote, assigning each input to the most probable class label. This method is fast, robust to noise, and capable of identifying non-linear patterns in the data. Additionally, it can work with numerical as well as categorical data. Random Forest has the advantage of not suffering from over-fitting even if more trees are added to the forest [3].

Ganatra and Patel [7] applied this algorithm to classify a large number of plant diseases. Their database had 14956 images which were divided into 38 different classes. The image processing included denoising the image and focusing on point of interest. After that, they used Otsu's thresholding technique [19] for segmentation. In the next step, 31 features were extracted for classification. These features can be divided into six categories: (1) Color Moments Features – using both RGB and HSV outputs. (2) Texture Features – using GLCM (3) Shape Features. (4) Vein Features - Image vein structure is obtained and white pixel from the area of disease affected leaf is calculated. (5) Zernik Moment Features - global properties that are obtained from the entire image. (6) Gabor Feature – obtain specific frequency content in the appropriate direction in an image. Using performance metrics such as accuracy, precision, recall, and F1-score, they found the random forest model to achieve the best results among the KNN, SVM, RF, and ANN classification models. In comparison to other works in this survey, the accuracy of the classification results of this work is lower, at 73.38%. The research in this paper, however, classified 38 different classes, contrary to other studies which categorized only a limited number of classes (3 or 4). This resulted in a significant reduction in model accuracy. Another example where a classification model using random forest has obtained superior results over other classic machine learning models is found in the work of Chauhan et al. [4]. The purpose of this study was to identify diseases of corn plants. The database contained 3823 images divided into 4 categories. During the image processing stage, gray-scale conversion and segmentation were performed on the image. The feature extraction stage included extracting features of color, texture, and shape. Finally, several models were trained when in the end the random forest model had the highest accuracy, precision, and recall metrics of them all.

1.2.2 Deep Learning

The use of deep learning technology has been successful in classifying images. It also has been studied into improving the quality of crop management and supporting better agricultural practices [8]. As part of deep learning, features are automatically extracted by the model during training instead of having to be separately extracted. Image classification problems have been widely addressed using CNN models in recent years. The multi-layered structure allows them to identify and classify objects easily with minimal pre-processing, analyze visual images successfully, and separate the required features easily [20]. It has been shown that deep learning algorithms outperform classical machine learning approaches and produce better results [17].

Convolutional Neural Network

The convolutional neural network (CNN) is made up of four layers: the convolutional layer, the pooling layer, the activation function layer, and the fully connected layer. Figure 2[1] shows a general CNN architecture.

Mohanty et al. [15]'s work is one of the most cited in the literature on detecting plant diseases with deep learning. They analyzed a very large data set of 54,306 images with 38 different classes and conducted a comprehensive study based on two well-known networks: AlexNet and GoogLeNet. They examined 60 different configurations for the model that included the two networks, two training approaches (transfer learning and training from scratch), 3 variations of the data (color, gray-scale, and segmented leaf),

[1]https://www.analyticssteps.com/blogs/convolutional-neural-network-cnn-graphical-visualization-code-explanation

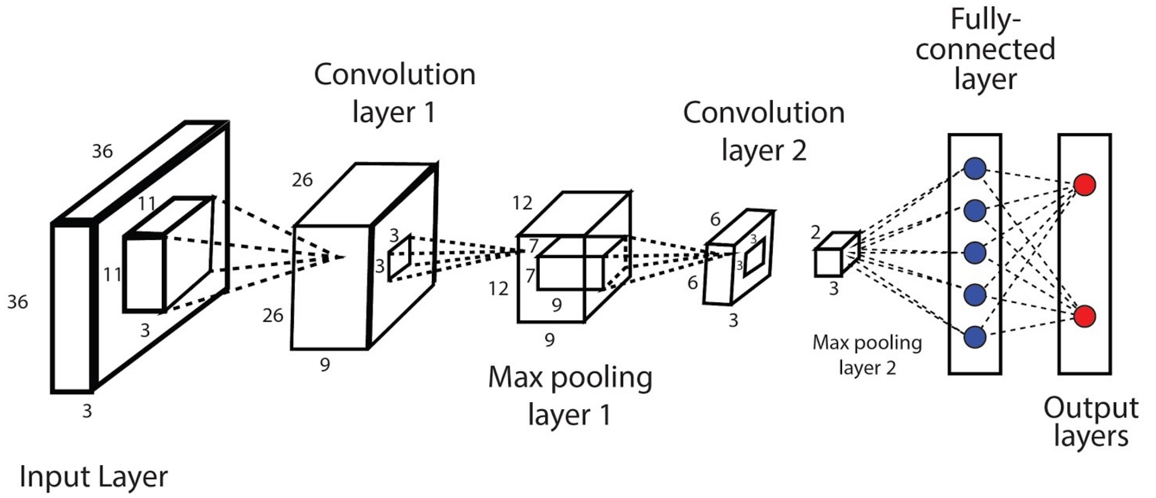

Figure 2: An example of CNN architecture.

and 5 different divisions of the data for train and test. The best result was obtained for a model using the GoogLeNet network, which is trained in the form of transfer learning, with color image input, and with 80% of the data used for learning and the rest for testing. This model has reached 99.34% accuracy. However, testing this model on two other (small) data sets, yielded results of only about 31.5%. Although this algorithm requires a lot of training (several hours on a powerful computer) the classification is remarkably fast (less than a second) and therefore suitable to fit in a mobile device or robot. It is important to note that all of the photographs are of experimental (lab) setups, not actual conditions in a growing field. Ferentinos [6] developed a disease detection model on data that contains both images under laboratory conditions and real images from the field. They also worked with a very large image database (87,848 images). They compared five CNN architectures: AlextNet, AlextNetOWTBn, GoogLeNet, Overfear, and VGG. The best model was obtained for VGG architecture, with 99.53% accuracy. Their model requires low computional power for the classification which make it feasible as well for an integration with a mobile device such as drone.

Overall, deep learning algorithms have surpassed traditional machine learning algorithms in performance. In addition to their accuracy in classification, they do not require feature engineering, thus they are a far better approach to detecting plant diseases. Even though they demand a great deal of training data, transfer learning supports this process while also improving accuracy [15]. Additionally, the model actually performs very quickly, which means the applications can be optimized for mobile devices even though the training process takes a long time.

2 PROJECT TOPIC

Diseases in plants are a major problem in agriculture that damages the crops and thus harms the farmers. Technology nowadays enables automatic plant disease detection systems through computer vision, thereby reducing the amount of time and effort spent on detecting diseases. For large-scale agricultural operations, the system could be combined with autonomous vehicles to accurately locate infectious disease problems quickly throughout the crop area. This is valid as long as the system is able to detect and diagnose specific diseases accurately and quickly under real-life conditions (i.e., in the field) and can be operated via an effective mobile interface that is easy to use [25].

This project topic is **First steps towards plant disease detection companion drone**. This project topic is the First step towards plant disease detection by a drone. In this project, I have created a plant disease detection model and a prototype for an autonomous drone that will be able to signal the presence of diseases on plants in the fields. In this way, an autonomous drone would be able to go through crop fields, get an instant indication for the presence of diseases in the plants and give a signal for the type of disease that it has found. This kind of drone could work alongside farmers who seek to locate and treat unhealthy plants in their fields.

3 OBJECTIVES

Incorrectly diagnosing agricultural diseases can lead to inappropriate use of chemicals and the emergence of resistant pathogen strains, higher input costs, and more outbreaks with considerable economic and environmental losses. Currently, disease diagnosis involves human scouting, which takes time and is expensive. Computer-vision-based models offer the potential to increase efficiency, but the great variation in symptoms due to aging tissues, genetic variation, and lighting conditions in trees makes detection less accurate. Thus, this project's objectives are to train a model with images of plant leaves to:

1. Accurately determine if a leaf is healthy or not and accurately classify its disease given it is a diseased leaf.
2. Recognize and distinguish between diseases, sometimes multiple ones on a leaf.
3. Deal with novel classes and symptoms.
4. Create a prototype of a disease detection drone.

4 DATA

The data for this project is taken from 'The Plant Pathology Challenge 2020 data set to classify foliar disease of apples' [24]. It is an expert-annotated data set of 3642 high-quality, real-life symptom images of multiple apple foliar diseases, with variable illumination, angles, surfaces, and noise. The data is labeled to four classes: 'healthy', 'multiple_diseases', 'rust', and 'scab'. It was made available to the Kaggle community for the Plant Pathology Challenge competition[2].

The data set was divided to 1821 labeled images for training and 1821 unlabeled images for testings. For this project, I used the 1821 labeled images as may data set. First, I divided the data to train and test set by 80-20, resulting in 1456 train images and 365 test images. Then, I have divided the train set again by 80-20 in order to have a train and validation sets. This resulted is a train set of 1164 images and validation set of 292 images, that were used to evaluate the model for the model tuning. All the data splits were stratified to make sure the distribution of the classes will stay the same in all the data sets.

Figure 3: Example of images from the data set. ***Left:*** healthy leaf. ***Right:*** leaf with crust disease.

4.1 Data preparation

Every image has been resized to to 224x224. Further more, a normalization of the data has been made by dividing all pixels values by 255. This is done for the data to be compatible with the networks initial values.

4.2 Data augmentation

I used data augmentation on the training data in order to train the model on a larger dataset. The images of the original train dataset has been randomly allocated different types of augmentations, including: horizontal flip, vertical flip, up to 90 degrees rotation, width shift, height shift, zoom, and brightness change. The augmentation was made by the ImageDataGenerator class from keras API[3].

[2] https://www.kaggle.com/c/plant-pathology-2020-fgvc7/overview

[3] https://keras.io/api/preprocessing/image/

5 METHODS

5.1 Deep learning image classifiers

For the past few years, several deep learning image classifiers architectures has become very popular. Too et al. [25] done a comparison between some of them, including VGG net model, ResNet, Inception V4, and DenseNet. According to the results of their experiment, DenseNets tend to yield a coherent increment inaccuracy that grows as the number of epochs increases, without signs of performance degradation. Furthermore, DenseNets preforms well in classification with a relatively smaller number of parameters and reasonable computational time. Thus, I decided to build my classifier based on DenseNet121 architecture.

5.1.1 DenseNet

A densely connected CNN architecture was introduced in Huang et al. [10]'s paper. The feed-forward connection between layers in the network ensures maximum information flow between them. Every layer receives inputs from all preceding layers and the feature maps are then used to produce all subsequent layers. With DenseNets, the vanishing gradient problem is minimized, and the number of parameters is reduced substantially [25]. In this project, I have used a DenseNets model with 121 layers.

Input weights were taken from ImageNet and loaded into the model. This procedure is known as transfer learning, which uses the weights of a network that was trained on a different data set. It has been previously shown the using transfer learning from ImageNet increases the results of plant disease classification [15].

Finally, I have added a dense layer with four neurons at the end of the network for the classification.

5.2 Model Tuning

I have done a hyper-parameters tuning for several parameters in the model. In this section, I will elaborate on each one of them.

5.2.1 Activation function

In order to fit the original DenseNet model to my classification problem, I added a fully connected dense layer with four neurons as the final output layer. This layer has an activation function, which is a hyper-parameter I have decided to fine-tune. I have done a comparison between six activation functions: 'sigmoid', 'relu', 'softmax', and 'elu'. The selected activation was 'softmax' with accuracy of 90%.

5.2.2 Number of trained layers

As the weights of the network are loaded from a trained network, it is possible to choose how many layers to re-train on the data set of this classification problem. The network is made out of 429 layers. The best accuracy result was given by training all of the network's layers.

5.2.3 Number of epochs

An epoch is a training iteration where at the end of it the network's weights are being updated. A low number of epochs can result in an under-fitted model, and a too-large number could result in an over-fitted model. Thus, by relying on Too et al. [25] work, I have compared between 10 and 30 epochs. As expected, 30 epochs achieved higher accuracy. A very small change (.<1%) for the favor of 30 epochs would have made me prefer to use 10 epochs because it runs much faster. But, since the difference in accuracy is 1.4% (91.4%), I decided to use 30 epochs for the final model.

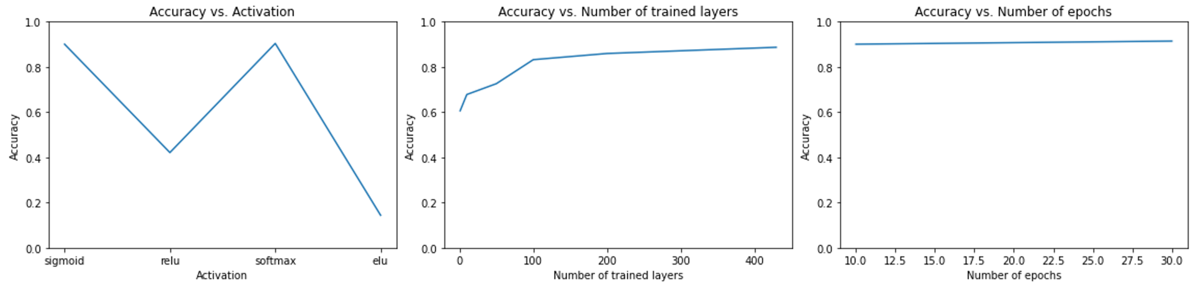

Figure 4: ***Left:*** Activation function tuning. ***Center:*** Number of trained layers. ***Right:*** Number of epochs.

6 RESULTS

6.1 Training

Model evaluation is based on categorical accuracy metric and categorical cross-entropy loss (loss). Model performance is measured through the evaluation of overall loss and accuracy based on the test dataset. The results are presented in Table 2. The experiment runs for a total of 30 epochs. Where epoch refers to the number of training cycles. Stochastic Gradient Descent (SGD) is used to train the network model. The hyper-parameters that were selected from the tuning process are summarized in Table 1. The batch size, learning rate, and weight decay parameters did not get tuned, as their values were obtained from previous work [25]. The batch size was set to 16, the learning rate was set to 0.001, and the weight decay to $1e^{-6}$.

Table 1: Summary of the model's parameters

Batch size	**Solver**	**Learning rate**	**Weight decay**	**Activation**	**Epochs**	**Training layers**
16	sgd	0.001	1e-6	softmax	10	400

Fig. 5 shows the accuracy and loss on the train and validation data sets. We can see that the model it fitted well between train and validation.

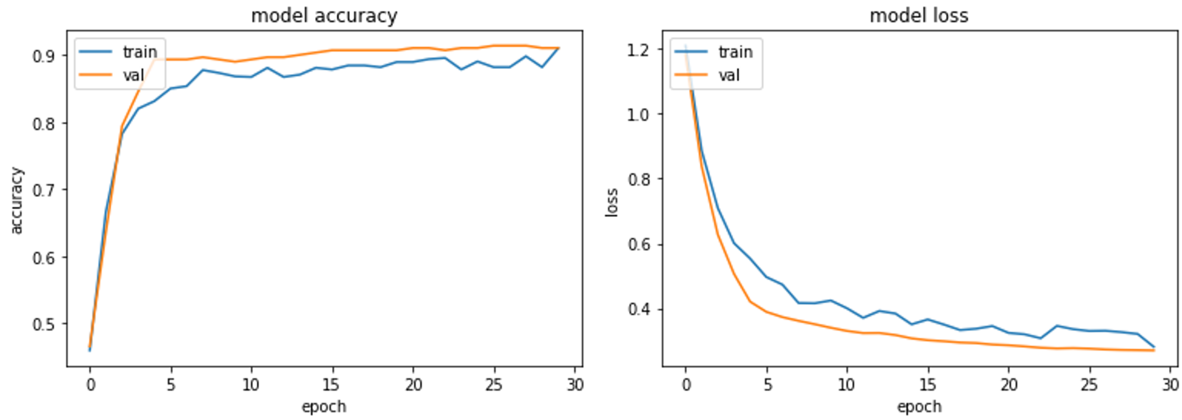

Figure 5: The accuracy and loss plots of training and validation.

6.2 Results of the experiment

The results of the experiment are summarized in Tables 2 and 3. The overall accuracy of the model on the test set was 92.6% with a loss of 0.231. Healthy leaves and leaves with rust disease were classified in high accuracy (recall of 0.99 and 1.0 respectively), while multiple diseases were classified poorly (recall=0.06) and the classification was dispersed between all other options quite evenly (see Fig. 6). This is not very surprising, as this class had the smallest amount of train images, in contrast to the other classes which had a similar distribution in the data. These images also contain multiple diseases which makes it harder for the model to distinguish between the specific disease to the multiple diseases.

Table 2: Accuracy and loss of training and test

Training accuracy%	**Training loss**	**Test accuracy%**	**Test loss**
91.1	0.281	92.6	0.231

Table 3: Classification Report

	precision	recall	f1-score	support
healthy	0.84	0.99	0.91	103
multiple_diseases	1.00	0.06	0.11	18
rust	0.95	1.00	0.98	125
scab	0.96	0.90	0.93	119

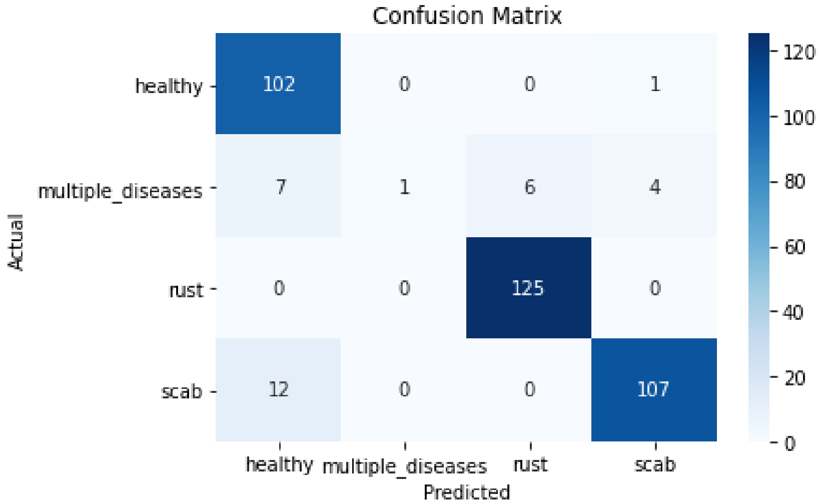

Figure 6: Confusion matrix on test set.

7 PROTOTYPE

I have built a LED system on a small commercial drone. The purpose of this system is to enable the drone to signal to the farmers that is has detected a diseased leaf, and to also give more information like the type of disease or the severity by using different color and intensity of illumination.

7.1 Design

I designed the system based on prior work of Szafir et al. [23] and Monajjemi et al. [16]. The interface is based on a 30 programmable LED strip (NEO pixels) that is mounted on the front of the drone (see Fig. 8). Each LED can be controlled individually (brightness and color) through a micro-controller such as Arduino.

7.2 Hardware

To make the prototype, I used the following hardware:

- **Drone:** The drone I used for this project was DJI Mavic Enterprise Dual[4].
- **Controller:** I used Arduino UNO R3 as a micro-controller to control the LEDs. The Arduino UNO is a common micro-controller that was used in many similar works [23, 1, 22]. The code was written in C++.
- **LEDs:** I used Adafruit NeoPixels as the programmable LEDs. A 30 LED strip with a density of 144 LEDs per meter is mounted on the drone. These LEDs can be individually programmed, which enables the control on the brightness and color of every single LED in the strip.

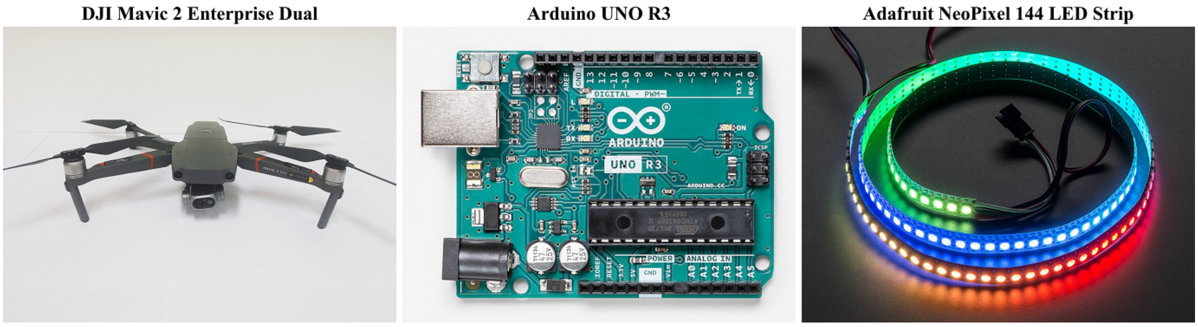

Figure 7: The hardware used for the creation of the prototype.

7.3 Setup

To implement the system on the drone, I designed a surface for the Arduino and the battery to sit on. I took measures, making sure the surface, its attachments, and the hardware will not touch the propellers of the drone and will not block any of it sensors. Then, I designed and 3D printed the surface and its attachments. I assembled the Arduino board, the battery, and LED connected everything together (Figure 8. The Arduino is powered by the battery through its DC (Direct Current) barrel jack input port. The LEDs are mounted on the front of the drone and powered by a 5V voltage that is derived from the battery by a step-down switching regulator.

By implementing a plant disease detection system on the drone's software, as the one presented in this work, the drone will be able to detect diseased plants and notify its surrounding. For instance, for this project's scope, the drone can show orange light if it detects rust disease, purple light for scab disease or red light for multiple diseases. This can be further extended to convey more information by using changing animations and brightness. Figure 9 shows an example. In the supplementary material to this work, you will find a short video illustration.

[4]https://www.dji.com/mavic-2-enterprise

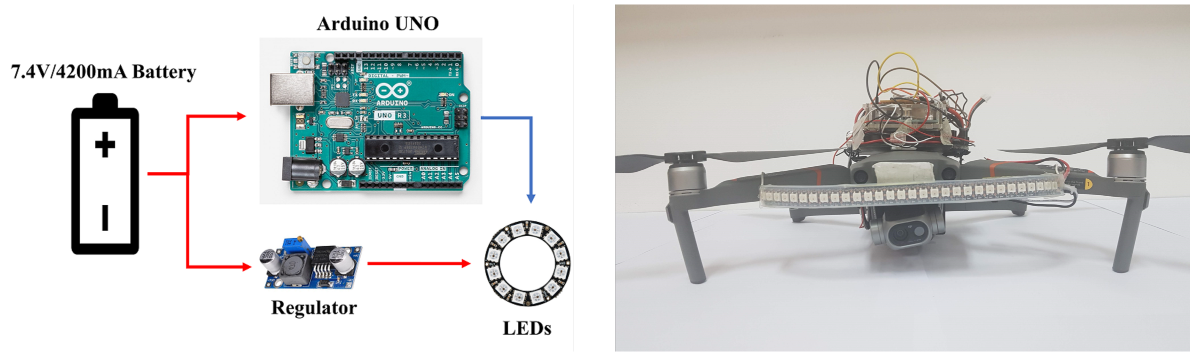

Figure 8: The prototype and system setup.

Figure 9: Concept images of the drone detecting a diseased plant and notify this with lights.

8 CONCLUSION

In this work, I implemented a DenseNet architecture for a plant disease classification task. Based on a 121 layers DenseNet netowrk, I fine-tuned the hyper parameters to maximize the model's accuracy on my data set. Then, I trained the model and evaluated it on a small test set.

The model had an overall accuracy of 92.6%, which I consider a decent result if compared to previous work in the field. One class in particular (multiple diseases) had extremely low recall, which effected the most on the error rate of the model. This class suffered from a small amount of data compared to the other, thus, I believe that an up-sampling process for this class could help in increasing the accuracy of it and by that the accuracy of the whole model.

After the modeling of the classifier, I designed and implemented a LED system on a small commercial drone, which in the future will be able to work in integration with a real-time disease detection that is based on the classifier.

This drone will then be able to accompany farmers and their fields, scanning and detecting diseases in the crops and provide the information of what if found to the farmers using LEDs in different colors and brightness.

References

[1] Kim Baraka, Ana Paiva, and Manuela Veloso. Expressive lights for revealing mobile service robot state. In *Robot 2015: Second Iberian Robotics Conference*, pages 107–119. Springer, 2016.

[2] Jayme Garcia Arnal Barbedo. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144:52–60, 2016. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2016.01.017. URL `https://www.sciencedirect.com/science/article/pii/S1537511015302476`.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Ms Deepika Chauhan et al. Detection of maize disease using random forest classification algorithm. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(9):715–720, 2021.

[5] T Daniya and S Vigneshwari. A review on machine learning techniques for rice plant disease detection in agricultural research. *system*, 28(13):49–62, 2019.

[6] Konstantinos P Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[7] Nilay Ganatra and Atul Patel. A multiclass plant leaf disease detection using image processing and machine learning techniques. *Int. J. Emerg. Technol*, 11(2):388421, 2020.

[8] Reem Ibrahim Hasan, Suhaila Mohd Yusuf, and Laith Alzubaidi. Review of the state of the art of deep learning for plant diseases: a broad analysis and discussion. *Plants*, 9(10):1302, 2020.

[9] Selim Hossain, Rokeya Mumtahana Mou, Mohammed Mahedi Hasan, Sajib Chakraborty, and M Abdur Razzak. Recognition and detection of tea leaf's diseases using support vector machine. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 150–154. IEEE, 2018.

[10] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[11] Monzurul Islam, Anh Dinh, Khan Wahid, and Pankaj Bhowmik. Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*, pages 1–4. IEEE, 2017.

[12] Gautam Kaushal and Rajni Bala. Glcm and knn based algorithm for plant disease detection. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 6(7):5845–5852, 2017.

[13] Jubin Dipakkumar Kothari. Plant disease identification using artificial intelligence: Machine learning approach. *Jubin Dipakkumar Kothari (2018). Plant Disease Identification using Artificial Intelligence: Machine Learning Approach. International Journal of Innovative Research in Computer and Communication Engineering*, 7 (11):11082–11085, 2018.

[14] Konstantinos G Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8):2674, 2018.

[15] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[16] M. Monajjemi, S. Mohaimenianpour, and R. Vaughan. Uav, come to me: End-to-end, multi-scale situated hri with an uninstrumented human and a distant uav. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4410–4417, 2016. doi: 10.1109/IROS.2016.7759649.

[17] Draško Radovanović and Slobodan Đukanovic. Image-based plant disease detection: a comparison of deep learning and classical machine learning algorithms. In *2020 24th International Conference on Information Technology (IT)*, pages 1–4. IEEE, 2020.

[18] Aakanksha Rastogi, Ritika Arora, and Shanu Sharma. Leaf disease detection and grading using computer vision technology & fuzzy logic. In *2015 2nd international conference on signal processing and integrated networks (SPIN)*, pages 500–505. IEEE, 2015.

[19] JA Saif, AA Al-Kubati, Abdultawab Saif Hazaa, and Mohammed Al-Moraish. Image segmentation using edge detection and thresholding. In *The 13 th International Arab Conference on Information Technology ACIT, Dec.(10-13)*, 2012.

[20] Melike Sardogan, Adem Tuncer, and Yunus Ozen. Plant leaf disease detection and classification based on cnn with lvq algorithm. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 382–385. IEEE, 2018.

[21] Jaskaran Singh and Harpreet Kaur. Plant disease detection based on region-based segmentation and knn classifier. In *International Conference on ISMAC in Computational Vision and Bio-Engineering*, pages 1667–1675. Springer, 2018.

[22] Sichao Song and Seiji Yamada. Designing led lights for a robot to communicate gaze. *Advanced Robotics*, 33(7-8):360–368, 2019. doi: 10.1080/01691864.2019.1600426. URL `https://doi.org/10.1080/01691864.2019.1600426`.

[23] Daniel Szafir, Bilge Mutlu, and Terry Fong. Communicating directionality in flying robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, page 19–26, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450328838. doi: 10.1145/2696454.2696475. URL `https://doi-org.ezproxy.bgu.ac.il/10.1145/2696454.2696475`.

[24] Ranjita Thapa, Kai Zhang, Noah Snavely, Serge Belongie, and Awais Khan. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Applications in Plant Sciences*, 8(9):e11390, 2020.

[25] Edna Chebet Too, Li Yujian, Sam Njuki, and Liu Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161:272–279, 2019.