ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Muhab HARIRI

DETECTION OF CALCIUM DEFICIENCY AND PHYSIOLOGICAL STATUS IN STRAWBERRY LEAVES USING DEEP LEARNING

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA-2021

ABSTRACT

MSc THESIS

DETECTION OF CALCIUM DEFICIENCY AND PHYSIOLOGICAL STATUS IN STRAWBERRY LEAVES USING DEEP LEARNING

Muhab HARIRI

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Assist. Prof. Dr. Ercan AVŞAR
Year: 2021, Pages: 61
Juries : Asst. Prof. Dr. Ercan AVŞAR
: Assoc. Prof. Dr. Sami ARICA
: Asst. Prof. Dr. Gökay DİŞKEN

Detecting diseases, disorders and physiological changes in plants is important to prevent crop damage and increase food production. In this work, a CNN network is proposed to detect calcium deficiency and a physiological change related to chlorophyll deficiency in strawberry plant. Transfer learning is applied to some common benchmark models to compare their performance with the proposed model. A dataset of images of strawberry leaves was collected from a greenhouse and used to train the models. Moreover, the dataset contains 1955 images for 3 different classes that are "healthy leaves", "leaves with calcium deficiency" and "old leaves" (lack of chlorophyll). The classification was done in two stages, a binary classification in which healthy leaves and leaves with calcium deficiency were used, the second stage involved multi-class classification after adding old leaves. It has been shown that the proposed model achieved an accuracy of 98.97% which is higher than transfer learning models which are used. In addition, effect of learning rate on the performance of the proposed model was discussed.

Keywords: Deep leaning, Convolutional neural network, Strawberry disorder, Artificial intelligence, Machine learning.

ÖZ

YÜKSEK LİSANS TEZİ

ÇİLEKTEKİ YAPRAKLARINDAKİ KALSİYUM EKSİKLİĞİ VE FİZYOLOJİK DURUMUN DERİN ÖĞRENME KULLANILARAK TESPİT EDİLMESİ

Muhab HARIRI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Dr. Öğr. Üyesi Ercan AVŞAR
Yıl: 2021, Sayfa: 61
Jüri : Dr. Öğr. Üyesi Ercan AVŞAR
: Doç. Dr. Sami ARICA
: Dr. Öğr. Üyesi Gökay DİŞKEN

Bitkilerdeki hastalıkları, bozuklukları ve fizyolojik değişiklikleri tespit etmek, mahsulün zarar görmesini önlemek ve gıda üretimini artırmak için önemlidir. Bu çalışmada, çilek bitkisinde kalsiyum eksikliği ve klorofil eksikliğine bağlı fizyolojik bir değişikliği tespit etmek için bir CNN ağı önerilmiştir. Önerilen model ile performanslarının karşılaştırılması amacıyla bazı bilinen modeller üzerinde transfer öğrenme yöntemi uygulanmıştır. Modelleri eğitmek için kullanılan veri seti bir seradan çilek bitkisinin yapraklarının fotoğraflanması ile oluşturulmuştur. Ayrıca, veri seti "sağlıklı yaprak", "kalsiyum eksikliği olan yapraklar" ve "yaşlı yapraklar" (klorofil eksikliği) olmak üzere 3 sınıf farklı sınıftan toplam 1955 görüntü içermektedir. Sınıflandırma iki aşamada yapılmıştır, sağlıklı yaprak ve kalsiyum eksikliği olan yaprakların kullanıldığı ikili bir sınıflandırma, ikinci aşamada ise yaşlı yapraklar eklendikten sonra çoklu sınıflandırma yapılmıştır. Önerilen modelin, kullanılan transfer öğrenme modellerinden daha yüksek olan % 98,97'lik bir doğruluğa ulaştığı gösterilmiştir. Ayrıca öğrenme oranının önerilen modelin performansı üzerindeki etkisi tartışılmıştır.

Anahtar kelimeler: Deep leaning, Convolutional neural network, Strawberry disorder, Artificial intelligence, Machine learning.

EXTENDED ABSTRACT

Agricultural production provides people with essential requirements to live. Due to the continued rise in the global population, the need to increase agricultural production is growing continuously in order to keep pace with the continuous increase in the demand for food. So, the workers in the agricultural field focus their attention on plant diseases and disorders that may cause crop shortage or potentially eliminate them. Therefore, the main aim is to prevent any problems that may negatively affect crop production and increase their production.

Experts mainly count on visual inspection to diagnose plant diseases which requires experience and may lead to wrong diagnosis. Therefore, using another inexpensive diagnostic method is important to allow diagnosis to be made with confidence, even in the absence of an expert.

The reliance on artificial intelligence (AI) has increased in recent years in all fields of science, as its applications are present in many fields and have become a part of daily life. Agriculture is one of these fields where an AI application can identify plant diseases and disorders. Advantages of this includes, being economic (saves money) and more efficient due to better accessibility, especially for farmers in remote areas. In general, remote areas suffer permanently from a shortage of specialists who have the capability of diagnosing and identifying plant diseases, disorders and physiological changes to determine an appropriate treatment.

Tipburn of strawberry is a kind of disorder that is associated with calcium deficiency in the plant. Symptoms associated with this disorder begin with a dark-colored on the edge of the leaf and later develop into necrotic edges. Another type of defect that affects strawberry is associated with lack of chlorophyll and is a result of aging of the leaf. The symptoms can be diagnosed by visual inspection, as the affected crops present leaves of a yellowish color. In this work, convolutional neural networks (CNN), a deep learning method, were utilized for automatic detection of calcium deficiency (tipburn) and aged leaves (old leaves) in strawberry leaves.

Deep learning is an AI method that deals with finding functions to allow the machine to work without directly being controlled by a person. A Convolutional Neural Network (CNN) is a deep learning algorithm that is commonly used in computer vision applications. CNN processes the images for detection, recognition or classifications tasks.

The image dataset of strawberry leaves used in this work was collected by taking photos from a greenhouse at an experimental farm in Çukurova University. The images in the dataset were manually labelled with one of the three classes by an expert. It contains 1955 images distributed as follows: 626 of healthy leaves, 805 of leaves with calcium deficiency and 524 of old leaves.

The dataset was divided to three sets: training, validation and test sets and then data augmentation was applied to the training set.

Data augmentation is a powerful technique used in deep learning, it prevents overfitting and makes the dataset worthy and more diverse. After data augmentation process, the number of images in the dataset was increased to 9689 images distributed as follows: 3436 of healthy leaves, 3217 of leaves with calcium deficiency and 3036 of old leaves.

The classification was done using transfer learning models and by a proposed CNN architecture. The models used for transfer learning were MobileNetV2, EfficientNet, InceptionV3, VGG16, VGG19, InceptionResNetV2, ResNetV2, and NASNetMobile. Training of transfer learning models were made using various collections of batch sizes and a number of training epochs. On the other hand, the proposed CNN model has a sequential of convolutional layer, pooling layer, and fully-connected layer with using dropout layer as regularization and ReLU as activation function.

Also, two different stages of classification were performed to give more details about the performance of the proposed model. The first stage included a binary classification in which healthy leaves and leaves with calcium deficiency were used, and the second stage involved multi-class problem after involving the images of the old leaves in training and testing of the models.

Adam optimizer was used for both of classification stages with default learning rate for the proposed model in binary classification and transfer learning models. The value of learning rate was tuned in multiple classification to investigate its effect on the performance of the model. Confusion matrix is generated to give more detailed examination. In addition, performance metrics such as accuracy, precision, recall, and F1-score were calculated

The results that were obtained in binary classification through transfer learning showed that MobileNetV2 and EfficientNet models achieved the highest accuracy with 99.65%, these two models have only one misclassification image of the 287 images in the test data. The misclassified sample was a false positive for MobileNetV2 and a false negative for EfficientNet. As for the proposed model, the test accuracy was 99.30%. On the other hand, in multi-class problem, EfficientNetB0 achieved the highest accuracy with accuracy of 97.95% among the transfer learning models. With regard to the proposed CNN architecture, the prediction accuracy was 98.97% after tuning the learning rate.

The results showed that increasing the depth of the model does not help to increase the accuracy of prediction. In addition, using models with high number of parameters is unnecessary for classification task with low number of classes.

This work confirms that tuning the hyper parameters is effective approach to increase the performance of the models, as tuning the learning rate in multiple classification produced and increasing in the accuracy of the models.

What distinguishes this work is the database which contains images are taken directly from the cultivated strawberry leaves whereas, in other similar work the database contains images taken under laboratory conditions. Therefore, the database which was used in this work is more realistic and hence the proposed model can be considered to be more appropriate for hands-on applications.

This work shows that using deep learning technique in detecting plant disorders and physiological changes is effective when images collected directly from the field are used.

ACKNOWLEDGEMENT

I express my sincere gratitude to Dr Öğr. Üyesi Ercan AVŞAR for supporting me and providing his advices to me during these past two years. Also, I would like to thank jury members of my master thesis and faculty members of Electrical and Electronics Engineering Department in Çukurova University.

In addition, I would like to thank Dr. MEHMET ALİ SARIDAŞ in Horticulture Department in Çukurova University for sharing his experience and for his contribution to this work.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

ABBREVIATIONS

AI	: Artificial Intelligence
DL	: Deep Learning
CNN	: Convolutional neural network
TL	: Transfer Learning
GD	: Gradient Descent
SGD	: Stochastic Gradient Descent
RMSProp	: Root Mean Square Propagation
FC	: Fully Connected
FN	: False Negative
FP	: False Positive
TN	: True Negative
TP	: True Positive
Tanh	: Hyperbolic Tangent Function
ReLU	: Rectified Linear Unit

1. INTRODUCTION

1.1. Description of Plant Diseases, Disorders and Physiological Changes

Tipburn of strawberry is one of the disorders that results from a local deficiency of calcium in the plant causing a decrease in photosynthetic activity. Physiological calcium deficiency is commonly linked to the failure of the plant to transfer sufficient levels of calcium to all leaf areas (Olle and Williams 2017). Symptoms associated with this disorder begin with a dark-colored on the edge of the leaf and later develop into necrotic edges.

Calcium is considered as an important component for safeguarding the cell wall health. Disturbances in the absorption and transport of calcium throughout the plant lead to a weakness in the cells around the border of leaves. This disorder results in a crop with wrinkled leaves and may cause reduced production. Therefore, detection of calcium deficiency is important to provide timely treatment. This disorder can be treated by controlling the environmental conditions such as humidity, light density and soil moisture (Kuronuma, Watanabe et al. 2018, Bárcena, Graciano et al. 2019).

Lack of chlorophyll which is a result of aging leaves is also considered as a problem and can cause physiological changes in strawberries. The resulting symptoms can be diagnosed by visual inspection, as the affected crops present leaves of a yellowish color.

In general, diseases and disorders cause a failure in plant growth. Disorders are caused by abiotic factors such as light intensity, humidity or nutrient quality whereas, diseases are caused by biological factors such as various viruses and bacteria that infect plants and then gradually spread among them.

Visual inspection is considered as common method for experts and farmers to diagnose diseases, disorders and physiological changes that affect the plant. This method may lead to uncertain or incorrect diagnosis therefore, utilizing an automated and accessible method for diagnosing is considered a promising way to cure the plants and provide the appropriate treatment more rapidly.

1.2. Problem Statement and Proposed Method

Early diagnosis of plant diseases, disorders and defects and providing them with appropriate treatment are one of the factors that lead to increased production. Farmers in remote areas of the world suffer from lack of expertise and knowledge in order to be able to diagnose plant diseases. This can have an impact on crop damage due to the postponement in identifying and providing appropriate treatment.

Deep learning, that allows machines to learn from data, has become widespread in recent years where many areas of life have become dependent on it to handle a specific problem. Image classification is an example of deep learning applications. It can be described as the process of analyzing an image in order to classify it to a specific class. Image classification can be utilized in various applications such as self-driving cars and face recognition as well as agricultural problems.

In this thesis, image classification technique is used to detect disorders and physiological changes in strawberry plants using a dataset that consists of 1955 images and includes 3 classes. The images are taken from a greenhouse at an experimental farm of the University of Cukurova.

Convolutional neural network which is a deep learning algorithm and is considered as one of the major tools for image classification. A convolutional neural network uses its layers to processes the images and then classifies it to a specific class in the output layer.

For the purpose of this thesis, a convolutional neural network model, which contains a number of convolutional, pooling and fully-connected layers, is proposed. In addition, transfer learning models are used to provide more understanding about the effectiveness of the proposed model by comparing the results.

Transfer learning is a machine learning method which indicates the use of a model that has been trained for a specific task but to process another related task. The models used for transfer learning were MobileNetV2, EfficientNet, InceptionV3, ResNetV2, VGG16, VGG19, InceptionResNetV2, and NASNetMobile.

Learning rate is one of the topics that is discussed in this work as well. It is the hyperparameter that controls the speed of learning of the convolutional neural network model until it arrives to the lowest value of error. Furthermore, it is considered as one of the most important adjustable values to reach the best efficiency for the model.

In this work, the classification was done in two stages, a binary classification in which healthy leaves and leaves with calcium deficiency were used, the second stage involved multi-class classification after adding old leaves. The effect of changing learning rate value on the proposed model was also investigated.

What distinguishes this work is the database which contains images are taken directly from the cultivated strawberry leaves whereas, in other similar work the database contains images of the leaves after picking and photographing it under laboratory conditions. Therefore, this work analyzes the results under more realistic conditions and hence the proposed model can be considered to be more appropriate for hands-on applications.

This work draws attention to the effectiveness of the deep learning technique in detecting plant diseases and disorders, using images captured in the field directly by smartphones. In addition, it is considered as a base for future application which can detect the diseases, disorders and physiological changes in strawberry plant after adding more diseases and disorders that affect the plant.

1.3. Outline of The Thesis

The rest of this thesis is organized as follows: Section 2 discusses the related works. Section 3 discusses basic concepts about deep learning, neural networks, convolutional neural network, optimization, regularization and learning rate. Section 4 presents the dataset used for training and testing, and experiments used to confirm the ability of deep neural networks to predict disorders and physiological changes in strawberry plants. Section 5 discusses the results after applying the models that were referred in “Problem Statement” section and the effect of changing the learning rate. Finally, Section 6 summarizes this work.

2. RELATED WORKS

One of the branches of deep learning methods is a conventional neural network (CNN). CNN is widely used in images data for many tasks like segmentation, object detection and image recognition. Agriculture is one of the common applications of CNN. Many researches have shown that using CNN in agricultural issues can be very beneficial, where datasets containing images collected from plants are used to do many tasks like weed recognition (Dyrmann, Karstoft et al. 2016), plant species detection (Kaya, Keceli et al. 2019), diseases detection in plants (Abdullahi, Sheriff et al. 2017, Amara, Bouaziz et al. 2017, Ferentinos 2018) and for counting fruits (Rahnemoonfar and Sheppard 2017).

Detection of plant diseases is an essential and complex issue since there are many forms of diseases in various species that may have different visual appearances. Mohanty et al. used a public database to classify 26 different diseases in 14 crop species using CNN architectures like AlexNet and GoogLeNet(Prasanna Mohanty, Hughes et al. 2016). Konstantinos P. Ferentinos used an open database that contains tens of thousands of photos images, includes 25 different plants in a set of 58 distinct classes(Ferentinos 2018). Many CNN architectures like Overfeat, AlexNet, AlexNetOWTBn and VGG are used for training the database.

Other works are done for detection the diseases in a single type of plant. for instance, Halil Durmus et al. Used deep learning network architectures like AlexNet and SqueezeNet for detecting some diseases that infect the plants of tomatoes (Durmuş, Güneş et al. 2017). They used a dataset provided by Plantvillage which contains 54309 images taken under controlled conditions. In a piece of work that used the same dataset, a CNN model based deep residual Learning method is used for lower computational load. Another works focused on a specific species like apple (Liu, Zhang et al. 2018), mango (Singh, Chouhan et al. 2019), wheat (Lu, Hu et al. 2017, Picon, Alvarez-Gila et al. 2019) , bell pepper leaf (Bhagat, Kumar et al. 2020), banana (Singh and Athisayamani 2020), tomato leaf (Sardogan, Tuncer et al. 2018), wheat plant (Genaev, Ekaterina et al. 2020), soya bean plant (Kashyap and Shrivastava 2020), rice leaf (Ghosal and Sarkar 2020) and cassava (Ramcharan, Baranowski et al. 2017).

There are various additional recent work focused on detecting plant diseases. For example, Santhana Hari et al. used a proposed model to detect diseases in 5 crop species like apple and maize. The model achieved an accuracy of 86% (Hari, Sivakumar et al. 2019). Akbar Hidayatuloh et al. used dataset that contains 1400 images for seven classes of tomato leaves including healthy leaf class. The average accuracy of detection in this class was 86.92% (Hidayatuloh, Nursalman et al. 2018). A deep CNN architecture was proposed by Madhulatha et al. to classify 54,323 plant leaves images of 38 different species. The best accuracy that was achieved 96.50% (Madhulatha and Ramadevi 2020). 38 different categories of plant images were classified using a dataset of 87,867 images using CNN network developed by Mukti et al. (Mukti and Biswas 2019). The proposed model attained overall accuracy of 99.80%. Rahmat Ullah et al. used CNN model to detect diseases in a total of 38 classes, compromises 12 classes as a healthy class. The model achieved recognition rate of 97.33% (Ullah, Dola et al. 2019). Jasim et al. collected images from ‘Plant Village dataset’ website that includes images of tomatoes, pepper, and potatoes. The dataset contains 12 classes of plant diseases and comprises of 20,636 images. The test accuracy was 98.29% (Jasim and AL-Tuwaijari 2020).

In a method developed for strawberry plants, Park et al. presented an approach based on CNN to diagnose the healthy and disease strawberry using images of its leaves and fruits (Park, JeeSook et al. 2018). The database was generated using laboratory conditions where the chambers are used to intentionally cause the disease. The database contained powdery mildew, gray mold rot, fusarium wilt and anthracnose diseases. The work attained test accuracy of 89.7% and 97.4% for leaf and fruit diseases experiments respectively.

Most of the previous work focused on disease detection. On the other hand, detection of disorders and physiological changes were not discussed. The authors of the two earlier works (Khan and Narvekar 2020, Khan and Narvekar 2020) used the terms disorder and disease to describe same health problem in the plants that were covered in their work. This reduces reliability as the meaning of disease differs from the meaning of disorder in agriculture.

3. MAIN CONCEPTS

3.1. Machine Learning and Deep Learning

Machine learning is one of the branches of artificial intelligence. It gives machines the ability to learn through training without the intervention of a human programmer. It can be said that machine learning is a system that has been trained instead of being clearly programmed. In other words, a machine learning system converts input data into meaningful outputs (Chollet 2018).

Supervised machine learning is considered as the most prominent part of machine learning. The algorithm in supervised machine learning is fed with a pair of input and output to let the algorithm know if it is classifying the data correctly or not. In contrast, unsupervised machine learning works when the algorithm is fed with the data without outputs in order to collect, classify or organize data according to its features in a way that enables humans to understand it.

In conjunction with the development of processors and the availability of huge amounts of data, the term "Deep Learning" has been emerged. Deep learning refers to machine learning algorithms that handle with a massive amount of data and use a relatively large number of layers to process it. Therefore, deep learning is part of machine learning, which is part of artificial intelligence (Figure 1).

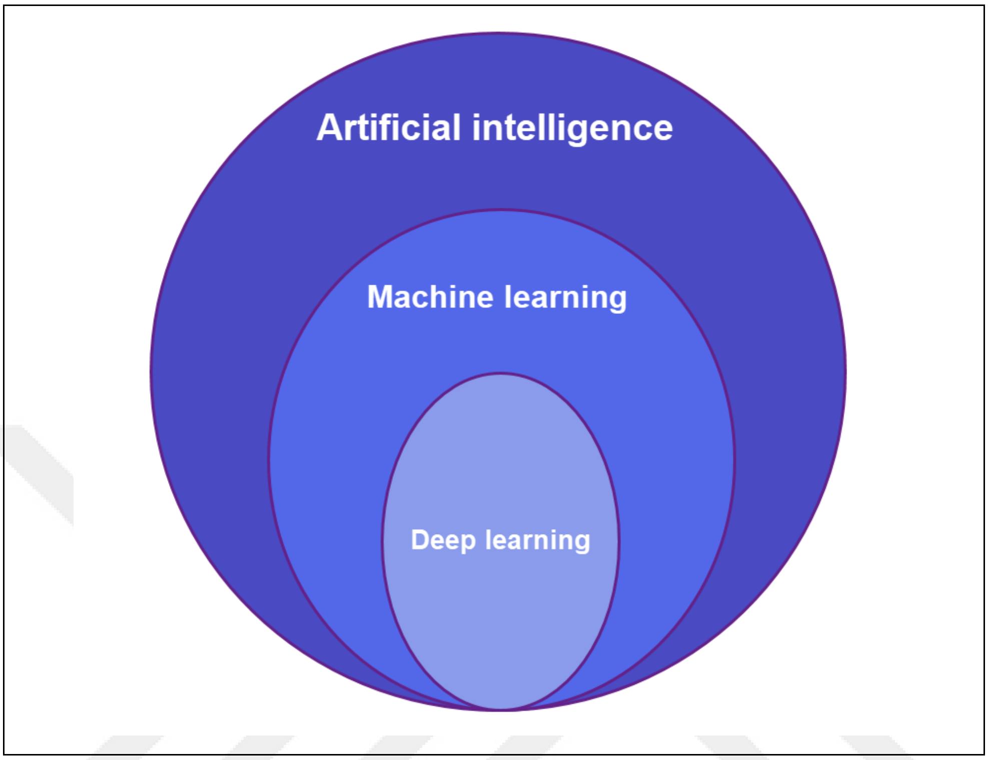


Figure 3.1. Illustration of Deep Learning as a subset of both Machine Learning and Artificial Intelligence

Deep learning handles a multi-layered structure of algorithms called neural networks, which are able to perform many tasks like classification or regression.

A typical neural network structure consists of a large number of nodes called "neurons" that are arranged in several layers. These layers start with the input layer that receives the data and end with the output layer which represents the final result, passing through a group of hidden layers where all the processing actually happens.

3.2. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning method which is used to recognize and classify the images and visual data or for being detected. These tasks can be done using neural networks but the images mostly consisting of large number of pixels which can be an issue because this makes data processing in neural networks very slow and time consuming. CNN, which extract the features from images, can solve this problem using its layers.

Convolutional Neural Networks generally consist of many layers:

- Input layer
- Convolutional layer with nonlinear activation function
- Pooling layer
- Fully connected layer
- Softmax layer
- Output layer

3.2.1. Input Layer

Input layer receives the image data. A regular color image data is a 3-dimensional matrix because it has three channels for the main colors: red, green and blue.

3.2.2. Convolutional Layer with Nonlinear Activation Function

Convolutional layer represents the first layer in CNN models. The task of convolutional layer is to extract the features in the image using filters. The filter, with specific dimensions, slide over the image step by step and the dot product is taken between the filter and the specific part of the input image. The output is indicated to as the feature map. Convolutional layers reduce the image size but maintains its basic features. These operations are illustrated in Figure 2.

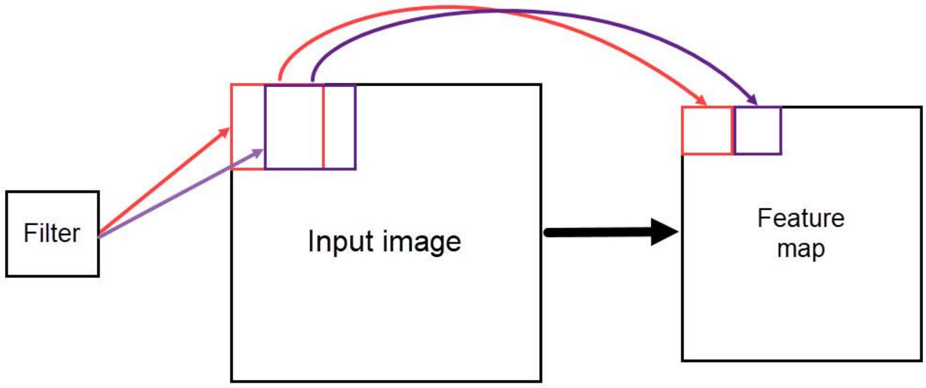


Figure 3.2. Convolutional layer operation.

After each convolutional layer a nonlinear activation function is applied. This is an additional step in convolutional operation. The idea of this layer is to increase the non-linearity in the input image where linearity is imposed during the convolutional process.

Rectified linear unit (ReLU) function has been proven that it works better than other nonlinear functions like tanh and sigmoid, as it can accelerate the training process and can solve the vanishing gradient problem. Figure 3 explains the difference between ReLU and other activation functions. The related functional expressions are provided in equations 3.1, 3.2, 3.3 and 3.4, respectively.

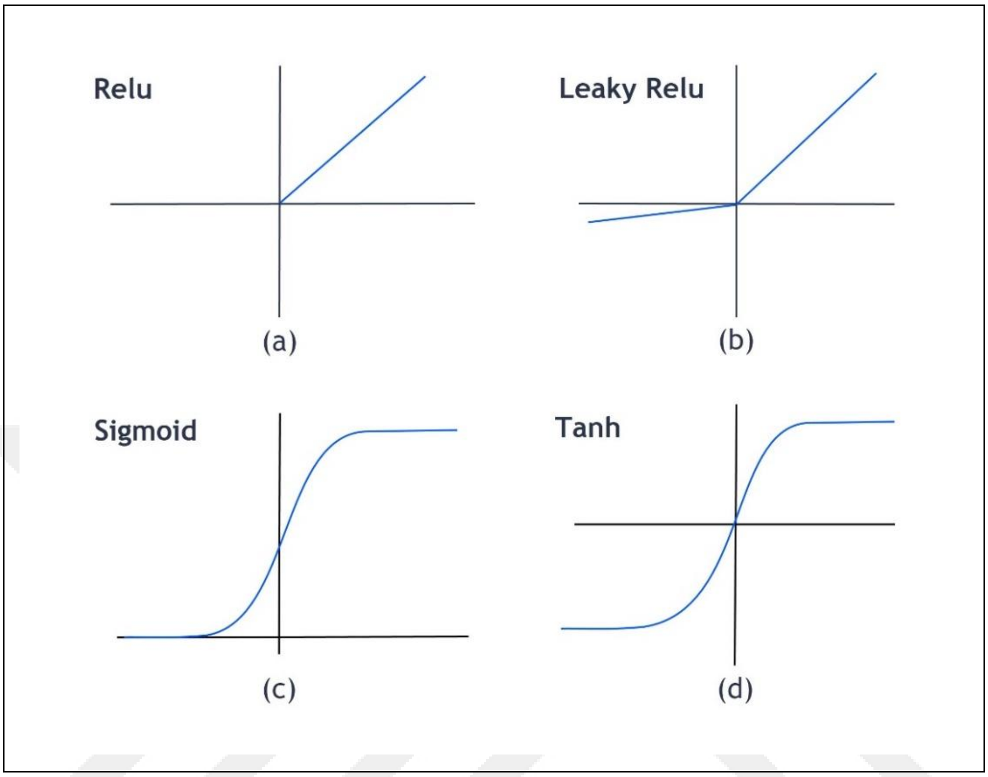


Figure 3.3. Comparison between activation functions: (a) Relu, (b) Leaky Relu, (c) Sigmoid (d) Tanh.

$$g(x) = \max(0, x) \tag{3.1}$$

$$g(x) = \max(\delta x, x) \quad with\ \delta \ll 1 \tag{3.2}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.4}$$

ReLU function returns the provided value as direct input and changes all the negative values to zero. Therefore, it is described as a simple mathematical operation.

3.2.3. Pooling Layers

Pooling layers are responsible for reducing the dimensions of the feature maps in order to reduce the amounts of parameters and computation. Pooling operation is done using two-dimensional filter slides over each feature map independently. Max Pooling is the common function used in pooling operations. It picks out the maximum value in each window covered by the filter. The output feature map includes the eminent features existing in the former feature map. Figure 4 shows an example of a 2x2 max pooling operation with a stride of 2.

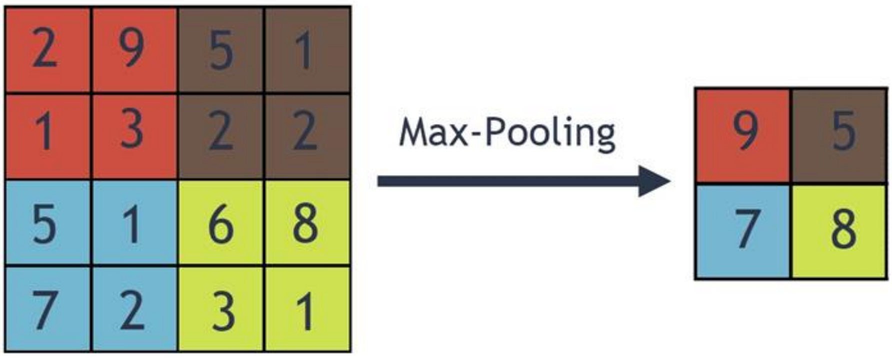


Figure 3.4. Max pooling operation using 2x2 filter.

3.2.4. Flatten Layer

Flatten layer convert a two-dimensional feature map resulted from pooling operations to one-dimensional feature vector (single array). This stage is important to make the fully connected layer, which does not process data with multiple dimensions. Figure 5 explains flatten layer operation.

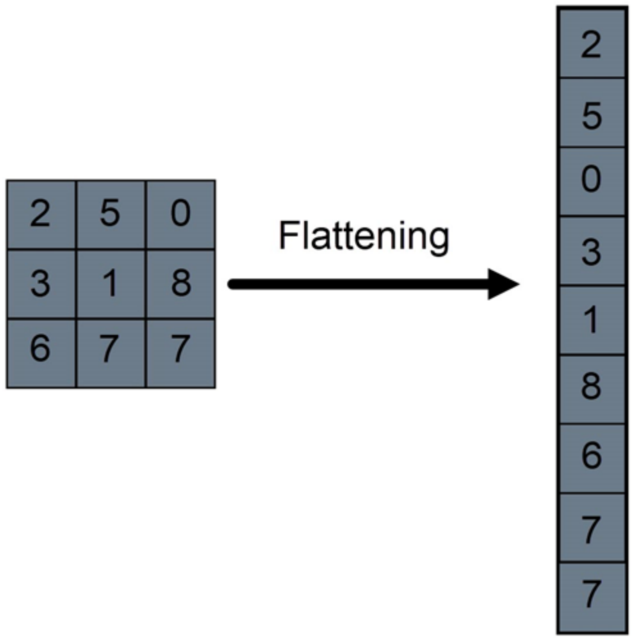


Figure 3.5. Flattening 2D feature map to 1D vector.

3.2.5. Fully Connected Layers

Fully connected layer is the same as regular artificial neural network and considered as the final classification stage. This layer is responsible for using high-level features, which resulted from convolutional and pooling operations, to assign a class label to the input image. A fully connected neural network consists of successive layers where each neuron in each layer connected with every neuron in the next and previous layer.

3.2.6. Softmax and Output Layers

At the end, there is an output layer which is responsible for producing the final result as probability using Softmax layer as activation function (Heaton 2018). The sum of output probabilities is one.

3.3. Optimization

The optimizer is one of the major issues to focus on in deep learning models. The use of an appropriate optimizer allows to achieve the best result since the optimizer is responsible for updating the weights to minimize the loss function.

Some types of optimizers in deep learning are given as subsections.

3.3.1. Gradient Descent with Momentum

Gradient descent (GD) is a simple optimization procedure to optimize the loss and accuracy of neural networks.

The idea of gradient descent is based on iterative updating of parameters using all the samples in the database for minimizing the error function. This operation is only valid if the database is small because it is time consuming. On the other hand, if the database is large, this process will be computationally expensive and time consuming since it involves also multiple back-propagation calculations. Therefore, a newer method called Stochastic Gradient Descent (SGD) has been developed. The idea of SGD is to pick one instance from a training set at every step and update parameters only based on that single and randomly selected record. The Figure 6 explains the difference between GD and SGD, whereas the GD operation looks like more stable in the way to global optimum, the SGD is faster for optimization.

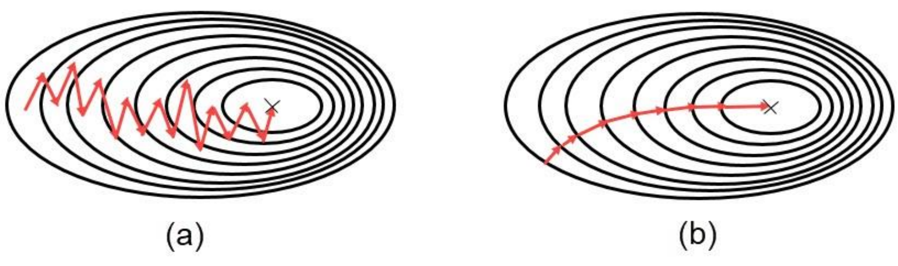


Figure 3.6. The difference in the convergence between SGD (a) and GD (b).

Stochastic Gradient Descent with Momentum is an update that has been added to SGD. It helps the original stochastic gradient descent algorithm to estimate in the relevant direction rather than randomly estimating as shown in Figure 6. This update is considered as a time-saving approach that works faster with greater precision.

The oscillations in stochastic gradient descent are very high. That can prevent from using high learning rate to accelerate the processing. For using a higher learning rate, the momentum should be used. Momentum helps to accelerate SGD in the relevant direction and damps oscillations by adding a fraction or part of previous time step into the current time step (Qian 1999).

The basic idea of SGD with momentum is to calculate exponentially weighted average of the gradients for using them in parameters updating.

The equations of gradient descent with momentum are as follows:

$$vdw = \beta \cdot vdw + (1 - \beta) \cdot dw \tag{3.5}$$

$$vdb = \beta \cdot vdb + (1 - \beta) \cdot db \tag{3.6}$$

$$W = W - \alpha \cdot vdw \tag{3.7}$$

$$b = b - \alpha \cdot vdb \tag{3.8}$$

Here, β represents the momentum and its range extends from 0 to 1 whereas the value of 0.9 is suggested. The value α represents the adjustable learning rate that is used for updating the model parameters W and b.

3.3.2. RMSprop

RMSprop optimizer term refers to "Root mean square propagation optimizer" and was introduced first by Geoffery Hinton (Tieleman and Hinton 2012). This method is a type of gradient descent algorithm and similar to SGD with momentum. The idea of RMSprop optimizer is smoothing the high fluctuation while increasing the learning rate in the relevant direction to global optimum, that can be done by taking moving average of the squared of gradients. A major advantage of RMSprop is that the learning rate is adjusted automatically and enables to choose different learning rates for different parameters in the model

The equations of RMSprop optimizer are as follows:

$$vdw = \beta \cdot vdw + (1 - \beta) \cdot dw^2 \tag{3.9}$$

$$vdb = \beta \cdot vdb + (1 - \beta) \cdot db^2 \tag{3.10}$$

$$W = W - \alpha \cdot \frac{dw}{\sqrt{vdw} + \varepsilon} \tag{3.11}$$

$$b = b - \alpha \cdot \frac{db}{\sqrt{vdb} + \varepsilon} \tag{3.12}$$

The value of vdw is relatively smaller than vdb because the slope moves slowly in the direction of global optimum comparing with its sharp fluctuations that presented by db. In addition, element wise operation makes the value of dw smaller while db will be relatively larger, so the updates W and b will make the slope move faster to global optimum while the oscillation will be lighter.

The value of momentum, β, is usually set to 0.9. The epsilon, ε, value in the denominator is very small. The target of using ε is to prevent getting a zero value in the denominator as vdw gradually gets closer to zero.

3.3.3. Adam

Adam stands for adaptive moment optimization (Kingma and Ba 2014). It is also one of the most widely used optimizers recently in deep learning models because of its variety of benefits. The method is easy to use, computationally efficient and requires a little memory. In addition, it is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients.

Adam optimizer is a replacement of existing stochastic gradient algorithms such SGD with momentum and RMSprop. Basically, it uses the advantages of RMSprop and SGD with momentum simultaneously. The optimizer is designed to use high learning rate like RMSprop and get moving average of gradients like SGD with momentum. Therefore, its equations are combinations of both of these two methods. The equations of Adam optimizer are as follows:

$$Vdw = \beta_1 \cdot Vdw + (1 - \beta_1) \cdot dw \tag{3.13}$$

$$Vdb = \beta_1 \cdot Vdb + (1 - \beta_1) \cdot db \tag{3.14}$$

$$Sdw = \beta_2 \cdot vdw + (1 - \beta_2) \cdot dw^2 \tag{3.15}$$

$$Sdb = \beta_2 \cdot vdb + (1 - \beta_2) \cdot db^2 \tag{3.16}$$

Updated weights and Biases are calculated as follows:

$$V^{corr}dw = \frac{V_{dw}}{1 - \beta_1{}^t} \quad 3.17$$

$$V^{corr}db = \frac{V_{db}}{1 - \beta_1{}^t} \quad 3.18$$

$$S^{corr}dw = \frac{S_{dw}}{1 - \beta_2{}^t} \quad 3.19$$

$$S^{corr}db = \frac{S_{db}}{1 - \beta_2{}^t} \quad 3.20$$

$$W = W - \alpha \cdot \frac{V^{corr}dw}{\sqrt{S^{corr}dw} + \epsilon} \quad 3.21$$

$$b = b - \alpha \cdot \frac{V^{corr}db}{\sqrt{S^{corr}db} + \epsilon} \quad 3.22$$

The equations show that Adam optimizer has properties of both RMSprop and SGD with momentum. In practical use, the hyperparameters like α should be tuned while β_1 and β_2 commonly use the values of 0.9 and 0.999. ϵ usually has the value of 10^{-8}. Using the previous values are not mandatory; however, it is observed that Adam optimizer usually works better when they are used.

One of the most important tuning parameters in the optimization method is learning rate.

3.4. Learning Rate

Learning rate is a hyperparameter that determine the size of the step and controls the speed of learning (Goodfellow, Bengio et al. 2016), it represents the amount of the weights that is updated during the optimization (Patterson and Gibson 2017). Learning rate is considered as the most important hyperparameter that is worth tuning, it can be range between 10^{-6} and one (Bengio 2012).

Using a high learning rate sometimes accelerate the learning process but it can lead to an increase in the error rate or to miss global optimum so the algorithm will fail to converge (Murphy 2012). On the other hand, a low value of learning rate can lead to long training time and may not reduce the error rate.

As expected, it is not feasible to define a value for learning rate that is suitable for all problems. The optimal value for the learning rate in a specific problem needs to be sought by observing the prediction performance for different values. In this work, many learning rate values are used in the proposed model in multiple classification. Changing the learning rates showed big difference in the resulted accuracy.

3.5. Regularization

Overfitting is one of the problems that obstructs deep learning models, as the model achieves high accuracy in the training data but fail to predict test data. Regularization is commonly used method to prevent from overfitting. Regularization process tries to control the weights associated with the features that cause the overfitting, so it leads to reduce the impact of these weights on the model.

Dropout is one of the regularization techniques. It drops randomly and temporally the neurons in neural network layers during training, this approach is beneficial to overcome overfitting) Srivastava, Hinton et al. 2014(. The advantage of using dropouts is that no high weight values will be assigned to a neuron in the model, so the weight values will lose its exaggerated effectiveness that causes overfitting. This method prevents the output of the layers to depend on specific neurons.

4. MATERIALS AND METHODS

4.1. Dataset Description

The dataset contains 1955 images of 29 different genetic types of strawberry divided into three classes; “healthy”, “calcium deficient” and “old leaf” which were determined by an expert. The images were collected from a greenhouse located in Adana, affiliated with University of Çukurova. The greenhouse is a part of an experimental farm belonging to the Horticulture Department of the university. The dataset contains 626, 524 and 805 images from “healthy”, “old leaves” and “calcium deficiency” labels, respectively. Figure 7 shows some sample images in the dataset.

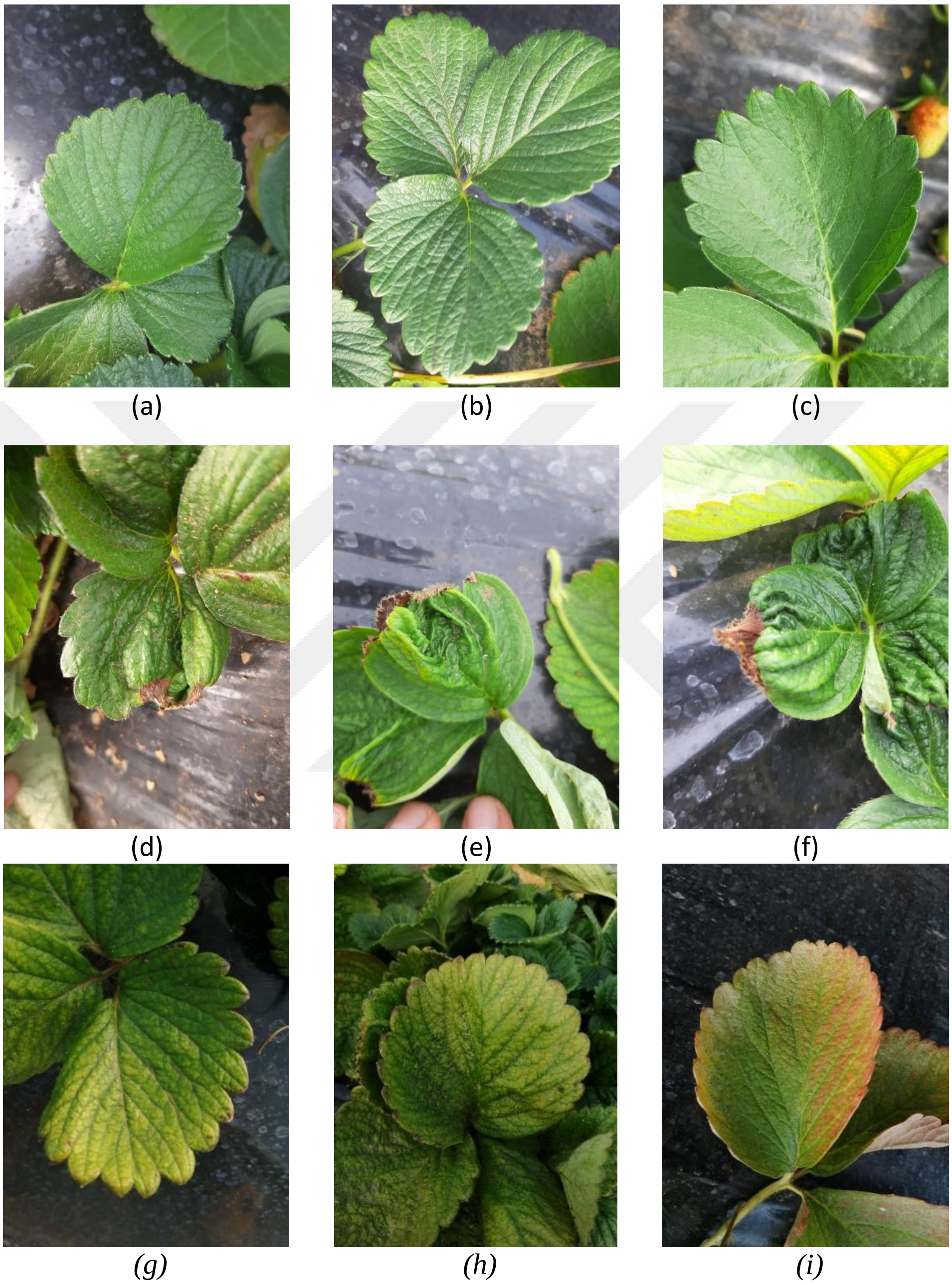


Figure 4.1. Samples of images of healthy leaves (a-c), leaves with calcium deficiency (d-f) and old leaves (g-f).

4.2. Data Augmentation

Data augmentation is a powerful technique used in deep learning to increase the size and diversity of data by generating new samples using existing ones (Mikołajczyk and Grochowski 2018). Furthermore, it has a regularization effect so helps to prevent overfitting and makes the dataset worthy.

Before applying data augmentation, the dataset was segmented into training, validation and test sets with a ratio of 60%, 20%, and 20%, respectively - this splitting ratio is most commonly used in neural network applications. Then data augmentation was applied to the training set which increased the number of images in the training set to 8905. The details of the augmented dataset are shown in **Hata! Başvuru kaynağı bulunamadı.**.

Table 4.1. The details of the augmented dataset.

Class Name	Number of images	Train/Validation/Test Split
Healthy	3436	3184/126/126
Calcium deficiency	3217	2895/161/161
Old leaves	3036	2826/105/105

4.3. Experiments

There are many convolutional neural network structures that are commonly used in the classification tasks. These CNNs are developed to achieve a good performance on database containing huge number of samples and hundreds of classes. These networks consist of many convolutional layers in the upper part which include a large number of parameters that should be optimized during the training process. The parameters in this part of pre-trained models are related to the problem so it should be updated when dealing with another problem. Whereas, the lower layers related with common features that are connected with many similar problems and can be used in different tasks, this approach is known as learning transfer. Several common transfer learning models are used in this work. In addition, smaller convolutional neural network is proposed to compare its performance with that of previous models.

4.3.1. Transfer Learning

Transfer learning is a machine learning technique in which an available model architecture that has been built and trained for a specific task is applied to a different but related task.

Many pre-trained CNN models were used in this work with adding a dense layer on top of the feature extraction part for each model. The weights of these models are trained using ImageNet database which contains more than a million of images classified into hundreds of labels (Russakovsky, Deng et al. 2015). These pre-trained models are VGG16, VGG19, MobileNetV2, ResNet152V2, EfficientNet, NASNetMobile, InceoptionV3 and InceptionResNet. The weights of the added dense layer are trained using the features extracted by the previous transfer learning models.

Since the task of this work is to detect two and three classes separately, the dense layer for all the models is set to have two and three outputs, respectively.

VGG16 model is an advancement of already existent “AlexNet” model but with a special architecture and more layers (Simonyan and Zisserman 2014). The model consists of 13 convolutional layers with 5 max pooling layers followed by 3 dense layers. The model is highly efficient but considered as time consuming and computationally expensive task because of its dense layers which includes high number of parameters. VGG19 is as new version of VGG16 with deeper architecture as it contains more 3 convolutional layers.

MobileNetv2 is updated version of MobileNetV1. The architecture of this model is designed to work better with mobile or devices with low computational power (Sandler, Howard et al. 2018). MobileNetV2 uses depth-wise separable convolutions, rather than standard convolution, that consumes less computational power. This leads to a reduction in model size.

EfficientNet is a name for a group of CNN models, it consists of eight different structures (B0 to B7). These structures are obtained by scaling all dimensions of network depth, width and resolution using a set of fixed scaling coefficients (Tan and Le 2019). Each model in the group consist of seven main blocks. These blocks include many sub-blocks which its number increases gradually from EfficientNetB0 to EfficientNetB7 and makes the model deeper. EfficientNetB0 and EfficientNetB2 models are used in this work and they consist of 237 and 340 layers, respectively.

Inception-v3 was introduced by google. It is designed to have small number of parameters with high depth (42 layers) but it is more efficient compared with VGGNet (Szegedy, Vanhoucke et al. 2016). Inception-v3 depends on factorization to reduce number of parameters with maintaining the efficiency. In addition, Inception-v3 uses auxiliary classifier to regularize and this is what set differs it from Inception-v1.

ResNet stands for residual networks and has a special feature know as skip connection. This is where the main input is fed independently to each layer, which means that each successive layer has two inputs. One from the previous layer and the second one from main input. Skip connection is considered as a successful method to eliminate vanishing gradient. ResNet have many versions with different number of layers. The second version of ResNet includes a batch normalization procedure before each layer (He, Zhang et al. 2016). In this work, ResNetV2-152 version which has 152 layers was used.

InceptionResNet2 is a hybrid architecture inspired both by Inception and Resnet builds. The network is 164 layers deep and mixes between inception modules with residual connections. The performance of the network is notably increased but the computational cost also increased (Szegedy, Ioffe et al. 2017).

NASNet stands for Neural Architecture Search network which was first introduced by Google AI. It basically uses reinforcement learning to build new model architecture (Zoph and Le 2016). The network consists of two kinds of cells or blocks, Normal cells and Reduction cells. Normal cells are CNN layers that are responsible about returning the feature maps with same dimensions as the input image. Reduction cells are CNN layers that are responsible about returning feature maps with reducing the height and width by a factor of two. NASNet has many versions, in this work the mobile version is used.

4.3.2. The Proposed Architecture (The Sequential Model)

The size of the input images to be received by the input layer in the proposed model is 224x224x3. The model consists of five 2D convolutional layers followed by four fully connected layers. The first 2D convolution layer has 35 filters with size of 5×5. The number of filters in the following convolution layers are 50, 65, 90 and 125, respectively, with size of 3×3. Max pooling layer is applied after each convolutional layer with a 2x2 window and a stride of 2. In addition, dropout layers are added with rate of 0.05 after each combination of convolution and maxpooling layers. The resulted tensor after applying these layers has a shape of 5x5x125. Then a flatten layer is used to convert the resulted tensor to vector. The followed fully connected layers have a unites of 256, 128 and 64 while last one has 2 unites for binary classification and 3 unites for multiple classification. The rate of dropout layer after each fully connected layer is 0.4. Relu activation function is applied to all convolutional and fully connected layers except the last fully connected layer where a softmax activation function is applied. These steps of the proposed architecture are illustrated in Figure 8.

Details like the total number of parameters and the number of learnable parameters for the sequential model as well as the benchmark models are given in the Table.4 2 and Table 4.3.

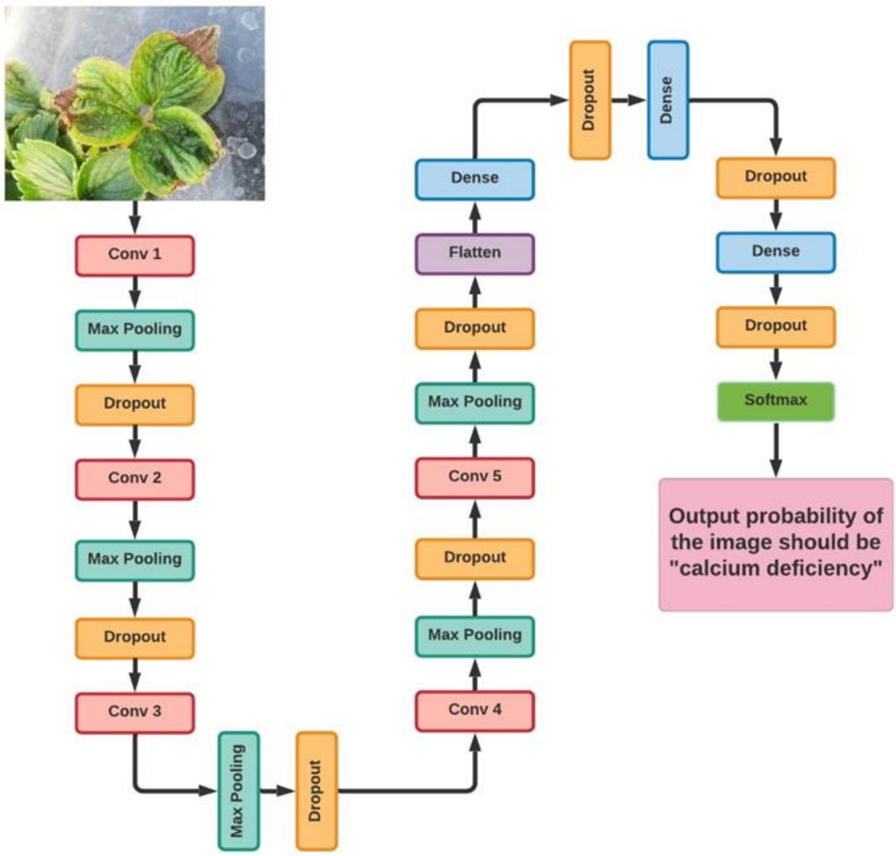


Figure 4.2. The proposed architecture.

Table 4.2. The total number of parameters for each model.

Model Name	Input Image Dimensions	Total Number of Parameters	
		Multi-class Classification	Binary Classification
VGG16	224x224x3	134,272,835	134,272,835
VGG19	224x224x3	139,582,531	139,582,531
MobileNetV2	224x224x3	2,261,827	2,261,827
EfficientNet B0	224x224x3	4,053,407	4,053,407
EfficientNet B2	260x260x3	7,772,789	7,772,789
InceoptionV3	299x299x3	21,808,931	21,808,931
ResNetV2	224x224x3	58,337,795	58,337,795
InceptionResNetV2	299x299x3	54,341,347	54,341,347
NASNetMobile	224x224x3	4,272,887	4,272,887
Proposed model	224x224x3	1,043,493	1,043,428

Table 4.3. Number of trainable parameters for each.

Model Name	Input Image Dimensions	Number of Trainable Parameters	
		Multi-class Classification	Binary Classification
VGG16	224x224x3	12,291	8,194
VGG19	224x224x3	12,291	8,194
MobileNetV2	224x224x3	3,843	2562
EfficientNet B0	224x224x3	3,843	2562
EfficientNet B2	260x260x3	4,227	2818
InceoptionV3	299x299x3	6,147	4098
ResNetV2	224x224x3	6,147	4098
InceptionResNetV2	299x299x3	4,611	3074
NASNetMobile	224x224x3	3,171	2114
Proposed model	224x224x3	1,043,493	1,043,428

4.3.3. Implementation Details

All of the transfer learning and proposed models in this paper were trained using Adam optimizer. With regard to transfer learning models, the learning rate is tuned to 0.001 (default value) while exponential decay rates are chosen as 0.9 and 0.999 for the first and second momentum respectively. On the other hand, the same values of hyperparameters are chosen in the proposed model for binary classification but with different learning rate values for multiple classification.

The codes were written using Python language, version 3.7.7 and Tensorflow 2.0.0 framework was used for training and testing the models. The operations were executed on the CPU of a PC with a base clock speed of 3.9 GHz.

5. RESULTS AND DISCUSSION

5.1. Binary Classification

5.1.1. Transfer Learning Results

The dataset is used to train the transfer learning models mentioned in Section 4.3.1. Number of epochs and batch size is picked from the collections {1, 3, 5} and {32, 64} respectively. The maximum number of epochs is chosen as 5 because the experiments showed that the accuracy doesn't increase if this number is increased. Confusion matrix (Table 4) is extracted to give more detailed examination. Table 5.1 shows the highest prediction accuracy achieved for each model.

Table 5.1. The confusion matrix for binary classification.

		Predicted	
		Calcium Deficiency	**Healthy**
Actual	**Calcium Deficiency**	True Positive	False Negative
	Healthy	False Positive	True Negative

Table 5.2. Detection performances of the transfer learning models.

Model Name	Batch Size	No of Training Epochs	Training Accuracy	Validation Accuracy	Confusion Matrix
VGG16	64	5	0.9372	1.0000	158 3 3 123
VGG19	64	3	0.8974	0.9844	156 5 5 121
MobileNetV2	32	5	1.0000	0.9791	161 0 1 125
EfficientNet-B0	32	3	1.0000	0.9756	160 1 0 126
	64	5	1.0000	0.9817	160 1 0 126
EfficientNet-B2	64	3	0.9688	0.9826	158 3 0 126
InceoptionV3	32	3	0.9688	0.9791	158 3 1 125
	32	5	1.0000	0.9861	158 3 1 125
	64	5	0.9844	0.9861	158 3 1 125
ResNetV2	32	3	0.9688	0.9895	158 3 0 126
	32	5	0.9062	0.9826	158 3 0 126
InceptionRes NetV2	32	5	0.9375	0.9861	156 5 2 124
NASNetMobile	32	1	0.9062	0.9686	157 4 4 122

While the confusion matrices in Table 5 show that MobileNetV2 and EfficientNet-B0 achieved the best prediction accuracy, MobileNetV2 model was the best in detecting calcium deficiency disorder which is considered as the target of the classification. The associated misclassified sample was a single false positive so it is a healthy sample. On the other hand, the associated misclassified sample for EfficientNet-B0 model was a single false negative misclassified sample which is a calcium deficiency sample.

ResNetV2 and EfficientNet-B2 are the other models with good prediction accuracy. Both of them have only three false negative misclassified samples. This means that they are ideal for correctly classifying healthy leaves correctly without mistakes. Table 6 shows samples of misclassified images with its actual class and calculated probability. The calculated probabilities show the models in general are confident about its predictions as the probability of the misclassified images were high. On the other hand, the prediction confidence was lower in some misclassified samples, perhaps due to the small area of the tipburn which is shown in the third sample in Table 5.3.

Table 5.3. Samples of the misclassified images.

Image	Actual Label	Misclassified by	Calculated Probability	
			Healthy	Calcium Deficiency
	Healthy	MobileNetV2	0.1843	0.8157
	Calcium Deficiency	EfficientNetB0	0.8695	0.1305
		EfficientNetB2	0.9082	0.0918
	Calcium Deficiency	ResNetV2	0.9234	0.0766
		EfficientNetB2	0.6258	0.3742
	Calcium Deficiency	ResNetV2	0.9406	0.0594

The results show that using the models with lowest number of parameters (i.e. MobileNetV2 and EfficientNets) was better in terms of test accuracy. On the other hand, the test accuracy was lower in models with highest number of total parameters (i.e. VGGs). These results mean that using models with low number of parameters are efficient and more appropriate when the number of target labels is small like this classification state.

The training time for the models changed with regard to number of epochs, model depth and size of the input image. The InceptionResNetV2 was the most time consuming, the model consumed approximately 6 hours for 5 epochs and batch size of 32. On the other, MobileNetV2 was the fastest to train as the model was originally designed for devices with low computations.

5.1.2. Results Associated with the Proposed Model

The proposed model contains lower number of parameters comparing with the models mentioned in section 5.1.1. which achieved best accuracy. The proposed model is trained from scratch for 50 epochs with batch size of 32 and 64. The epoch number for testing is chosen according to the best accuracy achieved in training and validation. Figure 9 shows training and validation accuracy for batch size of 32 and 64.

Figure 9a shows a fluctuating behavior in validation accuracy with batch size of 32. This means that the batch size does not have sufficient information to evaluate the generalization ability of the model. Therefore, the model was not evaluated according to this batch size. On the other hand, the fluctuations appeared to be more subsided when the batch size increased to 64 where the dataset was represented better. The batch size was not increased further because the number of samples in the dataset is limited.

The model achieved best accuracy at the epochs 45 and 46, so the weights belonging to these epochs are used for testing the model. The results are given in Table 5.4..

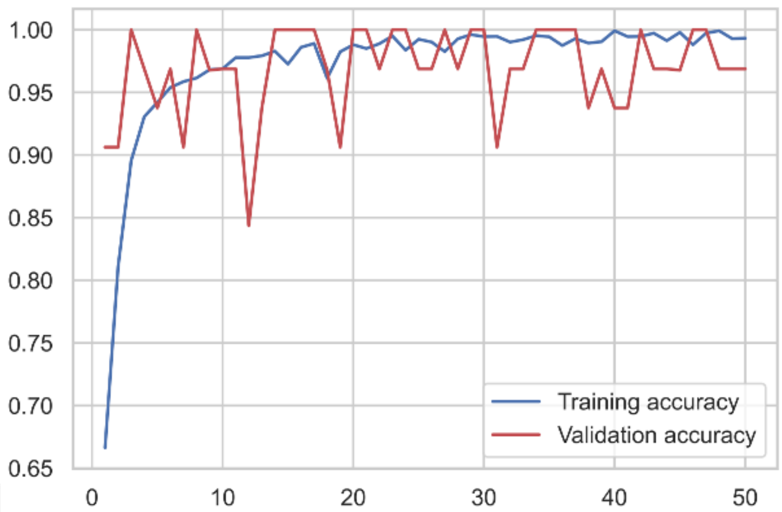
For more performance evaluation of the model, performance metrics such as accuracy, precision, recall, and F1-score were calculated. The related results are given in Table 8 and 9 while the equations used are as follows:

$$\text{Accuracy} = \frac{(True\ Positives + True\ Negatives)}{Total\ Number\ of\ Images} \quad 5.1$$

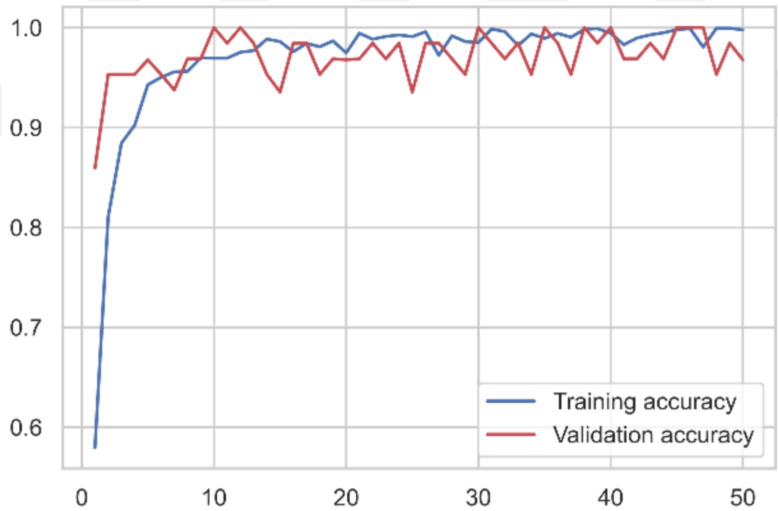
$$\text{Precision} = \frac{(True\ Positives)}{(True\ Positives + False\ Positives)} \quad 5.2$$

$$\text{Precision} = \frac{(True\ Positives)}{(True\ Positives + False\ Positives)} \quad 5.3$$

$$\text{F1 score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad 5.4$$



(a)



(b)

Figure 5.1. Training and validation accuracies for batch sizes of (a) 32 and (b) 64. Horizontal and vertical axes represent the epoch number and accuracy, respectively.

Table 5.4. Detection performance of the proposed model.

Batch Size	Epoch No	Training Accuracy	Validation Accuracy	Confusion Matrix	Prediction accuracy
64	45	0.9979	1.0000	$\begin{array}{c\|c} 160 & 1 \\ \hline 1 & 125 \end{array}$	0.9930
64	46	0.9990	1.0000	$\begin{array}{c\|c} 160 & 2 \\ \hline 0 & 125 \end{array}$	0.9930

Table 5.5. Performance measures of classification for epoch 45.

Precision (%)	Recall (%)	F1-score (%)
99.3	98.7	99

Table 5.6. Performance measures of classification for epoch 46.

Precision (%)	Recall (%)	F1-score (%)
100	98.7	99.3

The proposed model was able to achieve an accuracy of 0.9930. This accuracy is very close to the best accuracy achieved by transfer learning models (i.e. MobileNetV2 and EfficientNetB0). However, the proposed model was unable to detect all leaves with calcium deficiency as MobileNetV2 model did. Moreover, every sample consumed 234ms to be classified.

5.2. Multi-class Classification

5.2.1. Transfer Learning Results

In the multi-class problem, the same steps were followed as in binary classification. The models mentioned in Section 4.3.1 were trained for different number of epoch and batch size. The accuracy of the models did not increase significantly after the fifth epoch while the sets {1, 3, 5} and {32, 64} are used to choose the number of epoch and batch size respectively. Multi-class confusion matrix is generated for these models (Table 10). The best accuracy for each model is reported in Table 5.7.

Table 5.7. The confusion matrix for multi-class problem.

		Predicted		
		Calcium-Deficiency	**Healthy**	**Old-Leaves**
Actual	**Calcium-Deficiency**	P_{CC}	P_{HC}	P_{OC}
	Healthy	P_{CH}	P_{HH}	P_{OH}
	Old-Leaves	P_{CO}	P_{HO}	P_{OO}

Table 5.8. Detection performances of the transfer learning models.

Model Name	Batch Size	No of Training Epochs	Training Accuracy	Validation Accuracy	Confusion Matrix		
VGG16	64	5	0.8430	0.9375	139	17	5
					11	86	8
					1	13	112
VGG19	64	3	0.8098	0.8594	142	11	8
					15	70	20
					2	10	114
MobileNetV2	32	5	0.9688	0.9566	155	5	1
					5	97	3
					0	2	124
EfficientNet B0	32	5	0.9688	0.9490	159	1	1
					1	102	2
					0	3	123
EfficientNet B2	64	5	0.9688	0.9566	155	1	5
					3	100	2
					0	2	124
InceoptionV3	32	5	0.9062	0.9311	157	2	2
					5	96	4
					0	3	123
ResNetV2	32	5	1.0000	0.9286	155	2	4
					2	93	10
					0	3	123
InceptionRes NetV2	64	5	0.7812	0.9158	149	7	5
					5	91	9
					0	3	123
NASNetMobile	32	5	0.9688	0.9056	151	7	3
					8	83	14
					1	11	114

As presented in Table 11, the EfficientNetB0 and EfficientNetB2 models gave the highest prediction accuracy. According to the confusion matrices, there are eight misclassified samples for EfficientNetB0 and 13 misclassified samples for EfficientNetB2. Most of misclassified samples associated with EfficientNetB2 belong to “calcium deficiency” class. However, the number of misclassified images in EfficientNetB0 model associated with calcium deficiency was lower. In addition, EfficientNetB0 model was the best in predicting “old leaves” class comparing with other models including EfficientNetB2.

The results showed that increasing the depth of the model does not help increase the accuracy of prediction, as EfficientNetB2 model is deeper than " EfficientNetB0 " model.

Among the other models with high accuracy values, MobileNetV2 and InceptionV3 achieved 95.91% accuracy while the lowest test accuracy was obtained by the models with the highest number of total parameters (i.e. VGGs). The effect of number of was obvious either in binary or multiple classification, where the models with lower number of parameters gave better results. Some samples of the misclassified images are given in Table 12. The table shows that " EfficientNetB0" model was less confident in predicting the misclassified samples while "EfficientNetB2" model was more confident. The table shows that the correct class between the three classes had the second highest calculated probability for both EfficientNetB0 and EfficientNetB2 models.

Table 5.9. Samples of the misclassified images.

Image	Actual Label	Misclassified by	Calculated Probability		
			Healthy	Calcium Deficiency	Old Leaves
	Old Leaves	EfficientNetB0	0.5146	0.2233	0.2620
	Calcium Deficiency	EfficientNetB0	0.2405	0.3657	0.3936
		EfficientNetB2	0.0287	0.0860	0.8516
	Healthy	EfficientNetB0	0.2971	0.2546	0.4481
		EfficientNetB2	0.2971	0.0084	0.6944
	Old Leaves	EfficientNetB2	0.0350	0.5470	0.4178

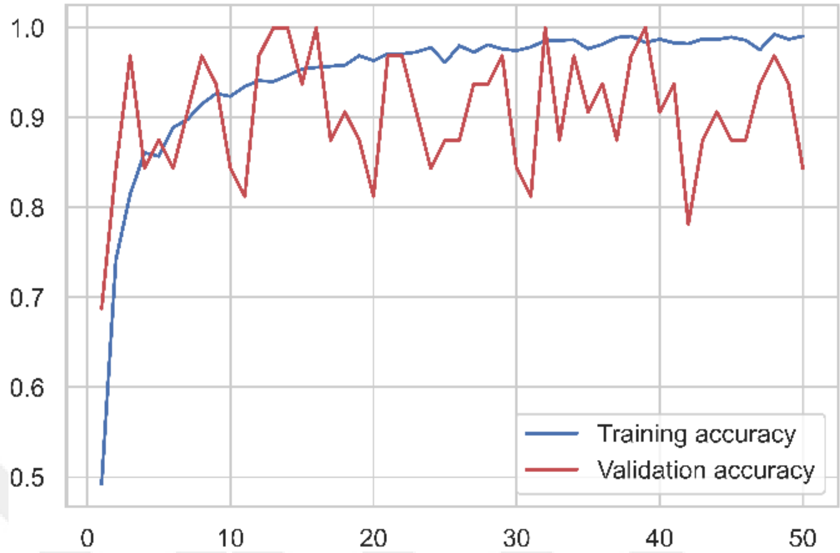
The training time for the models ranged from approximately ten minutes to eight hours. InceptionResNetV2 model was the most time-consuming to train due to its high depth and the input image size. On the other hand, VGG16 was the fastest to train as it has low depth and the input image size is relatively smaller.

5.2.2. Results Associated with The Proposed Model

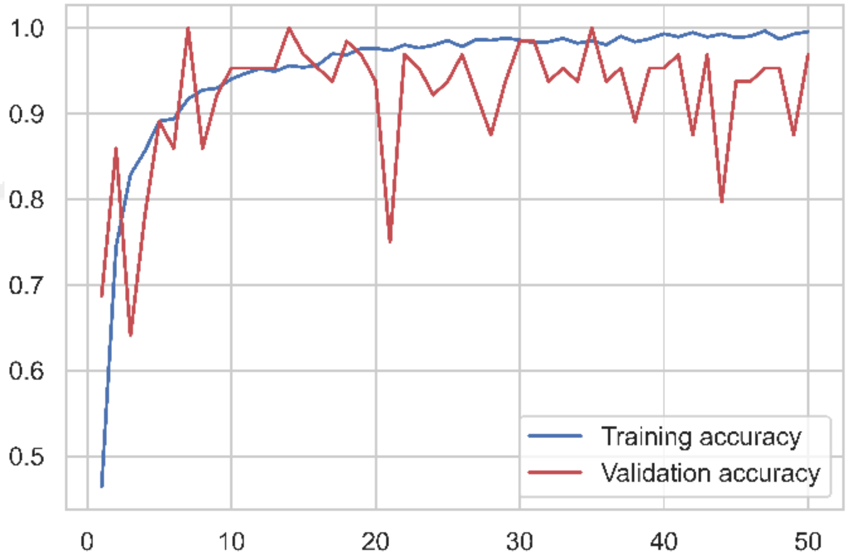
The proposed model that was used in binary classification, was reused for multiple classification. The model is trained from scratch for 50 epochs with batch size of 32 and 64. The plots for training and validation accuracies are reported in

Figure 5.2. The fluctuations in Figure 5.2 b showed to be smaller after increasing the batch size to 64. Due to the limited number of images in the dataset, the batch did not increase further.

The highest prediction accuracy is achieved at the epoch of 48 for batch size of 32 and at the epoch of 43 for batch size of 64. The corresponding results are given in Table 5.10. Performance metrics such as sensitivity, recall, and F1-score are calculated and reported in Table 5.11 and Table 5.12.



(a)



(b)

Figure 5.2. Training and validation accuracies for batch sizes of (a) 32 and (b) 64. Horizontal and vertical axes represent the epoch number and accuracy, respectively.

Table 5.10. Detection performance of the proposed model.

Batch Size	Epoch No	Training Accuracy	Validation Accuracy	Confusion Matrix			Prediction accuracy
32	48	0.9929	0.9688	148	7	6	0.9540
				0	102	3	
				1	1	124	
64	43	0.9894	0.9688	151	6	4	0.9617
				0	101	4	
				0	1	125	

Table 5.11. Performance measures of classification for batch size of 32.

Reference Class	Precision (%)	Recall (%)	F1-score (%)
Calcium deficiency	99.3	91.9	95.4
Healthy	93.9	98.4	96
Old leaves	92.7	97.1	94.8
Average	95.3	95.8	95.4

Table 5.12. Performance measures of classification for batch size of 64.

Reference Class	Precision (%)	Recall (%)	F1-score (%)
Calcium deficiency	100	93.7	96.7
Healthy	93.9	99.2	96.4
Old leaves	93.5	96.1	94.7
Average	95.8	96.3	95.9

The proposed model achieved accuracy level of 0.9540 and 0.9617 for epochs of {48,43} and batch size of {32,64} respectively. These results are very close to those achieved by InceoptionV3, MobileNetV2 and EfficientNetB2 models. However, the proposed model has a weakness in detecting the leaves with calcium deficiency correctly. The time consumed to classify a single image was approximately 224ms.

5.2.3. Learning Rate Effect

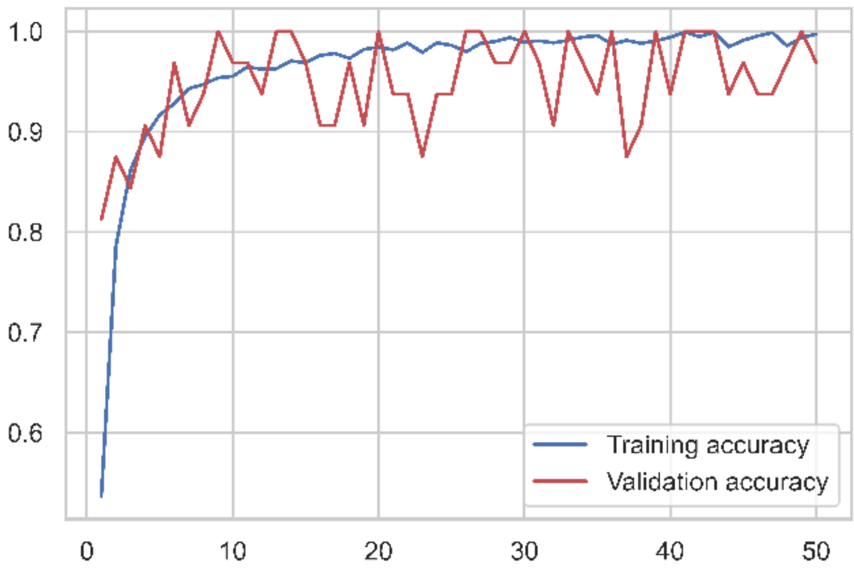
Many learning rates are used to analyze its effect on performance of the proposed model. The experiment showed that using a high value of learning rate (i.e. 0.01 or higher) caused hanging in the model with a high error value without any progress during training process. The same problem was encountered when a very low learning rate values were used (i.e. 0.00001 or lower). Therefore, 0.0005 and 0.0001 values were chosen as the learning rate values with which the experiments are repeated. The weights belonging to the best validation and training accuracies are used for testing the proposed model. The corresponding results are given in Table 5.13.

Table 5.13. Detection performance of the proposed model with tuning the learning rate.

Batch Size	Epoch No	Learning Rate	Training Accuracy	Validation Accuracy	Confusion Matrix		
32	42	0.0005	0.9946	1.0000	161	0	0
					0	102	3
					0	1	125
32	43	0.0001	0.9908	1.0000	149	3	9
					0	100	5
					0	1	125
64	41	0.0005	0.9956	1.0000	158	1	2
					0	101	4
					1	3	122
64	43	0.0001	0.9790	1.0000	157	1	3
					1	101	3
					0	2	124

Table 5.13 shows how changing the learning rate value was effective with regard to increase the accuracy of the proposed model. The proposed model was able to achieve an accuracy level of 98.97% after tuning the learning rate at 0.0005 with batch size of 32. This result is the best compared with the results that are obtained from the transfer learning models or the proposed model with default learning rate (0.001). By checking the rest of the results, the proposed model was able to achieve an accuracy level of 97.44% with batch size of 64 after tuning the learning rate to 0.0001. This result exceeds the best result achieved by the proposed model when default learning rate was used with batch size of 64.

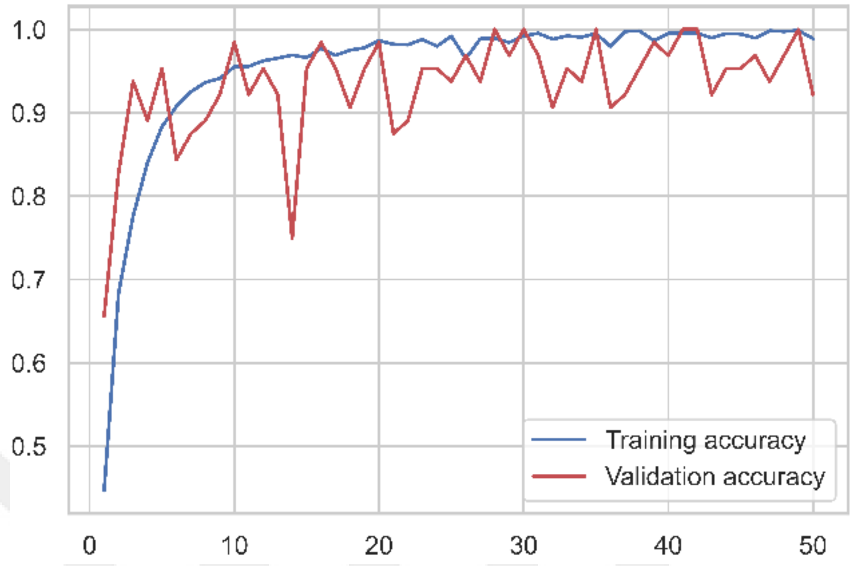
Figure 5.3. shows training and validation accuracy with batch size of 32 and learning rate of (a) 0.0005 and (b) 0.0001. Figure 5.3. shows training and validation accuracy for batch size of 64 and learning rate of (a) 0.0005 and (b) 0.0001.



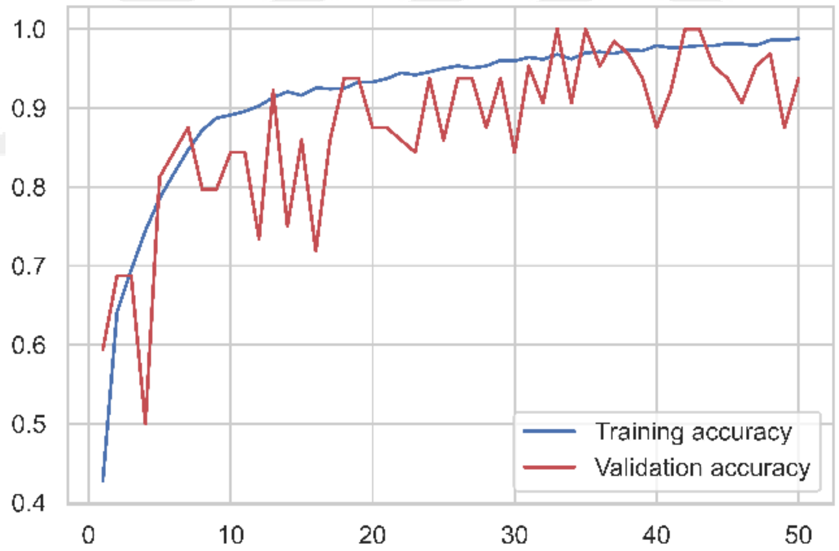
(a)

(b)

Figure 5.3. Training and validation accuracies for batch size of 32 and learning rate of (a) 0.0005 and (b) 0.0001.



(a)



(b)

Figure 5.4. Training and validation accuracies for batch size of 64 and learning rate of (a) 0.0005 and (b) 0.0001.

6. CONCLUSION

Tipburn of strawberry is one of the disorders that results from a local deficiency of calcium in the plant causing a decrease in photosynthetic activity. Lack of chlorophyll is also a factor that affects strawberries and results from aging of the leaf.

In this thesis, a CNN model is proposed to detect calcium deficiency disorder and physiological change connected with lack of chlorophyll in strawberry leaves. In addition, famous transfer learning models are trained to compare its results with the obtained results from the proposed model. The applied transfer learning models are VGGs, MobileNetV2, EfficientNet (B1 and B2), InceptionV3, ResNetV2, InceptionResNetV2, and NASNetMobile. A dataset that contains 626, 524 and 805 images of “healthy”, “old leaves” and “calcium deficiency” leaves, respectively, was used. The images were taken in real conditions and there were not any preprocessing steps that affect the classification.

The classification was done in two stages, a binary classification in which healthy leaves and leaves with calcium deficiency were used, then a multi-class classification after adding the old leaves. This approach was used to get more details about the performance of the proposed model.

According to the results, MobileNetV2 and EfficientNetB0 achieved the highest prediction accuracy in binary classification. The misclassified samples included one false positive for MobileNetV2 and one false negative for EfficientNetB0. The proposed model achieved a similar accuracy with one additional misclassified sample. MobileNetV2 was the only model which was able to classify all the leaves with calcium deficiency. On the other hand, EfficientNetB0 achieved the highest prediction accuracy in the multiple classification with 8 misclassified samples out of 392 samples in the test set. In addition, the proposed model has shown to be able to classify most of the healthy and old leaves using Adam optimizer and default learning rate but, was less efficient in classifying leaves with calcium deficiency. However, this problem was solved after changing learning rate to 0.0005 where the model was able to classify all leaves with calcium deficiency in the multiple classification.

The batch size in both stages of classification was increased from 32 to 64, as a smaller batch sizes proved they do not represent dataset adequately. This led in a notable decrease in the undesirable fluctuations in the validation set. However, increasing the number of samples in the validation set should be considered in future work as it may eventually result in achieving a more regular learning curve and a higher accuracy level. In addition, updating the dataset is expected by adding more images and more types of disorders, diseases and physiological changes in strawberries.

This work showed that using deep architectures or models with large number of parameters are unnecessary for this classification task. A better performance can be achieved with a smaller sequential model with lower parameters. To accomplish this achievement, learning rate is a very important hyper-parameter and should be tuned carefully. The experiments in this work showed that using lower value of learning rate was effective to increase the performance of the model.

REFERENCES

Abdullahi, H. S., et al. (2017). Convolution neural network in precision agriculture for plant image recognition and classification. 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Ieee.

Amara, J., et al. (2017). "A deep learning-based approach for banana leaf diseases classification." Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband.

Bárcena, A., et al. (2019). "Shade cloths and polyethylene covers have opposite effects on tipburn development in greenhouse grown lettuce." Scientia Horticulturae **249**: 93-99.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. Neural networks: Tricks of the trade, Springer**:** 437-478.

Bhagat, M., et al. (2020). Bell Pepper Leaf Disease Classification Using CNN. 2nd International Conference on Data, Engineering and Applications (IDEA), IEEE.

Chollet, F. (2018). Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek, MITP-Verlags GmbH & Co. KG.

Durmuş, H., et al. (2017). Disease detection on the leaves of the tomato plants by using deep learning. 2017 6th International Conference on Agro-Geoinformatics, IEEE.

Dyrmann, M., et al. (2016). "Plant species classification using deep convolutional neural network." Biosystems Engineering **151**: 72-80.

Ferentinos, K. P. (2018). "Deep learning models for plant disease detection and diagnosis." Computers and Electronics in Agriculture **145**: 311-318.

Genaev, M., et al. (2020). Application of neural networks to image recognition of wheat rust diseases. 2020 Cognitive Sciences, Genomics and Bioinformatics (CSGB), IEEE.

Ghosal, S. and K. Sarkar (2020). Rice Leaf Diseases Classification Using CNN With Transfer Learning. 2020 IEEE Calcutta Conference (CALCON), IEEE.

Goodfellow, I., et al. (2016). Deep learning, MIT press Cambridge.

Hari, S. S., et al. (2019). Detection of Plant Disease by Leaf Image Using Convolutional Neural Network. 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), IEEE.

He, K., et al. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

Heaton, J. (2018). Ian goodfellow, yoshua bengio, and aaron courville: Deep learning, Springer.

Hidayatuloh, A., et al. (2018). Identification of Tomato Plant Diseases by Leaf Image Using Squeezenet Model. 2018 International Conference on Information Technology Systems and Innovation (ICITSI), IEEE.

Jasim, M. A. and J. M. AL-Tuwaijari (2020). Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques. 2020 International Conference on Computer Science and Software Engineering (CSASE), IEEE.

Kashyap, Y. and S. S. Shrivastava (2020). Assessment of Soybean Leaf foliar Diseases using CNN & Weka tool. 2nd International Conference on Data, Engineering and Applications (IDEA), IEEE.

Kaya, A., et al. (2019). "Analysis of transfer learning for deep neural network based plant classification models." Computers and Electronics in Agriculture **158**: 20-29.

Khan, S. and M. Narvekar (2020). Disorder Detection in Tomato Plant Using Deep Learning. Advanced Computing Technologies and Applications, Springer**:** 187-197.

Khan, S. and M. Narvekar (2020). Disorder detection of tomato plant (solanum lycopersicum) using IoT and machine learning. Journal of Physics: Conference Series, IOP Publishing.

Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

Kuronuma, T., et al. (2018). "Tipburn severity and calcium distribution in lisianthus (Eustoma Grandiflorum (Raf.) Shinn.) cultivars under different relative air humidity conditions." Agronomy **8**(10): 218.

Liu, B., et al. (2018). "Identification of apple leaf diseases based on deep convolutional neural networks." Symmetry **10**(1): 11.

Lu, J., et al. (2017). "An in-field automatic wheat disease diagnosis system." Computers and Electronics in Agriculture **142**: 369-379.

Madhulatha, G. and O. Ramadevi (2020). Recognition of Plant Diseases using Convolutional Neural Network. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE.

Mikołajczyk, A. and M. Grochowski (2018). Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW), IEEE.

Mukti, I. Z. and D. Biswas (2019). Transfer Learning Based Plant Diseases Detection Using ResNet50. 2019 4th International Conference on Electrical Information and Communication Technology (EICT), IEEE.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective, MIT press.

Olle, M. and I. H. Williams (2017). "Physiological disorders in tomato and some methods to avoid them." The Journal of Horticultural Science and Biotechnology **92**(3): 223-230.

Park, H., et al. (2018). Crops disease diagnosing using image-based deep learning mechanism. 2018 International Conference on Computing and Network Communications (CoCoNet), IEEE.

Patterson, J. and A. Gibson (2017). Deep learning: A practitioner's approach, " O'Reilly Media, Inc.".

Picon, A., et al. (2019). "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild." Computers and Electronics in Agriculture **161**: 280-290.

Prasanna Mohanty, S., et al. (2016). "Using Deep Learning for Image-Based Plant Disease Detection." arXiv: arXiv: 1604.03169.

Qian, N. (1999). "On the momentum term in gradient descent learning algorithms." Neural networks **12**(1): 145-151.

Rahnemoonfar, M. and C. Sheppard (2017). "Deep count: fruit counting based on deep simulated learning." Sensors **17**(4): 905.

Ramcharan, A., et al. (2017). "Deep learning for image-based cassava disease detection." Frontiers in plant science **8**: 1852.

Russakovsky, O., et al. (2015). "Imagenet large scale visual recognition challenge." International journal of computer vision **115**(3): 211-252.

Sandler, M., et al. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition.

Sardogan, M., et al. (2018). Plant leaf disease detection and classification based on CNN with LVQ algorithm. 2018 3rd International Conference on Computer Science and Engineering (UBMK), IEEE.

Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

Singh, R. and S. Athisayamani (2020). "Banana leaf diseased image classification using novel HEAP auto encoder (HAE) deep learning." Multimedia Tools and Applications **79**(41): 30601-30613.

Singh, U. P., et al. (2019). "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease." IEEE Access **7**: 43721-43729.

Srivastava, N., et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research **15**(1): 1929-1958.

Szegedy, C., et al. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI Conference on Artificial Intelligence.

Szegedy, C., et al. (2016). Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition.

Tan, M. and Q. V. Le (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946.

Tieleman, T. and G. Hinton (2012). "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural networks for machine learning **4**(2): 26-31.

Ullah, M. R., et al. (2019). Plant Diseases Recognition Using Machine Learning. 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), IEEE.

Zoph, B. and Q. V. Le (2016). "Neural architecture search with reinforcement learning." arXiv preprint arXiv:1611.01578.

CURRICULUM VITAE

Muhab HARIRI was born on October 22th, 1992 in Aleppo/Syria. He has completed his elementary education at Alibaa Al Arabi school, secondary and high education at Al-Bassel school in Aleppo. He has completed undergraduate education at the department of Electrical and Electronics Engineering of Çukurova University in 2018.