DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A DEEP LEARNING APPROACH FOR THE PREDICTION OF DISEASES IN COTTON CULTIVATION

by

Burak KAYA

September, 2021

İZMİR

A DEEP LEARNING APPROACH FOR THE PREDICTION OF DISEASES IN COTTON CULTIVATION

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science, Computer Science Program

by

Burak KAYA

September, 2021

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A DEEP LEARNING APPROACH FOR THE PREDICTION OF DISEASES IN COTTON CULTIVATION"** completed by **BURAK KAYA** under supervision of **PROF.DR. EFENDİ NASİBOĞLU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

....

Prof. Dr. Efendi NASİBOĞLU

Supervisor

....

Prof. Dr. Yılmaz KILIÇASLAN

(Jury Member)

....

Asst. Prof. Dr. K. Ulaş BİRANT

(Jury Member)

Prof. Dr. Özgür ÖZÇELİK

Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First of all, I wish to thank my advisor Prof. Dr. Efendi NASİBOĞLU for his guidance, endless support, advices and understanding approach during thesis study.

I would like to thank Asst. Prof. Dr. Mahmut SİNECEN for his support, trust and suggestions since the beginning of the thesis.

I would like to thank Res. Asst. Önem YILDIZ for her valuable comments and support during my thesis.

Finally, I want to thank my one and only parents, Gülser and Sami KAYA, my brothers, Osman B. KAYA and M. Saadetdin KAYA for their faith, backing, efforts and support all the way through my entire life.

Burak KAYA

A DEEP LEARNING APPROACH FOR THE PREDICTION OF DISEASES IN COTTON CULTIVATION

ABSTRACT

In this thesis, a study on the detection and prediction of cotton diseases, which is a sub-title of environmental factors that are effective in the cultivation of cotton plants, with the help of image processing and deep learning methods is presented. In the first stage, the images of the cotton plant were preprocessed in order to minimize the problems that may be encountered during the application of the preferred deep learning methods. These data obtained as a result of the preprocessing were used as input data for the optimization of the applied deep learning models. With the help of this input data, the hyper-parameters of Convolutional Neural Networks, Long Short-Term Memory Networks and Convolutional Long Short-Term Memory Networks models are decided. In the last phase, the success rates of the predictions made on random images given as input to these optimized models were evaluated. The results obtained as a result of the study were analyzed and compared with the studies in the literature.

Keywords: Convolutional neural networks, convolutional long short-term memory, deep learning, image processing, long short-term memory

PAMUK YETİŞTİRİCİLİĞİNDEKİ HASTALIKLARIN DERİN ÖĞRENME YAKLAŞIMI İLE TAHMİN EDİLMESİ

ÖZ

Bu tezde, pamuk bitkisinin yetiştirilmesinde etkili olan çevresel faktörlerin bir alt başlığı olan pamuk hastalıklarının görüntü işleme ve derin öğrenme yöntemleri yardımıyla tespiti ve tahminlenmesi üzerine yapılan bir çalışma sunulmuştur. İlk aşamada, pamuk bitkisine ait görüntüler, tercih edilen derin öğrenme yöntemlerinin uygulanması esnasında karşılaşılabilecek sorunları minimize etmek adına ön işleme tabi tutulmuştur. Ön işlem sonucu elde edilen bu veriler, uygulanan derin öğrenme modellerinin optimizasyonu için girdi verileri olarak kullanılmıştır. Bu girdi verileri yardımıyla, Evrişimli Sinir Ağları, Uzun Kısa Vadeli Hafıza Ağları ve Evrişimli Uzun-Kısa Vadeli Hafıza Ağları modellerinin hiper-parametrelerine karar verilmiştir. Son aşamada ise, optimize edilen bu modellere girdi olarak verilen rastgele görüntüler üzerinden yapılan tahminlemelerin başarım oranları değerlendirilmiştir. Çalışma neticesinde elde edilen sonuçlar, analiz edilmiş ve literatürdeki çalışmalar ile karşılaştırılmıştır.

Anahtar kelimeler: Derin öğrenme, evrişimli sinir ağlar, evrişimli uzun kısa vadeli hafıza, görüntü işleme, uzun kısa vadeli hafıza

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER 1

INTRODUCTION

1.1 The Problem Statement

The cotton plant has a great importance for humans with its widespread and compulsory use. It has even greater importance for countries that cultivate cotton because of the added value and employment opportunities it creates. With increasing population and increasing interest in natural fiber and rising living standards the demand for cotton plants also increases. Despite that, since the ecology of a limited number of countries allows cotton farming, approximately the 80% of the world production is carried out by eight countries, including Turkey (Gençer, Özüdoğru, Kaynak, Yılmaz, & Ören, 2005).

There are many factors that negatively affect cotton cultivation/farming and among these factors are diseases. The most frequently seen cotton diseases are Seedling Root Rot Disease (Erdoğan, Bölek, & Göre, 2016), Verticillium and Fusarium Wilt Disease (Katsantonis, Hillocks, & Gowen, 2003) and Angular Leaf Spot Disease (Rezene, Tesfaye, Clare, & Gepts, 2018).

Artificial Neural Network (ANN) applications in the textile industry have become more and more popular since 1990 (ÇörekciOğlu, Ercan, & Aras EliBüyük, 2021). Some of the applications are used to find faults of the product and some (Abdel-Hamied, 2021) are used to detect diseases in order to take an action beforehand.

Since the detection of cotton diseases is important and there are newer methods that can be used other than ANNs such as deep learning methods, it is likely that these methods can be more efficient for detecting or even predicting these diseases.

1.2 Literature Review

Agriculture has an irreplaceable place in human life as it has always been. Many countries worldwide depend on the products they obtain from the soil. Since the beginning of the agricultural life, the factors that affect plant production, especially environmental factors, have been tried to be controlled. The environmental factors affecting plant production can be can be divided in four groups:

- *Live factors*, which are herbicides, diseases and harmful insects and helpful organisms,
- *Climate factors*, which are temperature, light, rain/snow, humidity and wind,
- *Soil factors*, which are the fabric of the soil, soil texture, soil salinity and organic substances,
- *Geographical and topographical factors*.

Plant diseases are one of the most important factors, as they affect the product by seriously reducing quality and quantity. It is therefore very important to detect and diagnose these diseases early. Many studies have been done and many methods have been used through time to this end.

As deep learning methods are becoming more and more popular, they have also been used in order to find out methods of detecting and classifying plant diseases. In Figure 1.1, the basic steps of plant disease detection and identification is shown.

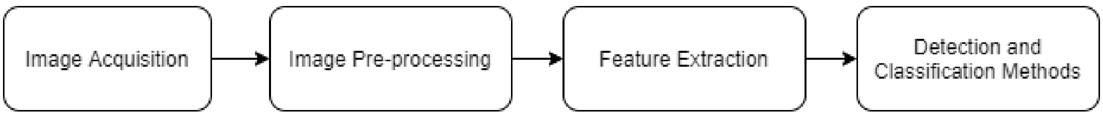

Figure 1.1 Basic steps of plant disease detection and identification

As shown in Figure 1.1, images are obtained using digital devices. In this step both healthy and infected plant images need to be captured. Then, a preprocess phase is applied where methods such as enhancement, segmentation, filtering etc. are used. In the next step, features such as shape, color, texture are obtained. Finally a classification method is used.

There are many studies in the literature where deep learning methods are used in agriculture, especially detecting diseases of plants (Al Hiary, Bani Ahmad, Reyalat, Braik, & ALRahamneh, 2011; Fuentes, Yoon, Kim, & Park, 2017; Mohanty, Hughes, & Salathé, 2016; Sladojevic, Arsenovic, Anderla, Culibrk, & Stefanovic, 2016; G. Wang, Sun, & Wang, 2017).

For cotton plant, support vector machine (SVM) which is a machine learning method is widely used (Bhimte & Thool, 2018; Patil & Zambre, 2014). Mostly using the images of cotton leaves, there are many published studies that focus on cotton other than SVM (Jamuna et al., 2010; H. Li et al., 2011; Revathi & Hemalatha, 2005, 2012; Rothe & Kshirsagar, 2015; Sarangdhar & Pawar, 2017; Yang et al., 2015).

1.3 Goal and Contributions

The goal of this study is to find out which deep learning method can be more useful in order to detect and predict the diseases that can be visually seen on cotton plant. To accomplish this goal, initially, cotton images are used to train deep learning algorithms to be used in this study. Several independent trainings have been processed in order to obtain different models and the test results have been evaluated. These models have been found to be faster and have lower error values than traditional techniques.

The contributions of this study are the following:

- Rather new to the cotton studies technologies are implemented to detect and predict the diseases and found the relationship between them.
- Developed some new models to be faster, easy to use while having rather high accuracy.

1.4 Organization

Organization for the rest of the thesis is as follows:

- The Second Chapter, “Cotton Diseases”, contains information about cotton diseases.
- The Third Chapter, “Image Processing”, gives information about image processing techniques.
- The Fourth Chapter, “Artificial Intelligence”, contains a summarized literature of machine learning and deep learning.
- The Fifth Chapter, “Materials and Methodology”, is where materials and methods used and implemented are explained.
- The Sixth Chapter, “Results and Future Studies”, is the chapter where the results are evaluated and compared to the present studies and it gives information about the future levels of this study.

CHAPTER 2

COTTON DISEASES

2.1 Diseases of Cotton

Diseases are among the most important biotic factors affecting cotton cultivation. They are considered to be the key reason for productivity decrease in cotton industry (Prajapati et al., 2016). It is said that there are great losses in both the quality and quantity of cotton cultivation because of these diseases.

Cotton diseases are mostly bacterial, fungal and viral diseases. They do not only appear on leaves but also on plant seedlings, cotyledons, root etc. Below are some cotton diseases:

- Angular Leaf Spot (Bacterial Blight),
- Magnesium Deficiency,
- Gray Mildew,
- Seedling Root Rot,
- Verticillium and Fusarium Wilt.

2.2 Angular Leaf Spot

Angular Leaf Spot also known as Bacterial Blight is caused by a bacteria called Xanthomonas campestris pv. Malvacearum. This disease occurs in angular shape with the color range between red and brown (Bhimte & Thool, 2018). An image of a leaf infected by this bacteria is shown in Figure 2.1.

Figure 2.1 Angular leaf spot disease (Bhimte & Thool, 2018)

2.3 Magnesium Deficiency

Magnesium deficiency usually occurs in strongly acidic, light and sandy soils where magnesium gets easily leached away (Bhimte & Thool, 2018). It starts at older leaves and turns their green color into purple. An image of a leaf infected by this disease is shown in Figure 2.2.

Figure 2.2 Magnesium deficiency leaf spot disease (Bhimte & Thool, 2018)

2.4 Gray Mildew

An organism named Ramularia areola Atk. causes this disease on cotton. This parasite usually attacks the older leaves. Unlike angular leaf spot disease, this disease is not necessarily seen in angular spots. An image of a leaf infected by this disease is shown in Figure 2.3.

Figure 2.3 Gray mildew disease (Bhimte & Thool, 2018)

2.5 Seedling Root Rot

Seedling root rot in cotton is caused by one or more of the fungi Rhizoctonia Solani, Fusarium, Verticillium, Pythium. These fungi, which spend the winter in the soil, germinate cotton seeds after planting, causing the seedlings to die before or after emergence (Baheti et al., 2016).

2.6 Verticillium and Fusarium Wilt

In verticillium and fusarium wilt, the leaves turn yellow and then dry. When planting is done late or when the disease is seen early, plant height decreases, the number of bolls per plant decreases and the bolls remain small. In Figure 2.4, a leaf affected with verticillium and fusarium wilt can be seen.

Figure 2.4 Verticillium and fusarium wilt

CHAPTER 3

IMAGE PROCESSING

3.1 Introduction

Many animal species receive information about their environment without being contingent on their eyes (McAndrew, 2004; Russ, 2006; Woods, Eddins, & Gonzalez, 2009). Unlike these animals, humans still rely on the ability of seeing and analyzing the objects around them.

Digital image processing was developed because of the need of translation between the human visualization and digital imaging decks. Digital image processing basically uses an algorithm to process digital images (Gonzalez & Woods, 2018).

3.2 Digital Image Processing

Vision is one of the most improved senses of a human, so it is safe to say that images take one of the most important roles in human consciousness. Unlike humans, imaging tools can work on the entire EM spectrum (Gonzalez & Woods, 2018).

The smallest element of a digital image is named a pixel. A digital image consists of multiple pixels with gray levels or density.

Today, many fields of study involve digital image processing with varying differences in terms of applied steps. In Figure 3.1, the processes applied to an image are shown.

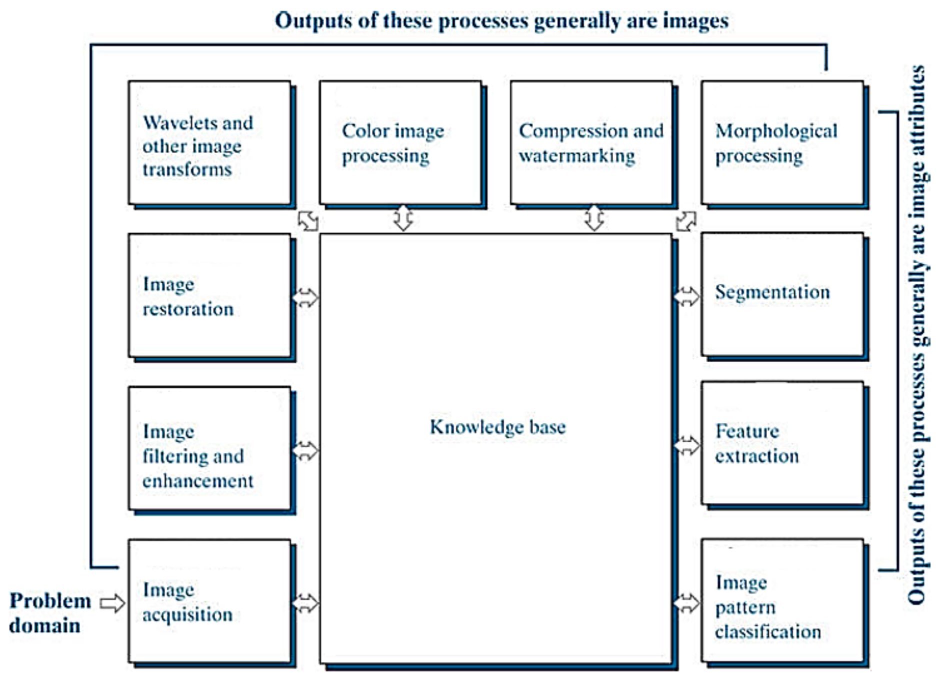

Figure 3.1 Fundamental steps in digital image processing (Gonzalez & Woods, 2018)

Steps of digital image processing can be described as follows:

- Image Acquisition: The image is captured by a visual tool (eg. camera). If needed, the image is digitized using an analogue-to-digital convertor.
- Image Filtering and Enhancement: This is the step where the image is manipulated so that it can be used with higher efficiency in specific applications. The main reason to use enhancement techniques is to highlight the details that are hidden or make a specific feature more noticeable.
- Image Restoration: Restoring an image in order to have an improved quality on the appearance of an image.
- Color Image Processing: This is the step which the color of the image is used for extracting more important features of the image for specific applications.

- Wavelets: They basically are the grounds of showing the image with different degrees of resolution. This process is useful for image data compression.
- Compression: Compression is used to decrease the size of storage needed to save an image.
- Morphological Processing: There are components that have a use in representation and description of shape. Morphological processing is where one uses tools to extract these components from the image.
- Image Segmentation: Segmentation is applied in order to divide an image into parts.
- Feature Extraction: Detection and description of the features are contained in the feature extraction process.
- Image Pattern Classification: It is the step where labeling is done for an object in the image depending on its description of features.

3.3 Computer Vision

Computer vision (CV) is a field where computers are used to gather information and have a high level understanding of the objects in the environment with no physical interaction and through optical devices such as cameras, photograph machines, etc. Computer vision involve artificial intelligence, image processing and pattern recognition techniques as well as different kind of classifiers (Forsyth & Ponce, 2011; Jain & Kasturi, 2011).

CHAPTER 4

ARTIFICIAL INTELLIGENCE

4.1 Introduction

The term "artificial intelligence" (AI) is coined by John McCarthy in 1955 who defines AI as creating human-like machines (Hamet & Tremblay, 2017). The field of AI research in computer science can be defined as the group of studies of intelligent agents; devices that recognize the environment and take actions that are more suitable to achieve a certain goal. In other words, AI means a machine copycatting perceptive behaviors of humans (Ongsulee, 2017).

The main goals of AI studies are knowledge representation, learning, planning and communication. AI approaches are mostly based on statistical methods and computational intelligence. Therefore, AI concerns many fields such as mathematics, psychology, linguistics, neuroscience, computer science.

In 1950, Alan M. Turing published his work proposing the idea of possibility of creating machines that can think (Turing, 1950). Walter Pitts and Warren McCulloch were the first ones to describe "neural network" (McCulloch & Pitts, 1943). These studies later created a new field named Machine Learning (ML).

4.2 Machine Learning

Machine Learning is a field of study where algorithms automatically learn to decide the action they need to take by the use of data and through the experience they gain. The term was coined by Arthur L. Samuel in 1959 (Samuel, 1959). In 1960, Nils J.

Nilsson has published a book named "Learning Machines" which mostly dealt with pattern classification using ML.

In machine learning, algorithms are trained using statistical methods in order to make classifications or predictions and point out some key information in data mining. The insights that are obtained stimulate the decision making process afterwards.

A standard machine learning algorithm consists of the components below:

- Decision: A group of calculations that take the data, process it and return the predicted value the algorithm is seeking to decide.
- Error Function: It is used to measure how accurate the predicted value when it is compared to the expected value.
- Optimization: It means the algorithm calculating the miss in between the predicted and expected values and updating the network depending on the decision was taken.

The analytical accuracy is expected to improve every iteration it teaches itself from the data it analyzes since the machine learning algorithm updates itself autonomously. This aspect of learning is very important because it is realized with no human involvement and it helps uncovering some insights that are not easy so obtain even when it is not specifically expected to do so. In Figure 4.1, the working principle of a learning process method is shown.

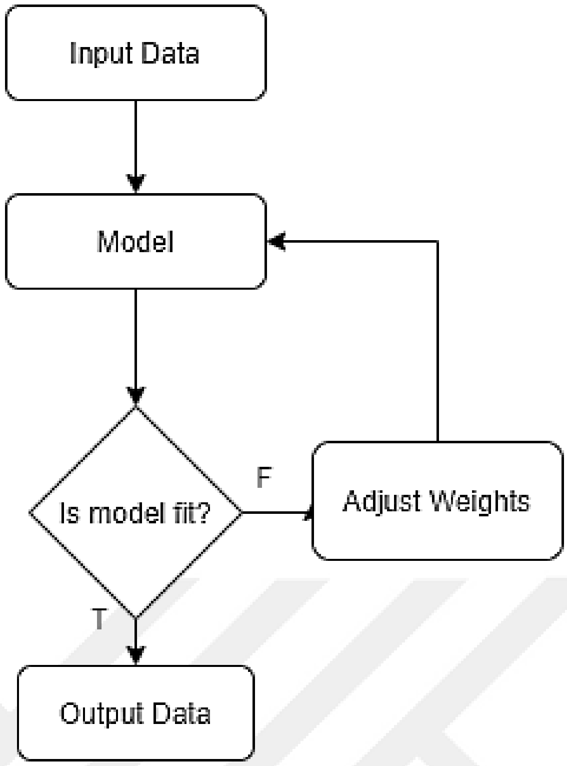

Figure 4.1 Learning process

When entering the machine learning world for the first time, it usually comes very complex and hard to understand. It need lots of effort to fully understand everything that happens during the learning process. Some of the most important terms to understand the idea of learning process in machine learning can be listed as follows:

- Activation Function: It is used to create a nonlinear transformation of input data. It is usually made of weights and biases where weight defines the influence of input to output for that connection and bias can be thought as the constant value. Some of the popular activation functions are rectified linear units, tanh and sigmoid functions.
- Backpropagation: It is the framework used to adjust weights to minimize the loss function.
- Batch Normalization: Normalizing values between zero and one in order to help train the network faster.

- Cost Function: It is the difference between the expected and calculated output values. It is one of the keystones to learning.
- Dropout: It is a regularization technique to prevent overfitting in deep learning. In dropout, some of the nodes and their connections are removed generally randomly.
- Epoch: Definition of single cycle forward and backward pass through the training dataset for every example.
- Forward Propagation: Defines a single forward pass in neural networks.
- Iteration: Total number of forward and backward passes in a neural network.
- Learning Rate: It is the amount that the weights are updated during the training. A value is chosen between zero and one.
- Learning Rate Decay: The concept that allows adjusting the learning rate during the training.
- Optimizer: They are defined as algorithms used to increase or decrease the effects of the attributes of ANN. Most commonly used optimization algorithms are AdaDelta, AdaGrad, Adam, Gradient Descent, Stochastic Gradient Descent and Minibatch Gradient Descent. Advantages and disadvantages of these algorithms are given in Table 4.1.

Table 4.1 Advantages and disadvantages of optimizers

Optimizer	Advantages	Disadvantages
AdaDelta	No need to set a default learning rate	Expensive on computation
AdaGrad	Able train on sparse data Do not need to tune learning rate manually	Slow training
Adam	Too fast, converges rapidly Rectifies vanishing learning rate	Expensive on computation
Gradient Descent	Easy to implement and compute	May trap at local minima
Stochastic Gradient Descent	Converges in less time	Can keep going after achieving local minima
Minibach Gradient Descent	Less variance	May trap at local minima

ML models are usually classified by the need of human intervention on raw data or whether a reward is offered. Machine learning approaches are grouped with respect to these criteria as follows:

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

4.2.1 Supervised Learning

Supervised learning is a subcategory of ML. It is one of the most important methods in machine learning since a high number of supervised learning techniques are used widely. Supervised learning is technically defined as being able to learn from annotated training data (Cunningham, Cord, & Delany, 2008). As it is understood from the name itself, this learning method is built on the idea of having a supervisor that gives the instructions to the learning method on the labels to associate with training data. In this aspect, it is fair to say that supervised learning algorithms induce a model from training data which are labelled to classify other unlabeled data or predict an outcome accurately.

Supervised learning methods use a training dataset to train models in order to obtain the expected value in output. This training set contains input data and real output data that make learning possible for model over time and through experience. Then the model calculates its accuracy using a loss function and adjusts itself by optimizing weights until either the error has been minimized enough or a stop condition is hit. Supervised learning handles two types of problems and those problems are as follows:

- Classification: The algorithm tries to label test data into specific categories with highest accuracy possible. When it gets the labels within the training data, it tries to obtain some information on how to test data is should be defined or labeled or classified. Most commonly used classification algorithms can be listed as support vector machines (SVM), decision trees, naive bayes, artificial neural networks and k-nearest neighbors.
- Regression: It is used to have an insight and information about independent and dependent variables. Mostly preferred regression algorithms for studies can be listed as linear regression, logistic regression and polynomial regression.

4.2.1.1 Linear Regression

All linear regression models contain a dependent variable and one or more independent variable(s). These models are chosen to predict about future outcomes (Montgomery, Peck, & Vining, 2021). Basically, if a model's function is considered as given in Equation 4.1, the linear regression model is the function given in Equation 4.2 where β describes the slope, α describes the intercept and ε describes the error.

$$y = \alpha + \beta x_i \tag{4.1}$$

$$y_i = \alpha + \beta x_i + \varepsilon_i \tag{4.2}$$

4.2.1.2 Logistic Regression

Linear regression is being chosen with continuous dependent variables while logistic regression is chosen when dealing with categorical dependent variables which means the outputs are binary, in other words the outputs are such as true/false or yes/no.

4.2.1.3 Support Vector Machine

Support vector machines are used for both data classification and regression (Suykens & Vandewalle, 1999). With that said, it is mostly used to solve classification problems. It finds the maximum distance between two separate classes and it constructs a hyperplane which also known as the decision boundary, separates the classes on both of the plane.

4.2.1.4 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) is considered as a simple method. Its effectiveness comes from it being a non-parametric classification method (Hand, 2007). KNN keeps all training data for classification unlike some other supervised learning models. It is easy to use and have rather low calculation time but as the size of the test data grows, the time needed to finish the process also grows which makes it less appealing for classification studies.

4.2.1.5 Artificial Neural Networks

Artificial neural networks (ANN) is a subcategory of ML and considered to be the keystone of deep learning algorithms. ANN is considered as a very basic level of nonlinear learning that mimics the neuron cluster that can be encountered in real life. As it is shown in Figure 4.2, a real neuron uses dendrites to gather inputs from other neurons and combines the data information, generating a nonlinear response when the threshold is reached, which it sends to other neurons using the axon (Larose & Larose, 2014). It is seen that artificial neuron has the same process. It collects the inputs (x) from the neurons before itself or from the dataset and combine them via a function such as summation ($\sum$), then apply a usually nonlinear activation function to generate the output (y) which will be transferred to the next neuron or given out as the output of the whole system.

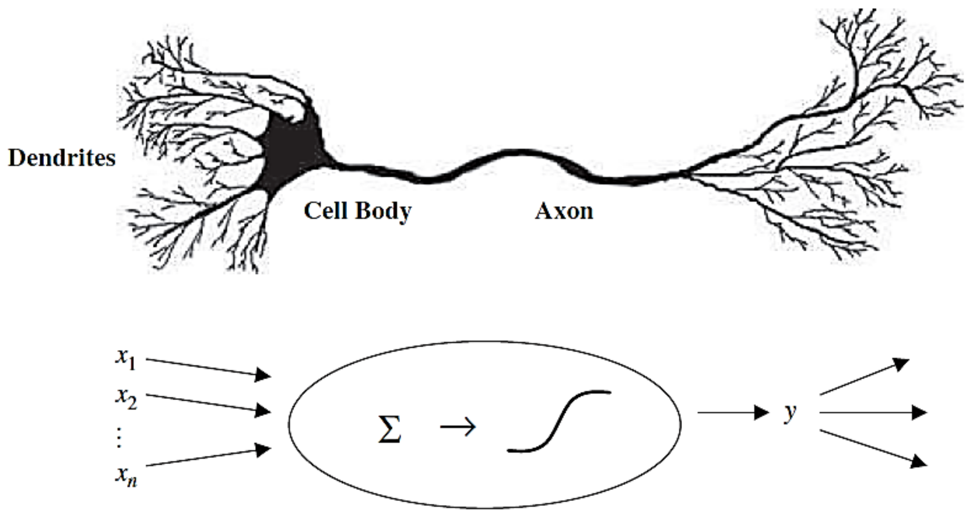

Figure 4.2 Real neuron and artificial neuron model (Larose & Larose, 2014)

There are three components of artificial neural networks:

- Neurons: Every specific neuron has input(s) and generates one single output to be transferred to multiple neurons when in need. The output of a single artificial neuron can be calculated by taking the weighted sum of all of its inputs, with an added bias and passing it through an activation function.
- Weights (Connections): An artificial neural network consists of many connections. Each connection provides an output to other neurons and it has a weight value representing the importance of connection. There is no limit on a neuron being able to have multiple inputs and output connections if the network is built correctly (Abbod, Catto, Linkens, & Hamdy, 2007).
- Propagation: It is applied to compute the input value for a neuron. Its value is calculated as the output of all the connections as a weighted sum. A bias term can be added to the result of the propagation (Dawson & Wilby, 1998).

Artificial neural networks require a large size of data as the training dataset where labels are included. When a certain observation from the training dataset is put into process, an output value is generated from the output node. Then after the comparison of this output value to the real value and the error is calculated as the output value that is subtracted from real value. The main goal of a network is to construct a model with such parameters that will minimize this error.

One of the most important perks of using an ANN is that they are quite robust since the network built has a large number of neurons (nodes). Having this perk allows the network to decide the action to be taken.

One of the most important cons of using an artificial neural network is that all attribute values must be standardized. In other words, the values must be between zero and one, including categorical variables. For this standardization process, the most commonly used normalization method is the min-max normalization. The equation of min-max normalization is given in Equation 4.3.

$$a^* = \frac{a-min(a)}{range(a)} = \frac{a-min(a)}{(a)\ -min(a)} \tag{4.3}$$

Since artificial neural networks creates continuous type of output data, they can be used for estimation or prediction which will allow them to be used for classification problems. Artificial neural networks can be seen in many disciplines thanks to their feature of being able to be trained over and over on nonlinear studies.

4.2.2 Unsupervised Learning

Contrary to supervised learning, these algorithms do not need human involvement. They are mostly used for exploratory data analysis, customer segmentation, and image recognition because they bring out the similarities and differences in information.

Some of the most commonly used and seen approaches for unsupervised learning are the following:

- Hierarchical Clustering: As the name implies, it is a method used for analysis when a hierarchy of clusters is desired.
- K-Means: The value k in K-Means determines the number of sets and must take this value as a parameter. The algorithm puts records of a statistically similar nature into the same group. An element is only allowed to belong to a set. The cluster is the value that represents the central cluster.
- DBSCAN: It is a density-based clustering algorithm (Ester, Kriegel, Sander, & Xu, 1996).
- Autoencoders: An autoencoder is a specific type of artificial neural network used to learn efficient codings of unlabeled data (Kramer, 1991).
- Self-organizing Map (SOM): SOM is a type of artificial neural network but it is trained using competitive learning rather than the error-correction learning used by other artificial neural networks. The SOM was introduced by the Finnish professor Teuvo Kohonen (Kohonen & Honkela, 2007).
- Deep Belief Networks (DBN): The DBN consists of directed RBMs (Restricted Boltzmann Machine) except the first RBM.

4.2.3 Semi-supervised Learning

Semi-supervised learning algorithms can be used when collection of small sized labeled data with a large sized unlabeled data is desired for training process. It is of great interest both in theory and in practice because it requires less human effort and gives higher accuracy (Zhu, 2005).

4.2.4 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward (Hu, Niu, Carrasco, Lennox, & Arvin, 2020). Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. The working principle of a basic reinforcement learning algorithm is shown in Figure 4.3.

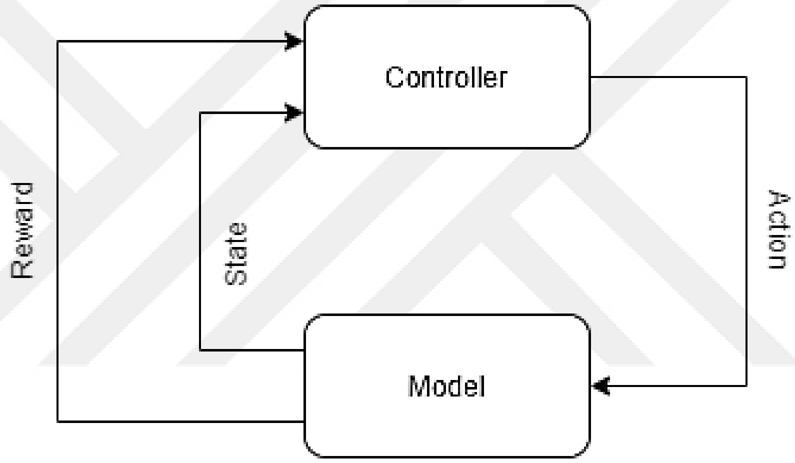

Figure 4.3 Basic diagram of reinforcement learning

In reinforcement learning, the controller receives the state values from the controlled model and a reward value associated with the last state transition. Then it decides upon an action to take which meets the criteria in order to send back to the model. As a response, the model makes a transition to a new state. This cycle is repeated until the stopping criteria is met. The main problem here is that a way of controlling the system/model needs to be learnt in order to maximize the total reward. The reward function is the objective feedback from the environment and rewards are integer or scalar variables associated with some states or state-action pairs. (Sutton & Barto, 1999). It defines the goal of the agent in a given state/situation. The state and action concepts of reinforcement learning are general. The action defines a decision the agent can possibly take and the state defines any factor that the agent considers for the decision making process.

There are two important actions needed to be taken in order to allow reinforcement learning algorithms to achieve its goal; the first is to use samples to compactly represent the dynamics of the control problem and the second is to use powerful function approximation methods to compactly represent value functions (Szepesvári, 2010).

4.3 Deep Learning

Deep Learning (DL) can be thought of as the mathematically complex evolutionary form of ML algorithms. Most of the simplest ML algorithms work very well on a large proportion of problems that might be considered as important. However, they are not very successful in solving some major AI problems like recognizing objects or recognizing speech (Goodfellow, Bengio, & Courville, 2016). This led researchers to look for a more advanced method; a method that can overcome the obstacles such as the difficulty when working with high dimensional data and the computational cost that comes with the hardware which are needed to supply the high dimensional spaces of complicated functions to be learnt.

Deep learning is defined as a neural network with more than three layers. These neural networks attempt to mimic the behavior of the human brain which allows them to learn from large amount of data. Unlike machine learning, deep learning eliminates some of data preprocessing. Deep learning algorithms can process unstructured data such as text, images etc. and they extract features autonomously without depending on human experiences.

Deep learning algorithms perform feature extraction in an automated way, which allows researchers to extract discriminative features with minimal domain knowledge and human effort (Najafabadi et al., 2015). These algorithms have a layered structure. In this architecture, high level feature extraction comes with a high computational work where it can be handled with high performance processing units.

Graphic processing units (GPU) are known for their high performance in computing large-scale data and matrices on a single machine. Thus, some distributed deep learning frameworks were developed in order to speed up the training process of deep learning models (Azarkhish, Rossi, Loi, & Benini, 2018; Yan, Zhu, Shyu, & Chen, 2016).

The historical journey of DL starts with the invention of the perceptron. The concept of perceptron traces back to 1958 when Rosenblatt published his study (Rosenblatt, 1958). Recurrent connections which are one of the keystones for deep learning were invented at the beginning of the 1980s with the study of Hopfield (Hopfield, 1982). After only four years, Rumelhart's team published their study where they invented the backpropagation algorithm without which deep learning would not exist as it exists now (Rumelhart, Hinton, & Williams, 1986). This study was followed by a paper published in 1989 where training deep recurrent networks was detailed (Williams & Zipser, 1989).

Despite promising breakthroughs in the late 1980s, in the 1990s AI entered a new winter era, during which there were few developments. Deep learning approaches were discredited because of their average performance, mostly because of a lack of training data and computational power. But during this era, convolutional neural networks (CNNs) which are one of the greatest invention in deep learning were developed in the early 1990s (Le Cunn & Bengio, 1995) and it was followed by the invention of long short-term memory (LSTM) which is still widely used today for sequence modeling (Hochreiter & Schmidhuber, 1997). At the end of the 1990s, Yoshua Bengio and Yann

LeCun, regarded today as two of the godfathers in deep learning, generalized document recognition via neural networks trained by gradient descent and introduced graph transformer networks (LeCun, Bottou, Bengio, & Haffner, 1998). In 2000s, belief networks were popular (Geoffrey E. Hinton, Osindero, & Teh, 2006; Honglak Lee, Grosse, Ranganath, & Ng, 2009) and neural networks were applied in multiple stages (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009).

Entering 2010s, a new and more progressive era began for deep learning. Many studies were published where one could pile up autoencoders for feature extraction (Vincent et al., 2010). Use of the rectified linear unit (ReLU) as an activation function in deep learning got popularized (Nair & Hinton, 2010). Deep learning became mainstream when ImageNet's success was thrown away with deep convolutional network study in 2012 (Krizhevsky, Sutskever, & Hinton, 2012). Adadelta, a popular adaptive learning rate method, was invented the same year (Zeiler, 2012). The idea of dropout was first published (Geoffrey E. Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). In 2014, Kingma designed a new approach to autoencoders, in a Bayesian framework called the variational autoencoder (VAE) (Kingma & Welling, 2014), gated recurrent units, an improvement on LSTMs, were introduced (Cho et al., 2014) and generative adversarial networks (GAN) were proposed (Goodfellow et al., 2014). On the same year, Adam, probably the most used optimizer, was described by (Kingma & Ba, 2017) and Ross Girshick's team, revolutionized object detection with deep learning and method called R-CNN (Girshick, Donahue, Darrell, & Malik, 2014). In 2015, the idea of merging CNN and LSTM (Donahue et al., 2015.; Shi et al., 2015) shared along with batch normalization (Ioffe & Szegedy, 2015). In 2016 Google officially released Tensorflow (Abadi et al., 2016) which is labeled as a system for large scale machine learning.

In Table 4.2, a brief information about the most commonly used deep learning networks is shared and their key points are listed:

Table 4.2 Summary of popular deep learning networks

Deep Learning Networks	Description	Key Point
CNN	Usually used for image recognition	Extended for computer vision
DBN	Unsupervised learning method	Restricted Boltzmann Machines with directed connections
GAN	Unsupervised learning method	Game theoretical network
RNN	Usually used for natural language processing and speech processing	Good for sequential information
RvNN	Usually used for natural language processing	Uses like a tree-like structure
VAE	Unsupervised learning method	Probabilistic graphical model

4.3.1 Convolutional Neural Network

Convolutional neural network (CNN) is a widely used algorithm in deep learning. The CNN algorithm, which is a forward-looking neural network, was inspired by the visual center of animals. It is thought to be a version of multilayer perceptrons (MLP). It is used for different applications such as computer vision (Krizhevsky et al., 2012), speech processing (Dahl, Yu, Deng, & Acero, 2012) and natural language processing (Yin, Kann, Yu, & Schütze, 2017). CNN has three major advantages. These are; equivalent representations, parameter sharing and sparse interactions.

The term convolution means generating a third mathematical function with the combination of two functions. The mathematical convolution operation here can be thought of as the response of a neuron to stimuli from its own stimulus field. CNN algorithms are applied in many different fields such as natural language processing (NLP), biomedical, especially in the field of image and sound processing.

Typically, CNN architecture consist of convolutional layers, pooling layers and dense layers. The network structure of a convolutional neural network is shown in Figure 4.4.

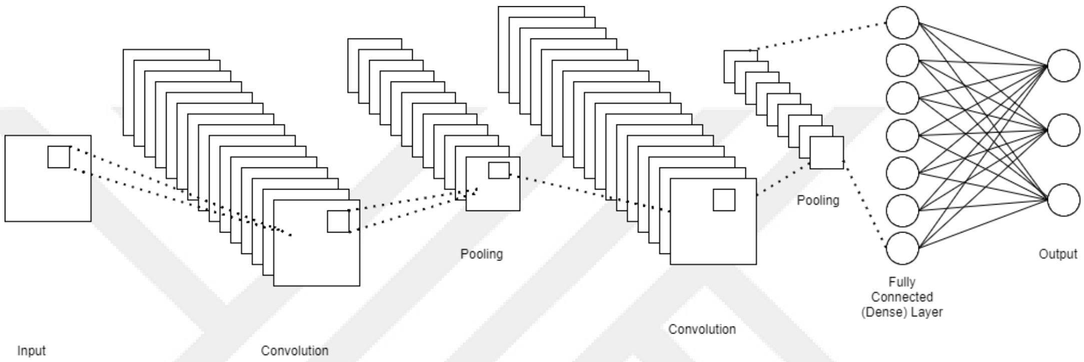

Figure 4.4 Structure of a convolutional neural network

Each of these convolutional layers contains several filters, in other words, kernels. These kernels have the same parameters which are weight and bias and they are used to generate feature maps. At the end of the convolutional layer, an activation function is applied.

Pooling layers, commonly known as subsampling layers, are used for decreasing the parameters in the network, speed up the training process and try to keep network from overfitting.

Fully connected layers, which are also known as dense layers, are like regular artificial neural networks. The purpose of these layers is to take features from previous layers and generate a high level data. The last layer of a CNN is usually used to generates a value which contains the information of label probability of the given instance (Pouyanfar et al., 2019).

4.3.2 Deep Belief Network

Deep belief networks (DBNs) are probabilistic generative models with multiple layers of stochastic hidden units above a single bottom layer of observed variables that represent a data vector (Dahl et al., 2012). DBNs have directed connections to all other layers from the layer before them except the first two layers. DBNs can also be defined as a set of Restricted Boltzmann Machines (RBM) (G. E. Hinton, 2006).

Restricted Boltzmann Machines are some type of undirected graphical model constructed from a layer of stochastic hidden units and a layer of stochastic visible units that will either be Bernoulli or Gaussian distributed conditional on the hidden units (Smolensky, 1986). RBMs only contains inter layers.

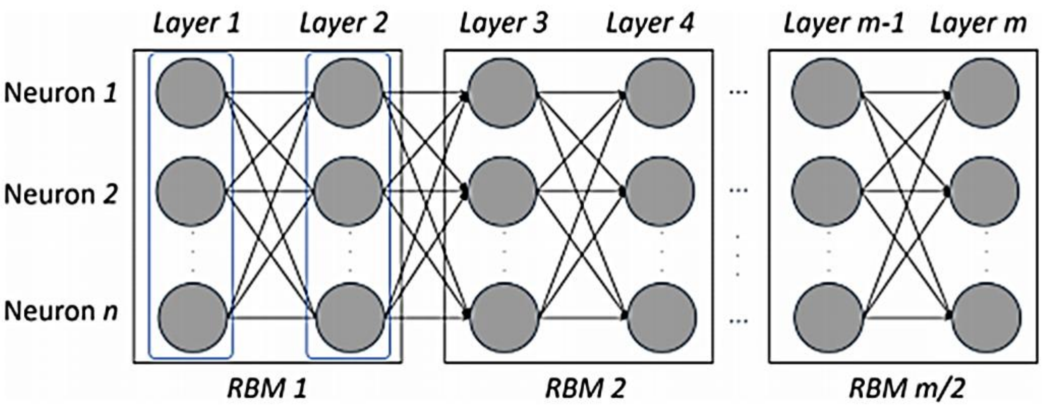

Figure 4.5 Basic diagram of deep belief network (Khanna & Awad, 2015)

In Figure 4.5, the basic diagram of deep belief networks is given. As it can be seen, except for the first RBM, it consists of directed RBMs. This network architecture allows to reduce the training complexity significantly and makes deep learning possible.

4.3.3 Generative Adversarial Network

Generative adversarial network (GAN) is a framework for estimating generative models via an adversarial process (Goodfellow et al., 2014). It contains two different models which are a generative model G and a discriminative model D. The generative model contains information about the distribution of data where the discriminative model predicts a value to decide if a sample came from the training data rather than G.

A generative model is trained by maximizing the probability of a discriminative model making a mistake.

4.3.4 Recurrent Neural Network

Recurrent neural network (RNN) is a very popular algorithm in deep learning. They are usually used in natural language processing and speech processing applications (Cho et al., 2014). RNNs use sequential information on the network unlike traditional methods (Pouyanfar et al., 2019). This is very important because the embedded structure in the data array transmits useful information.

One of the most important issues of RNNs is that it is too sensitive to the vanishing and exploding gradients (Glorot & Bengio, n.d.). This means, as new inputs enter the system, the effect of the initial inputs is reduced which means that this sensitivity reduces over time.

While feedforward architectures represent time explicitly with a fixed length, recurrent architectures represent time recursively. This feature enables the system to save complex signals as inter layers of this architecture which can be thought of as memories.

RNNs were not appealing thanks to vanishing and exploding gradient problems making RNNs hard to train until it was proven that this could be partly avoided with a simple solution such as gradient clipping (Mikolov, 2012). This method allowed these models to be trained on large datasets with more efficiency by backpropagation and stochastic gradient descent (Werbos, 1988; Williams & Zipser, 1989). In Figure 4.6, the diagram of a simple recurrent network is shown where x_t input, y_t is probability and h_t is the state of the hidden layer at that specific time.

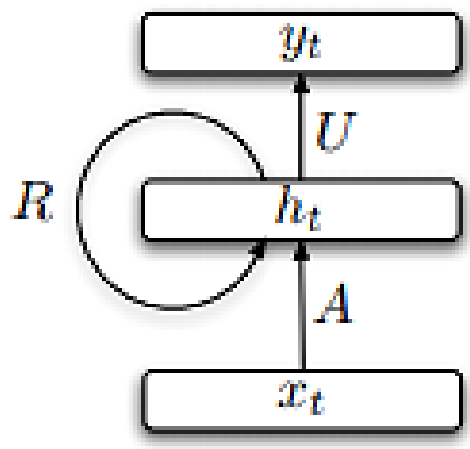

Figure 4.6 Diagram of a simple recurrent network (Tomas Mikolov, Joulin, Chopra, Mathieu, & Ranzato, 2015)

One of the biggest problems for RNNs are vanishing gradients because this causes the magnitude value to shrink exponentially. Another reason that caused RNNs to suffer from the vanishing gradient problem was the standard nonlinearities having zero gradient. But this problem was partially solved DL by using a simple technique named the rectified linear units (Nair & Hinton, 2010).

One of the most important architectures was the long short-term memory (LSTM) to solve the vanishing gradient problem (Hochreiter & Schmidhuber, 1997). LSTM is thought of as a slightly changed type of simple RNNs. It relies on a structure designed with gates which control the flow of information to inter layers. The aim when using these gates is to allow the network to remember information for a longer period. Figure 4.7 shows the block diagram of LSTM where i, o, f and c are input gate, output gate, forget gate and memory cells, respectively.

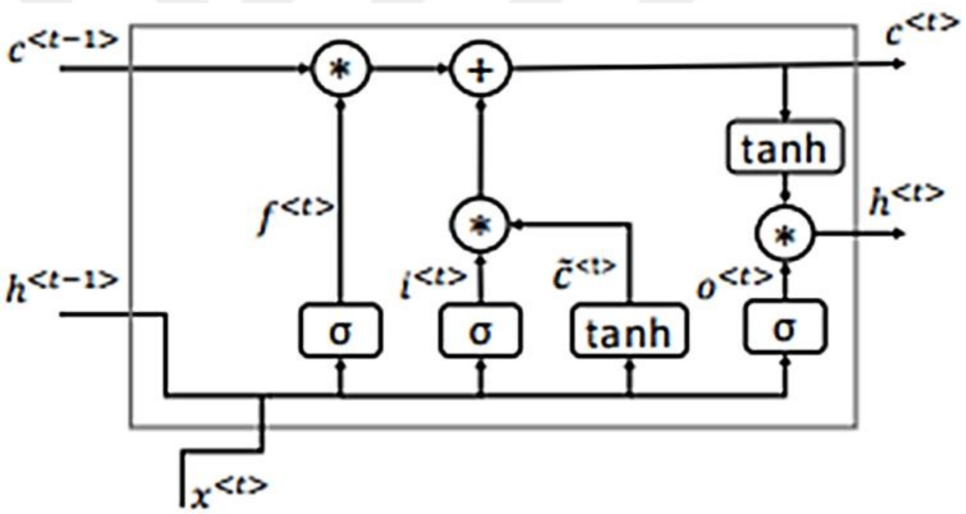

Figure 4.7 Block diagram of long short-term memory (Samui, Chakrabarti, & Ghosh, 2018)

4.3.5 Recursive Neural Network

Recursive neural networks (RvNN) can be used both to classify the outputs using compositional vectors and make predictions in a hierarchical structure. It is accepted that RvNN is especially successful with the cases that deal with natural language processing problems (Socher, Lin, Ng, & Manning, n.d.).

4.3.6 Variational Autoencoder

Autoencoders are unsupervised network models that are trained to recover data from compressed versions by compressing data with their encoder and decoder structure. Variational Autoencoder (VAE), utilizes the log-likelihood of the data and leverages the strategy of deriving a lower bound estimator from the directed graphical models with continuous latent variables (Kingma & Welling, 2014).

CHAPTER 5

MATERIAL AND METHODOLOGY

5.1 Introduction

In this chapter, the analysis of the cotton plant images using deep learning algorithms Convolutional Neural Network, Long Short-Term Memory and Convolutional Long Short-Term Memory (ConvLSTM), a combined version of these two, is explained. Since the images are two dimensional, 2D image analysis techniques are given. For four different diseases detection and prediction processes are explained.

5.2 Material

Cotton cultivation has a special place in Turkey. Considering the behavior of cotton plant and the climate of Turkey, cotton harvesting must be finished before the fall when heavy rains visit the country.

Parts of the cotton images used for this study were obtained from Söke in Aydın during the summer and early fall in 2019. 592 images (280 infected and 312 healthy) were taken from the cotton field in Söke and 1408 were obtained from online repositories. The total values of labeled images are given in Table 5.1.

Table 5.1 The number of images used for each disease

Disease	Count
Leaf Angular Spot	172
Magnesium Deficiency	172
Gray Mildew	172
Verticillium and Fusarium Wilt	172
Healthy	1312

5.3 Method

This section contains detailed information about methods which are used in this study. The subsection about 2D image analysis explains how preprocessing and feature extraction works. In the subsection on detection and recognition, how the chosen algorithms are used in order to detect and predict diseases (decision of parameters, optimizers etc.) is explained.

5.3.1 2D Image Processing

This section contains explanatory information about preprocessing and feature extraction processes applied on images in order to have a deeper understanding of cotton images. As a result, a deeper insight into how deep learning algorithms convert an image into something they can understand and handle can be obtained. A flowchart of 2D image processing method is shown in Figure 5.1.

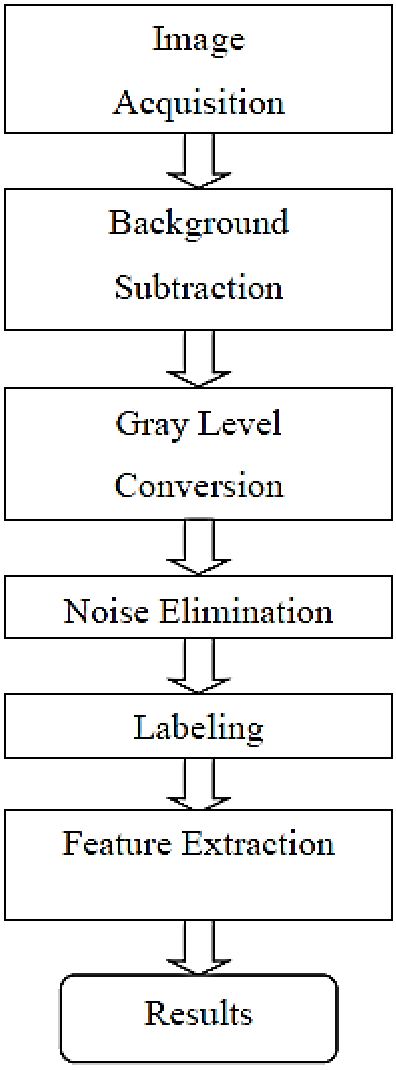

Figure 5.1 Flowchart of the 2D image processing method (Sinecen, 2011)

The images are preprocessed with two dimensional processing techniques. At first, all the infected and healthy cotton plant images are converted to gray scale images as shown in Figure 5.2. Then, the gray scaled image is used for edge detection with 5x5 filter (Figure 5.3) and at last a threshold of 125 is applied to that image (Figure 5.4).

Figure 5.2 Gray scale image of cotton infected with verticillium and fusarium wilt

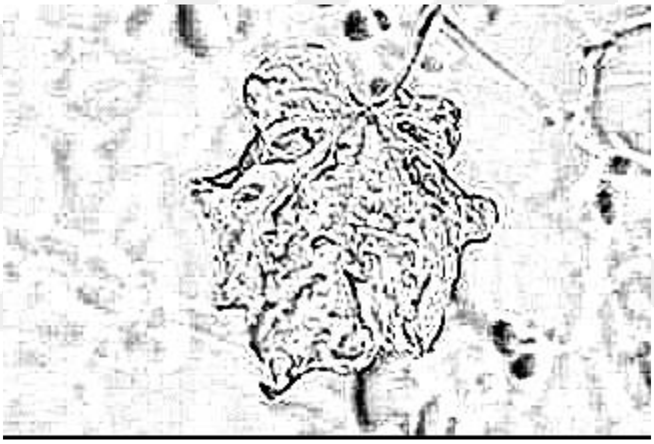

Figure 5.3 Edge detected image of the infected cotton

Figure 5.4 Threshold applied edge detected image

Today, one can make use of libraries to prepare an image to feed the deep learning algorithm used. Scikit-image, OpenCV, Matplotlib and SciPy are some of the most used image processing libraries in python that allow people to prepare their data faster for their classification tasks.

5.3.2 Detection and Recognition

The detection process of cotton leaves inside the images obtained and the recognition of whether the cotton plant is infected (if infected which disease it has) or not is explained in this section. This classification process is realized using the convolutional neural network, long short-term memory and convolutional long short-term memory algorithms. Each method is explained briefly.

The programming part for this thesis is realized on PyCharm by JetBrains, which is an integrated development environment (IDE), used for specifically for the Python programming language.

Images used for this thesis are color images, which means they contain data of red-green-blue (RGB) combination. RGB images need to be stored in three dimensional arrays in order to be processed. First two dimensions correspond to the height and width of the image and the third dimension corresponds the colors that present in each pixel. Since the classification algorithms need their input to be arrays, a conversion process must be applied to images using one of the libraries mentioned in section 5.3.1.

Cotton plant images used in this study has 480*360 pixels (the ones obtained from camera) and 324*217 pixels (the ones obtained from online repository) resolutions. Therefore, they need to be reshaped in order to be processed by the deep learning algorithms. Reshaping should be done in consideration of output classes to be predicted. Table 5.2 shows the classifier outputs.

Table 5.2 Class labels of cotton diseases

Disease	Class
Healthy	1
Leaf Angular Spot	2
Magnesium Deficiency	3
Gray Mildew	4
Verticillium and Fusarium Wilt	5

After the reshaping, the data need to be normalized since neural network models usually require scaled data. The data are normalized using the min-max normalization method mentioned in section 4.2.1.4. The values are converted from 0-255 to 0-1 since the model has target as one-hot coded with total number of 5 classes.

In the next three subsections, models are trained, get their validation score and are tuned many times and three models are chosen for each method which are thought of as weak, medium and strong models.

5.3.2.1 Classification using CNN

Convolutional neural networks are mainly used in image analysis tasks such as object detection and image recognition. In this study, the 80 percent of the dataset is used as training dataset and the rest is used as validation dataset.

With purpose of deciding the best CNN architecture, different architectures were examined. For these architectures, number of convolutional layers, dropout values and number of nodes for each convolutional layer were changed and loss values were stored through epochs in a csv type of file. The validation loss graphs for convolutional neural network's weak, strong and medium models are shown in Figure 5.5. For these different tryouts a fixed learning rate, a fixed number of batch size and a fixed number of dropout value with a fixed optimizer model were chosen. These values are given in Table 5.3.

Table 5.3 Values of learning rate, batch size and dropout for CNN

Parameter	**Value**
Learning Rate	0.1
Batch Size	64
Dropout	0.9

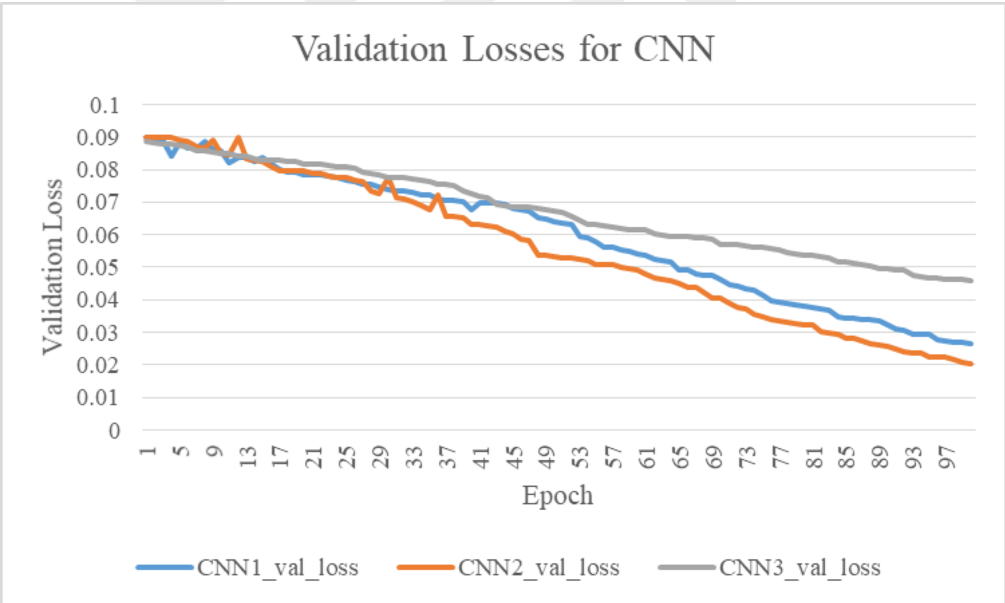

Figure 5.5 Validation loss graph for convolutional neural networks best, medium and worst case

Figure 5.5 shows that the CNN2 reaches the lowest validation loss value. Therefore this architecture can be chosen for classifying of cotton diseases. CNN2 has 4 convolution layers with ReLU and 2 dense layers at the end of the architecture.

5.3.2.2 Classification using LSTM

Image recognition using LSTM is not as popular as image recognition with CNN. However, it is used from time to time since they have memory units which help recognition. As mentioned in section 4.3.4, LSTM models are intended for use with data that is comprised of long sequences of data. Addition to what CNN needs to compute, LSTM have time steps which are defined as how many steps back is desired to go because LSTM is considered as a sequence generator.

The validation loss graphs for the weak, strong and medium models for the long short-term memory network are shown in Figure 5.6. For these different tryouts a fixed learning rate, a fixed number of batch size, a fixed number of dropout value and a fixed step back value with a fixed optimizer model are chosen. These values are given in Table 5.4.

Table 5.4 Values of learning rate, batch size, dropout and step for LSTM

Parameter	**Value**
Learning Rate	0.1
Batch Size	64
Dropout	0.9
Step	5

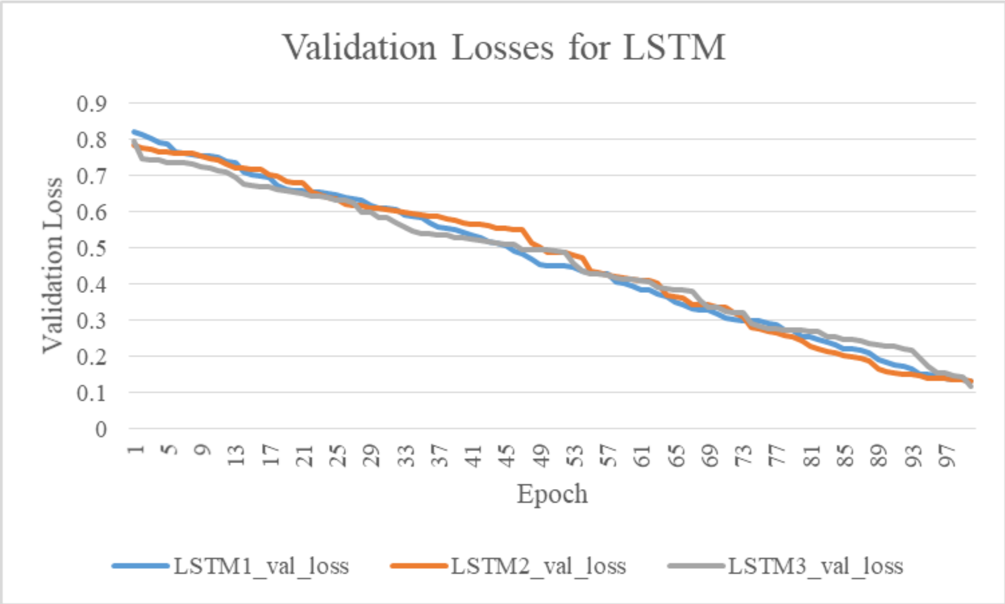

Figure 5.6 Validation loss graph for long short-term memory networks best, medium and worst case

Figure 5.6 shows that the LSTM3 reaches the lowest validation loss value. Therefore this architecture can be chosen for classifying of cotton diseases. LSTM3 has the highest number of sequential layers with SGD and 2 dense layers at the end of the architecture.

5.3.2.3 Classification using ConvLSTM

ConvLSTM is a method where LSTM part is usually used to find out about the relationships among the label variables and the CNN part is used to learn feature representations from image data (J. Wang et al., 2016). With this structure containing both convolution and memory parts, it is expected that this architecture to be more successful on flowing image data such as videos. In Figure 5.7, the architecture of a ConvLSTM is shown.

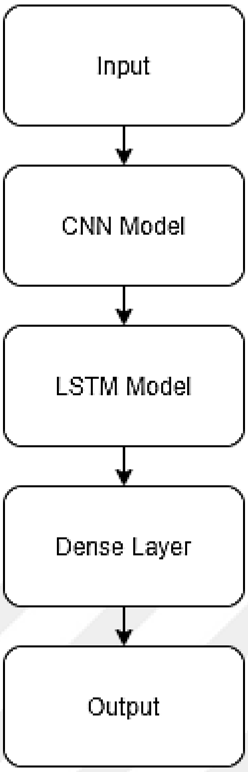

Figure 5.7 Architecture of ConvLSTM

The validation loss graphs for the weak, strong and medium models for convolutional long short-term memory network are shown in Figure 5.8. For these different tryouts a fixed learning rate, a fixed number of batch size, a fixed number of dropout value and a fixed step back value with a fixed optimizer model were chosen. These values are given in Table 5.5.

Table 5.5 Values of learning rate, batch size, dropout and step for ConvLSTM

Parameter	**Value**
Learning Rate	0.1
Batch Size	64
Dropout	0.9
Step	5

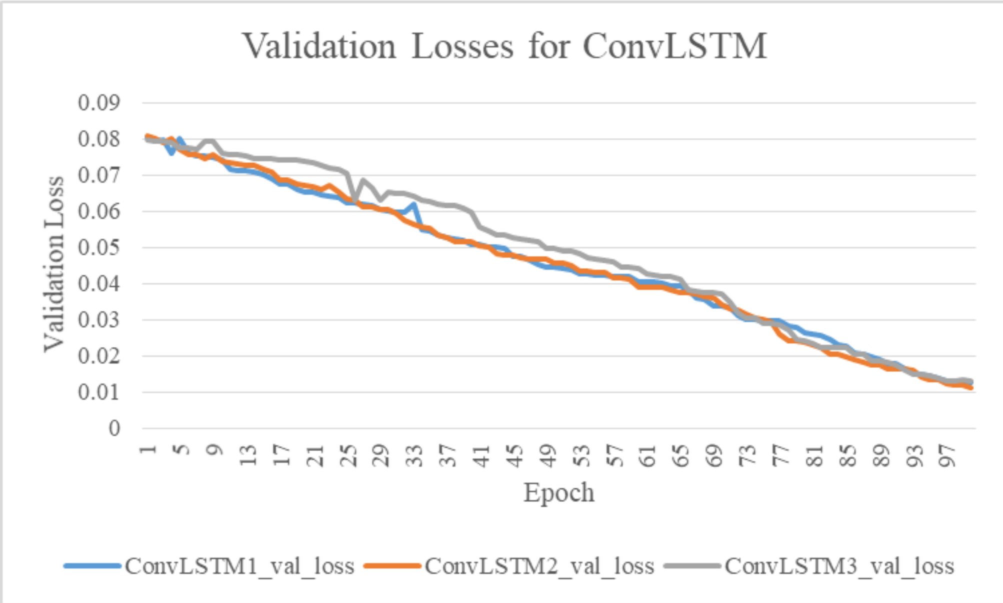

Figure 5.8 Validation loss graph for convolutional long short-term memory networks best, medium and worst case

Figure 5.8 shows that the ConvLSTM2 reaches the lowest validation loss value. Therefore, this architecture can be chosen for classification of cotton diseases. ConvLSTM2 has highest number of sequential layers with ReLU and 2 dense layers at the end of the architecture. Unlike other two methods, ConvLSTM models have a lower difference in validation loss.

5.4 Results and Discussion

Until this section, the preprocessing part of the cotton images has been explained in order to have a better insight and understanding on how the input data needs to be prepared to feed deep learning algorithms. Then, the decision process was needed to choose the optimum architecture to perform prediction/classification of the diseases that affects cotton plant. To make that decision, validation scores of many tries for the chosen three deep learning method needed to be compared and the architecture with lowest validation loss needed to be chosen. While calculating the validation scores, all parameters that does affect the model but not structure itself are fixed into an optimal point. In Table 5.6, the architectures that are chosen for each deep learning method that are planned to be used in prediction process are given.

Table 5.6 Architectures chosen for each deep learning method

Deep Learning Method	**Architecture Number**
CNN	CNN2
LSTM	LSTM3
ConvLSTM	CNN2

For the classification part of the work, these three architectures were used with learning rate ranging from 0.001 to 0.9, dropout ranging from 0.2 to 0.9, batch size ranging from 32 to 256.

Figure 5.9 shows how accuracy of the designed CNN model for classification of health of cotton plant changes. This model has 90.13 as its lowest value where learning rate is 0.001 and the batch size is 32 while having 95.03 as its highest value where learning rate is 0.5 and the batch size is 128. It is seen that changing the learning rate or batch size alone affects the accuracy for the model by not that much. This situation can be explained by the working principle of CNNs. It is well known that CNNs have high success rate at classifying images (Hussain, Bird, & Faria, 2018; Hyungtae Lee & Kwon, 2017; Lei, Pan, & Huang, 2019; W. Li, Chen, Zhang, Li, & Du, 2019; Wei et al., 2016; Zhang, Li, & Du, 2018).

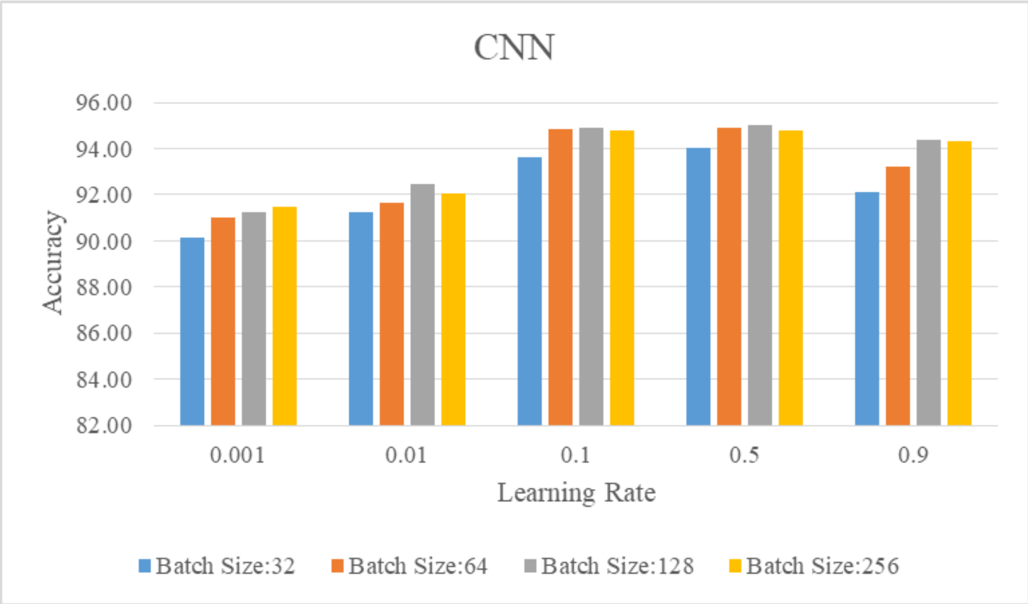

Figure 5.9 Accuracy of CNN

Figure 5.10 shows the behavior of accuracy for the designed LSTM model to classify health of cotton plant. This model has 82.30 as its lowest value where learning rate is 0.001 and the batch size is 32 while having 89.70 as its highest value where learning rate is 0.9 and the batch size is 128.

Unlike CNN, LSTM alone is not preferred in image classification problems. Even though it has an increasing accuracy value as the learning rate increases, its highest accuracy value is below the lowest accuracy value of CNN.

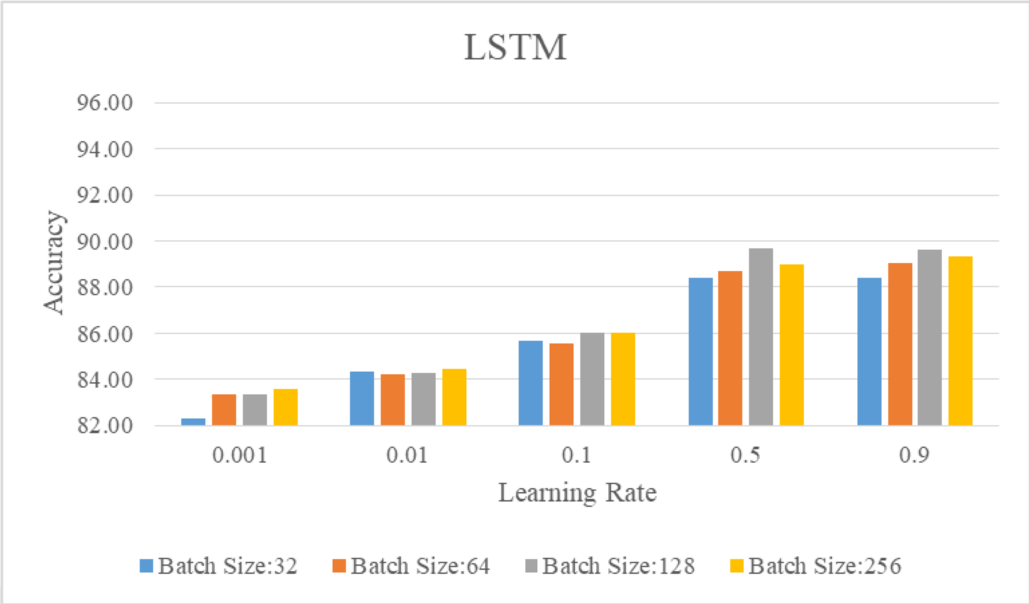

Figure 5.10 Accuracy of LSTM

Figure 5.11 shows the accuracy values of the designed ConvLSTM model to classify health of cotton plant. This model has 91.30 as its lowest value where learning rate is 0.001 and the batch size is 32 while having 95.82 as its highest value where learning rate is 0.5 and the batch size is 128. ConvLSTM has highest minimum accuracy value among the three deep learning models. ConvLSTM accuracy values oscillate less than other deep learning methods as learning rate and batch size changes.

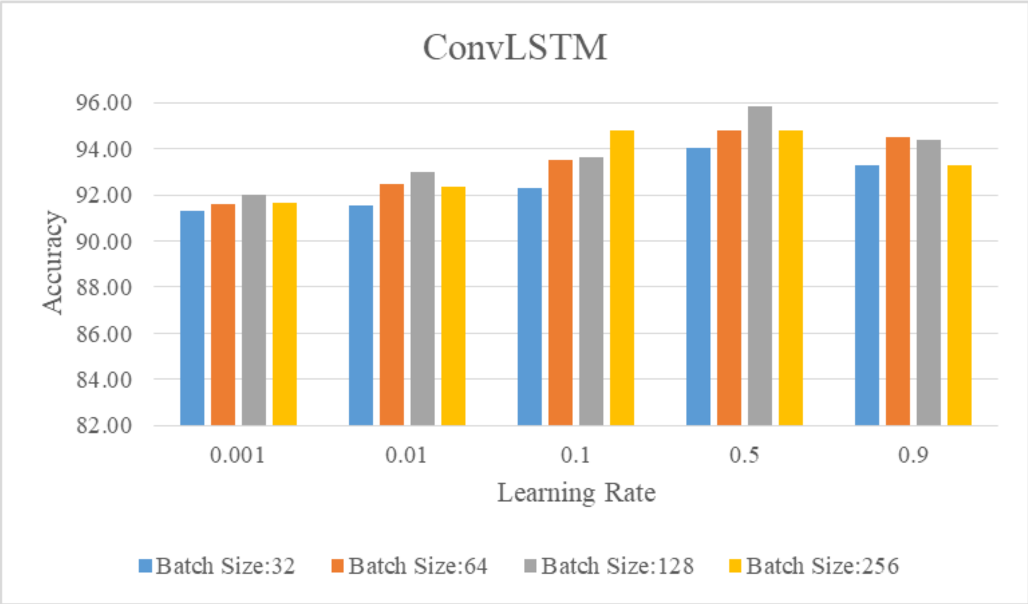

Figure 5.11 Accuracy of ConvLSTM

Knowing that ConvLSTM is a combined method, it is not surprising that it gives the best accuracy among the three models. With having such low oscillation, it can be said that ConvLSTM reaches stability before the others. That characteristic can be very important since it is well known that having a high learning rate usually increases the needed computational time.

After making this comparison between these three methods, the 50 of randomly chosen cotton images were disrupted. Some of their parts with size of 5x5 were taken out of the image randomly. The methods compared in this chapter are used to predict the shuffled images for 1000 times. Table 5.7 contains the minimum and maximum accuracy of these three models on the disrupted images. While ConvLSTM has the highest accuracy value among these three deep learning algorithms with the decided architectures when they are handling regular (undisrupted) images, when they are expected to predict the diseases on disrupted data CNN has highest accuracy value.

Table 5.7 Minimum and maximum accuracy for ConvLSTM on disrupted images

Deep Learning Method	Minimum	Maximum
CNN	87.26	91.23
LSTM	72.25	86.37
ConvLSTM	88.14	91.20

In this chapter, the material and methods that are used in this work are investigated. The architectures used in this study are explained and compared. Processed images are used for classification and the performance of classification is addressed. In next chapter, the work done is concluded.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In the thesis, a 2D image processing method is explained in order to have an understanding of how deep learning algorithms use images to prepare for data extraction. This revealed that preprocessing is an important step for deep learning in order to have successful classification and prediction results.

The cotton cultivation is very crucial in Turkey and in the world. Detecting cotton diseases with no human interaction can save time which might be enough for saving the plant. Therefore, the comparison of deep learning algorithms are important for cotton disease classification. After many experiments, extracting architectures with high success in detecting and classifying cotton diseases create a ground for future studies.

Tuning architectures of CNN, LSTM and ConvLSTM also has a great importance because after optimizing the parameters of these deep learning algorithms, they are used to make predictions on disrupted image. If models are not optimized with robustness, overfitting can be a real issue. Since three architectures have their score almost as high as they are used in undisrupted image, it is fair to say that these architectures can be used to predict with chosen parameters. The images are disrupted in a way where making prediction on those images means these three algorithm may help having information of infected cotton plants before they show full symptoms.

Software used for the analysis given in the thesis has been developed using the Python programming language. The libraries such as NumPy, Matplotlib, Scikit-image, Tensorflow were used during programming in order to realize preprocessing, feature extraction, classification and prediction.

As a result, if a system with high accuracy is desired, it is better to choose ConvLSTM networks at high learning rate. If a faster system that can work with rather lower but enough accuracy is desired, it is better to choose CNNs over ConvLSTM. Because ConvLSTM has the memory parts every time it steps back in order to get the valuable data from the previous steps. But this increases the computational time. LSTM alone should not be chosen for input data containing only cotton plant images unless a very low accuracy rate is tolerable. Even with the lowest accuracy among these three algorithms, accuracy value of LSTM is surprisingly high on this specific image set. The reason behind this can be using the time stamps. Time stamps are used to create sequences to help LSTM work better.

6.2 Future Work

The work presented in this thesis is hoped to inspire future studies. Further studies on predicting diseases of cotton plant can be enhanced with adding some data other than images, such as sensor data. Thus, a better prediction system can be created for cotton disease prediction. With a fully automated disease prediction and early warning system, not only cotton cultivation but any agricultural production would have a new perspective.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation*, 265-283.

Abbod, M. F., Catto, J. W., Linkens, D. A., & Hamdy, F. C. (2007). Application of artificial intelligence to the management of urological cancer. *The Journal of Urology*, *178*(4), 1150–1156.

Abdel-Hamied, M. R. (2021). Analytical study of factors affecting yarn coefficient of mass variation estimated by artificial neural networks. *Journal of Fiber Bioengineering and Informatics*, *14*(1), 13–20.

Al Hiary, H., Bani Ahmad, S., Reyalat, M., Braik, M., & ALRahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, *17*(1), 31–38. https://doi.org/10.5120/2183-2754

Azarkhish, E., Rossi, D., Loi, I., & Benini, L. (2018). Neurostream: Scalable and energy efficient deep learning with smart memory cubes. *IEEE Transactions on Parallel and Distributed Systems*, *29*(2), 420–434. https://doi.org/10.1109/TPDS.2017.2752706

Baheti, H., Bhandari, P., Thakre, V. R., Tomar, A., & Rai, M. K. (n.d.). Studies on Trichodermaviride 1.15% WP for Bioefficacy, Phytotoxicity and effect on beneficial Microbes against Seedling Root Rot in Cotton. *The Journal of Rural and Agricultural Research*, 16(2), 49-53.

Bhimte, N. R., & Thool, V. R. (2018). Diseases detection of cotton leaf spot using image processing and SVM classifier. *2018 Second International Conference on*

Intelligent Computing and Control Systems (ICICCS), 340–344. https://doi.org/10.1109/ICCONS.2018.8662906

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv preprint ArXiv:1406.1078.*

ÇörekciOğlu, M., Ercan, E., & Aras EliBüyük, S. (2021). Yapay Sinir Ağı Yöntemlerinin Tekstil Sektöründe Kullanım Uygulamaları. *Teknik Bilimler Dergisi*, *11*(2), 14-20. https://doi.org/10.35354/tbed.884531

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia* (21-49). Berlin: Springer.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42. https://doi.org/10.1109/TASL.2011.2134090

Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, *43*(1), 47–66.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 2625-2634.

Erdoğan, O., Bölek, Y., & Göre, M. E. (2016). Biological Control of Cotton Seedling Diseases by Fluorescent Pseudomonas spp. *Tarım Bilimleri Dergisi*, *22*(3), 398–407.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Conference on Knowledge Discovery and Data Mining*, 226–231.

Forsyth, D., & Ponce, J. (2011). *Computer vision: A modern approach.* New Jersey: Prentice hall.

Fuentes, A., Yoon, S., Kim, S., & Park, D. (2017). A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors, 17*(9), 2022. https://doi.org/10.3390/s17092022

Gençer, O., Özüdoğru, T., Kaynak, M. A., Yılmaz, A., & Ören, N. (2005). Türkiye'de pamuk üretimi ve sorunları. *TMMOB, Ziraat Mühendisleri Odası, Türkiye Ziraat Mühendisligi, VI. Teknik Kongresi. I, 3*(7), 1-8.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587. Columbus, OH, USA: IEEE. https://doi.org/10.1109/CVPR.2014.81

Glorot, X., & Bengio, Y. (n.d.). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249-256.

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing*, *Hoboken*. New Jersey: Pearson.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. London: MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*, 1-9.

Hamet, P., & Tremblay, J. (2017). Artificial intelligence in medicine. *Metabolism*, *69*, 36–40.

Hand, D. J. (2007). Principles of data mining. *Drug Safety*, *30*(7), 621–622.

Hinton, G. E. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hinton, Geoffrey E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, Geoffrey E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv preprint ArXiv:1207.0580.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.

Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, *69*(12), 14413–14423.

Hussain, M., Bird, J., & Faria, D. (2018). A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence, 191-202.*

Ioffe, S., & Szegedy, C. (n.d.). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448-456.

Jain, R. K., & Kasturi, R. (n.d.). *R, & Schunk, BG (1995). Machine Vision*. New York: McGraw-Hill.

Jamuna, K. S., Karpagavalli, S., Vijaya, M. S., Revathi, P., Gokilavani, S., & Madhiya, E. (2010). Classification of seed cotton yield based on the growth stages of cotton crop using machine learning techniques. In *2010 International Conference on Advances in Computer Engineering*, 312–315. Bangalore, India: IEEE. https://doi.org/10.1109/ACE.2010.71

Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? *2009 IEEE 12th International Conference on Computer Vision*, 2146–2153.

Katsantonis, D., Hillocks, R. J., & Gowen, S. (2003). Comparative effect of root-knot nematode on severity of Verticillium and Fusarium wilt in cotton. *Phytoparasitica*, *31*(2), 154–162.

Khanna, R., & Awad, M. (2015). Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. https://doi.org/10.1007/978-1-4302-5990-9

Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. *ArXiv preprint ArXiv:1412.6980.*

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *ArXiv preprint ArXiv:1312.6114.*

Kohonen, T., & Honkela, T. (2007). Kohonen network. *Scholarpedia*, *2*(1), 1568.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, *37*(2), 233–243.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097-1105.

Larose, D. T., & Larose, C. D. (2014). *Discovering knowledge in data: An introduction to data mining* (Vol. 4). New Yok: John Wiley & Sons.

Le Cunn, Y., & Bengio, Y. (1995). Convolutional neural networks for images, speech and time series. *AT&T Laboratories*, 1–14.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, Honglak, Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, 609–616.

Lee, Hyungtae, & Kwon, H. (2017). Going Deeper With Contextual CNN for Hyperspectral Image Classification. *IEEE Transactions on Image Processing*, *26*(10), 4843–4855.

Lei, X., Pan, H., & Huang, X. (2019). A dilated CNN model for image classification. *IEEE Access*, *7*, 124087–124095.

Li, H., Ji, R., Zhang, J., Yuan, X., Hu, K., & Qi, L. (2011). WEB-based intelligent diagnosis system for cotton diseases control. In *Computer and Computing Technologies in Agriculture IV* (pp. 483–490). Berlin: Springer.

Li, W., Chen, C., Zhang, M., Li, H., & Du, Q. (2019). Data augmentation for hyperspectral image classification with deep CNN. *IEEE Geoscience and Remote Sensing Letters*, *16*(4), 593–597.

McAndrew, A. (2004). An introduction to digital image processing with matlab notes for scm2511 image processing. *School of Computer Science and Mathematics, Victoria University of Technology*, *264*(1), 1–264.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Mikolov, Tomáš. (n.d.). *Statistical language models based on neural networks*. Phd. Thesis, Brno University of Technology, Brno.

Mikolov, Tomas, Joulin, A., Chopra, S., Mathieu, M., & Ranzato, M. (2015). Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753.*

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, *7*, 1419.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. New York: John Wiley & Sons.

Nair, V., & Hinton, G. E. (n.d.). Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 8, 807-814.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1.

Ongsulee, P. (2017). Artificial intelligence, machine learning and deep learning. *2017 15th International Conference on ICT and Knowledge Engineering (ICT KE)*, 1–6.

Patil, S. P., & Zambre, R. S. (2014). Classification of cotton leaf spot disease using support vector machine. *International Journal of Engineering Research*, 3(4), 1511-1514.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., et al. (2019). A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Computing Surveys*, *51*(5), 1–36.

Prajapati, B. S., Dabhi, V. K., & Prajapati, H. B. (n.d.). A Survey on Detection and Classification of Cotton Leaf Diseases. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, 2499-2506.

Revathi, P., & Hemalatha, M. (2005). Classification of Cotton diseases using cross information gain_minimal resource allocation network classifier with particle swarm optimization. *Journal of Theoretical and Applied Information Technology*, 60(1), 184-188.

Revathi, P., & Hemalatha, M. (2012). Advance computing enrichment evaluation of cotton leaf spot disease detection using image edge detection. *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, 1–5. Coimbatore: IEEE.

Rezene, Y., Tesfaye, K., Clare, M., & Gepts, P. (2018). Pathotypes characterization and virulence diversity of Pseudocercospora griseola the causal agent of angular leaf spot disease collected from major common bean (Phaseolus vulgaris L.) growing areas of Ethiopia. *J Plant Pathol Microbiol*, *9*(445), 2.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Rothe, P. R., & Kshirsagar, R. V. (2015). Cotton leaf disease identification using pattern recognition techniques. *2015 International Conference on Pervasive Computing (ICPC)*, 1–6. Pune, India: IEEE.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Russ, J. C. (2006). *The Image Processing Handbook (4th ed.).* North Carolina: CRC Press LLC

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229.

Samui, S., Chakrabarti, I., & Ghosh, S. K. (2018). Tensor-train long short-term memory for monaural speech enhancement. *ArXiv preprint ArXiv:1812.10095.*

Sarangdhar, A. A., & Pawar, V. R. (2017). Machine learning regression technique for cotton leaf disease detection and controlling using IoT. *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, 449–454. Coimbatore: IEEE.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., & Woo, W. (n.d.). *Convolutional LSTM network:* A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802-810.

Sinecen, M. (2011). *Developing 3 dimensional image analysis methods for characterization and classification of aggregates*. Phd. Thesis, Dokuz Eylül University, İzmir.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, *2016*, 1–11. https://doi.org/10.1155/2016/3289801

Smolensky, P. (n.d.). Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Process*, *56*(4), 194-281.

Socher, R., Lin, C. C.-Y., Ng, A. Y., & Manning, C. D. (n.d.). Parsing natural scenes and natural language with recursive neural networks. *International Conference on Machine Learning*, 8, 129-136.

Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience*, *11*(1), 126–134.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *4*(1), 1–103.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *LIX*(236), 433–460.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (n.d.). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(12), 3371-3408.

Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience, 2017*, 1–8.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016, June 1). Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*, 2285–2294.

Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Yan, S., et al. (2016). HCP: A flexible CNN framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(9), 1901–1907.

Werbos, P. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, *1*(4), 339–356.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Woods, R. E., Eddins, S. L., & Gonzalez, R. C. (2009). *Digital image processing using MATLAB (3rd ed.)*. Tennessee: Gatesmark Publishing.

Yan, Y., Zhu, Q., Shyu, M.-L., & Chen, S.-C. (2016). A classifier ensemble framework for multimedia big data classification. *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, 615–622.

Yang, C., Odvody, G. N., Fernandez, C. J., Landivar, J. A., Minzenmayer, R. R., & Nichols, R. L. (2015). Evaluating unsupervised and supervised image classification methods for mapping cotton root rot. *Precision Agriculture*, *16*(2), 201–215.

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *ArXiv preprint ArXiv:1702.01923.*

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *ArXiv preprint ArXiv:1212.5701*.

Zhang, M., Li, W., & Du, Q. (2018). Diverse region-based CNN for hyperspectral image classification. *IEEE Transactions on Image Processing*, *27*(6), 2623–2634.

Zhu, X. J. (2005). Semi-supervised learning literature survey. *Computer Science*, *1*(2), 1-28.