UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARIANA DA SILVA BARROS

Development of a Deep-Learning based System for Disease Symptoms Detection over Crop Leaves Images

Recife

2021

MARIANA DA SILVA BARROS

Development of a Deep-Learning based System for Disease Symptoms Detection over Crop Leaves Images

A M.Sc. Dissertation presented to the Center of Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Concentration Area: Computer Engineering

Advisor: Stefan Michael Blawid

Recife

2021

Catalogação na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

B277d Barros, Mariana da Silva
Development of a deep-learning based system for disease symptoms detection over crop leaves images / Mariana da Silva Barros. – 2021.
112 f.: il., fig.

Orientador: Stefan Michael Blawid.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2021.
Inclui referências.

1. Engenharia da computação. 2. Deep learning. 3. CNN. 4. Multi-label. 5. Doenças de plantas. I. Blawid, Stefan Michael (orientador). II. Título

621.39 CDD (23. ed.) UFPE - CCEN 2022 – 51

Mariana da Silva Barros

“Development of a Deep-Learning based System for Disease Symptoms Detection over Crop Leaves Images”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Engenharia da Computação

Aprovado em: 22/12/2021.

BANCA EXAMINADORA

_____

Prof. Dr. Adriano Lorena Inacio de Olivera
Centro de Informática / UFPE

_____

Profa. Dra. Mylene Christine Queiroz de Farias
Departamento de Engenharia Elétrica / UnB

_____

Prof. Dr. Stefan Michael Blawid
Centro de Informática / UFPE
(**Orientador**)

I dedicate this dissertation to my family and friends.

ACKNOWLEDGEMENTS

First of all, I want to thank God for giving me the strength to overcome my limitations and don't give up. Without Him, I would not have arrived where I am today.

During my whole life and especially throughout my academic journey, the experiences I have lived and the people I have met made me who I am today. Therefore, I want to thank everyone in my life for all the support and help I have received. I thank my parents, Edna and Antônio, for being the basis that made me arrive here. I thank my brother, Tiago, and my boyfriend, Marcos, for all the encouragement given when I needed it.

I would like to thank my advisor, Prof. Stefan Blawid, for guidance and support during the development of this work. I also thank the Informatics Center (CIn) for providing the education, opportunities, and infrastructure that allows the formation of so many people. I also would like to thank the professors and students from CliFiPe and from the Phytopathology Department from UFRPE, and especially Prof. Rosana Blawid, for the support and partnership in the research developed.

More important than where we arrive at the end of a path are the friendships that we make during the journey. I would like to thank my friends Ladson Gomes, Igor Moura, Antônio Netto, Lais Bandeira, Gabriela Alves, Júlia Feitosa, Carlos Pena, Lucas Cavalcanti, Thiago Moura, and so many others for being part of my life. In particular, I want to say thank you for the members from Cultiv.aí for working in this project along with me.

I would also like to thank especially every member from the RobôCIn and the E.S.T.U.F.A., for sharing their routines, hopes, and dreams with me, and for helping me every day to become a better person.

ABSTRACT

Family farming represents a critical segment of Brazilian agriculture, involving more than 5 million properties and generating 74% of rural jobs in the country. Yield losses caused by crop diseases and pests can be devastating for small-scale producers. However, successful disease control requires correct identification, which challenges smallholders, who often lack technical assistance. The present work proposes a system that alerts smallholder farmers and phytopathology experts about possible crop disease outbreaks, enabling a faster diagnosis and intervention. To this extent, we detect disease symptoms in images of plant leaves taken by farmers directly in the field using a mobile phone app developed for this purpose. The implemented module is part of a service platform connecting producers and experts, designed in partnership with phytopathology professionals from the Federal Rural University of Pernambuco (UFRPE). The work uses deep learning and Convolutional Neural Networks to perform the image classification. The classification experiments were applied over a dataset composed of leaf images of grape crops cultivated in the state of Pernambuco, whose image collection was also part of the present work. Therefore, pictures taken under field conditions sometimes present low quality, which decreases the classification performance. Thus, we also classify the images regarding their quality, to exclude challenging images from disease detection and reduce the number of erroneously classified images entering the database. The multi-label technique is employed in this scenario, enabling a single neural network model to classify whether a leaf picture reveals crop disease symptoms and whether they present good enough quality to do so reliably. The multi-label mechanism is also a promising approach to include additional picture properties in the future, like disease agents. The developed classification system achieves a recall value of 97.6% for symptom detection and a precision value of 94.8% for image quality classification.

Keywords: deep learning; CNN; multi-label; crop disease; symptoms detection; family farming.

RESUMO

Agricultura familiar representa um segmento crítico da agricultura brasileira, envolvendo mais de 5 milhões de propriedades e gerando 74% dos empregos rurais no país. As perdas de rendimento causadas por pragas e doenças na colheita podem ser devastadoras para os pequenos produtores. No entanto, o controle de doenças bem-sucedido requer uma classificação correta, o que desafia os pequenos proprietários, que muitas vezes carecem de assistência técnica. O presente trabalho propõe um sistema que alerta pequenos produtores e especialistas em fitopatologia sobre possíveis surtos de doenças em plantas, permitindo um diagnóstico e intervenção mais rápidos. Nesse sentido, nós detectamos sintomas de doenças em imagens de folhas de plantas tiradas diretamente por agricultores no campo usando um aplicativo de celular desenvolvido com esse propósito. O módulo implementado é parte de uma plataforma de serviços que conecta produtores e especialistas, projetado em parceria com profissionais de fitopatologia da Universidade Federal Rural de Pernambuco (UFRPE). O trabalho usa aprendizagem profunda ("deep learning") e redes neurais convolucionais (CNNs) para realizar a classificação das imagens. Os experimentos de classificação foram aplicados sobre um conjunto de dados composto por imagens de folhas de videira cultivadas no estado de Pernambuco, cuja coleta também foi parte do presente trabalho. Portanto, algumas imagens coletadas sob as condições do campo apresentam baixa qualidade, o que diminui o desempenho da classificação. Assim, nós também classificamos as imagens com relação à sua qualidade, para excluir imagens desafiadoras da detecção de doenças e reduzir o número de fotos classificadas erroneamente entrando na base de dados. A técnica de "multi-label" é aplicada neste cenário, permitindo a um único modelo classificar se as imagens mostram sintomas e se elas apresentam qualidade suficiente para que isso seja feito de maneira confiável. O mecanismo "multi-label" também é uma abordagem promissora para incluir futuramente propriedades adicionais da imagem, como agentes causadores de doenças. O sistema de classificação desenvolvido alcança um valor de "recall" de 97.6% para detecção de sintomas e um valor de precisão de 94.8% para classificação de qualidade das imagens.

Palavras-chave: deep learning; CNN; multi-label; doenças de plantas; detecção de sintomas; agricultura familiar.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BR	Binary Relevance
CliFiPe	Clínica Fitossanitária de Pernambuco
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FN	False Negatives
FP	False Positives
GDP	Gross Domestic Product
GPU	Graphics Processing Unit
IoT	Internet of Things
k-NN	K-Nearest Neighbours
LP	Label Powerset
mAP	Mean Average Precision
ML	Machine Learning
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Units
RL	Reinforcement Learning
RPN	Region Proposal Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TanH	Hyperbolic Tangent
TN	True Negatives
TNR	True Negative Rate
TP	True Positives

TPR True Positive Rate

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

The agricultural sector in Brazil plays an essential role in the nation's income generation. In 2012, it was responsible for 4.45% of the country Gross Domestic Product (GDP), reaching US$100.1 billion, according to Oliveira et al. (2014). Generally, agribusiness is associated with large companies. However, a considerable part of Brazilian production comes from smallholder producers classified as family farmers.

According to Sampaio & Vital (2020), family farms comprise establishments that are characterized by: (i) A workforce recruited predominantly from the family, (ii) A family income mainly originating from activities related to the establishment itself, (iii) An establishment administered by the producer or his family, and (iv) An area not superior to four fiscal modules (one fiscal module is equivalent to 5 acres, as stated in the Forest Code by Embrapa (2012 (accessed December 1st, 2021))). In addition, another condition can complement this definition, according to Gasson et al. (2008): A business management predominantly familiar.

There are currently about 5 million properties in Brazil that subsist on family farming, employing more than 10 million people. According to the Brazilian government (IBGE, 2017 (accessed March 1, 2021)), smallholder farmers are responsible for 74% of rural jobs and 33.2% of the agricultural GDP. The 2017 IBGE Agricultural Census stated that, although it occupies only 33.49% of the cultivated land, family farms constitute 91.42% of the agrarian establishments, employing 80.91% of the rural population. In addition, Sampaio & Vital (2020) states that, for the last years, the data has shown that family farming in the country has been developing robustness.

About 25% of farm products come from family farming in the Northeast region of Brazil. In Pernambuco, located in this region, family farming employs ca. 83% of the people living in the countryside. Although it occupies only 47% of the area, it contributes with 45% of the income from the state's rural establishments, as states IBGE (2017 (accessed March 1, 2021)). Furthermore, the relative land area for permanent crops occupied by family farming in Pernambuco is larger compared to the one in the whole country, resulting in more than 55% of the cultivated land (SAMPAIO; VITAL, 2020).

Despite the great importance of family farming, small-scale producers face several challenges and are economically vulnerable, especially in a region that still demonstrates one of the

highest poverty rates in the country. Pests and diseases, for example, are two of the biggest problems in agriculture and are responsible for losses that reached up to 43% of the annual production in 2016 (IBGE, 2017 (accessed March 1, 2021)). Several studies point to them as one of the most significant factors of reduced productivity in any crop species, as stated by Oliveira et al. (2014). Among the causes of pests and diseases is the misuse (or lack of use) of phytosanitary products for control and fertilizers (OERKE; DEHNE, 2004). The crop losses can occur both in the field and during storage, and the damage depends on several factors related to environmental conditions, the plant species, the socioeconomic conditions of farmers, and the level of technology used (OLIVEIRA et al., 2014).

This situation aggravates due to the aging and the low educational level of small-scale producers. 15.6% of the farmers that used phytosanitary products do not know how to read or write (IBGE, 2017 (accessed March 1, 2021)). In addition, they lack the financial resources to seek private agricultural consultants, and governmental support is sparse. Producers trying to insert themselves into agribusiness are specifically negatively impacted.

According to the Agricultural Census in 2006 and 2017 (IBGE, 2017 (accessed March 1, 2021)), about 80% of the agricultural establishments declared not having received any technical support required to adopt modern technology and improve productivity. In this scenario, technical assistance's public and private services play a crucial role for small-scale producers. These services inform them about management and technological innovations and empower them to adopt these innovations correctly, reducing the risks inherent to agricultural activity. Extensionists, i.e., technical assistants employed by the government, could assist farmers. However, only a few extensionists are available in Brazil, compared to the number of small-scale properties. The ratio amounts to about one extensionist to 150 small farmers. On the other hand, small-scale producers cannot afford private assistance.

1.2 PROBLEM

Given the situation presented in the previous section, the correct disease identification is vital for any treatment strategy that explicitly targets the causing agents, avoiding the excessive use of non-specified agrochemicals. Furthermore, accurate detection allows the producer to take prevention and treatment measures while the crop is still in the field (pre-harvest). It will also contribute to using the correct phytosanitary product and dosage to solve the problem causing minimal harm to the crop and the people.

Traditionally, crop inspection has been performed visually by experts in the field (BARBEDO, 2018). However, diagnosis is demanding even for trained pathologists and requires the analysis of disease symptoms and distribution, cultural practices, and environmental data. In addition, trained pathologists receive multiple demands and are not always available to execute the inspection, especially in poor and isolated areas.

Over the years, with the technology evolution, several works have been developed aiming to help automate agriculture-related tasks using artificial intelligence. According to Albanese, Nardello & Brunelli (2021), farmers and researchers have been teaming up to create intelligent systems for precision agriculture, which includes, for example, smart sensors networks and systems for monitoring resources usage. Furthermore, the advancement of machine learning and deep learning techniques enabled the development of various studies using image classification to identify diseases over plant images. One of the groundbreaking approaches was the one developed by Mohanty & Salathé (2016), which created a dataset composed of plant leaves images taken in a laboratory, exposing different crops and diseases.

However, even with the development of image classification techniques, images may present some characteristics that make it challenging to classify the diseases, as explored by Barbedo (2018). A single plant leaf, for instance, can reveal symptoms of multiple diseases, making it difficult for the classifier to perform its task. In other cases, the disease symptoms might not appear uniformly over the whole leaf but only as localized small spots. Furthermore, environmental conditions can interfere with classifying pictures taken in the field. Such hampering factors include a noisy background, sunlight, shadows, or images out of focus, for example. Image 1 displays some field pictures presenting some of these challenges.

Figure 1 – Examples of challenging field leaf images for disease classification

Source: the author

Nowadays, some image datasets are available for training models in similar applications, as,

for example, the PlantVillage dataset (created by Mohanty (2016 (accessed March 1, 2021))) and the Digipathos dataset (created by Embrapa (2014 (accessed August 31, 2021))). However, they include mainly pictures taken in laboratories, not reflecting the field conditions. Therefore

The present work proposes a deep-learning assistant system to identify pictures of plant leaves that reveal disease symptoms. The creation of the dataset used in the experiments involved the images collection in grape plantations in the countryside of the state of Pernambuco. The assistant was developed in partnership with experts from the Clínica Fitossanitária de Pernambuco (CliFiPe), or Phytosanitary Clinic of Pernambuco, hosted by the Federal Rural University of Pernambuco (UFRPE). The goal is to provide technical aid in plant disease diagnosis, prevention, and treatment more efficiently to family farmers. Furthermore, the efficiency gain shall enable the experts from CliFiPe to reach out to a more significant part of the local farming community. The neural network was trained with grape leaves images; however, the same techniques and principles can also be applied to other crops. The system can classify the grape leaves' images, achieving a recall value of over 97%.

1.3 OBJECTIVES

The main objective of this work is to implement a system capable of identifying diseases and pests over plant images. The dissertation has the following specific goals:

- To implement a system that detects whether a given leaf image taken in the field shows disease symptoms or not. The system must achieve a classification rate of at least 90% for pictures revealing visible disease symptoms.
- To identify the quality of the photo taken by the farmer in the field, to decide if it is enough to perform the diagnosis. The system must achieve a classification rate of at least 85% in the cases where the image has poor quality.
- To create an image dataset capable of performing the Neural Network training, composed of pictures of grape leaves cultivated in the state of Pernambuco.
- To enable the integration of one or more new datasets to either retrain or fine-tune the Neural Network model improving classification performance.
- To develop a strategy that extends the system to detect the disease-causing agent in the pictures. This work must compare possible approaches that will allow the system to

identify which class of agents (fungi, virus, or bacteria) caused the symptoms displayed in the picture when revealed.

Additionally, the project in partnership with the clinic, that comprises the system, also presents some objectives for the future. It aims to distribute a mobile phone application that allows the farmers to take pictures of the crop leaves and displays the classification results. Moreover, once the image dataset created in this work is sufficiently diverse, it will become available to other researchers in the domain.

1.3.1 Contributions

This work presents contributions to disease prevention and mobile services in agriculture.

- It creates a dataset that will be made public for enabling related research.
- It initiates the distribution of the developed mobile application to professors, agronomists, and farmers.
- It allows the expert to offer more direct assistance to farmers and improve the communication between them.
- It informs the farmer on disease outbreaks to take the related treatment measures.
- It improves a platform to allow experts to store information, view model's results and change them, and communicate with farmers.

1.4 WORK STRUCTURE

The rest of this dissertation is organized as follows: Chapter 2 introduces essential concepts related to the domain where the work is inserted and the techniques used in its development. Chapter 3 presents related works in the literature that employ similar ideas and concepts to the ones used in this work. Chapter 4 describes the proposed approach to the general system and delineates where is inserted the present research. Chapter 5 explains in detail the methods and approaches used in the implemented experiments. Chapter 6 informs and discusses the obtained results. Finally, Chapter 7 concludes the dissertation and indicates future works.

2 THEORETICAL BACKGROUND

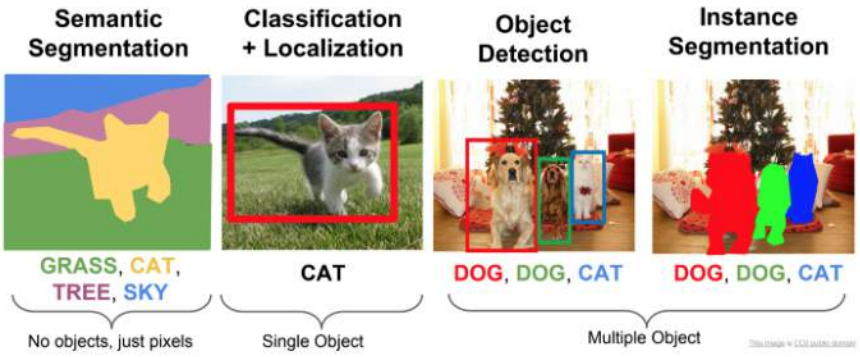
In this chapter, some concepts will be discussed in order to establish the theoretical foundation related to the developed research. First, Section 2.1 introduces the use of machine learning to perform image classification. Then, Section 2.2 details a neural network and inserts it in the deep learning domain. Section 2.3 presents a CNN and describes its parts. Section 2.4 explains the process of a CNN training. Section 2.5 shows the state-of-the-art CNN techniques used for image classification. Section 2.6 describes multi-label classification and inserts it in this work. Finally, Section 2.7 displays the evaluation metrics used in classification experiments.

2.1 MACHINE LEARNING AND IMAGE CLASSIFICATION

According to Sze et al. (2017), Artificial Intelligence (AI) is the science and engineering of creating intelligent machines that can achieve goals as humans do. Inside the AI domain, Machine Learning (ML) is the field of study that gives computers the ability to learn without being explicitly programmed. Theodoridis (2015) describes it as the use of a machine or computer to learn in analogy to how the brain learns and predicts.

Today, the use of Machine Learning is getting more and more common in people's lives. The applications from this domain are numerous, but one crucial example is object recognition, which describes a set of Computer Vision related tasks that involve identifying objects in images. Figure 2 shows some tasks related to object recognition.

Figure 2 – Object recognition related tasks



Source: (AGARWAL, 2019)

According to Agarwal (2019), among them, it is possible to distinguish three main tasks: image classification, object detection, and object segmentation.

1. Image Classification (IC): Given an image with a single object, it aims to recognize semantic categories of objects in it.

2. Object Detection (OD): A general case of the problem involving classification and localization, when the number of objects is unknown. OD locates objects in the image with a bounding box (a box around the detected object) and classifies each one of them.

3. Object Segmentation (OS): Image partitioning and understanding to what object each segment belongs. OS identifies each pixel from the recognized objects present in the image and assigns a category label.

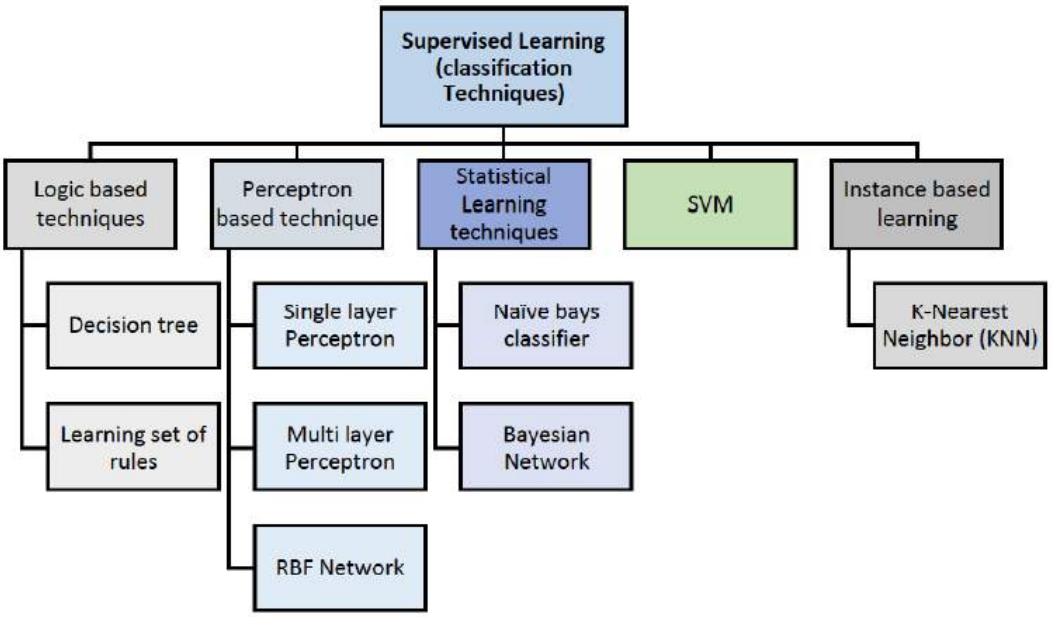
In general, the goal in classification is to assign an unknown pattern to one out of several established classes (THEODORIDIS, 2015). In other words, it is the task of predicting a category for a given object. Image classification refers to assigning a label to an image, as stated by Abraham et al. (2021). A system classifying input images will return to which class the image belongs.

With the development of researches and studies in the area, image classification has been broadly used in many different domains, as, for instance, education, security, health, commerce, and agriculture. Some of the numerous applications of image classification include handwriting recognition, face detection, scene parsing, vision for autonomous driving, hand gesture recognition, and diseases identification (RAWAT; WANG, 2017).

Pardede et al. (2020) declares that the methods and techniques for image classification can be grouped based on the depth of methods: shallow architectures and deep architectures. The former include traditional machine learning methods, and some of the most used are presented in Figure 3 by Soofi & Awan (2017).

Some of the most common examples of shallow architectures are Support Vector Machine (SVM) (BOSER; GUYON; VAPNIK, 1992), Naïve Bayes (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997), and K-Nearest Neighbours (k-NN) (FIX; HODGES, 1989). Deep learning methods involve mainly CNNs that will be described in more detail in the next sections.

Figure 3 – Supervised learning classification techniques



Source: (SOOFI; AWAN, 2017)

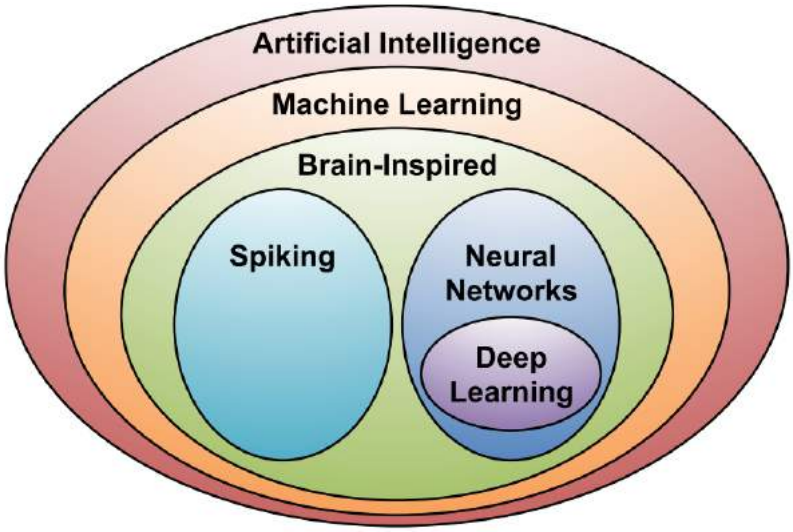
2.2 NEURAL NETWORKS AND DEEP LEARNING

In Section 2.1, it is presented the definitions of Artificial Intelligence and Machine Learning. Sze et al. (2017) continues his explanation on the subject. According to the authors, a sub-field of machine learning is brain-inspired computation, where the way the brain works inspires some aspects of its basic form and functionality. Inside that, there is the area of spiking computing and, on the other side, the neural networks field. Figure 4 shows how neural networks and deep learning are included in the field of Artificial Intelligence, according to Sze et al. (2017).

As the name suggests, the biological nervous system operation inspired artificial neural networks. Neurons interconnect via functional links in natural neural networks, called "synapses". Synapses can be either activated or inhibited, and they mediate information between connected neurons in a hierarchically structured way, as stated by Theodoridis (2015).

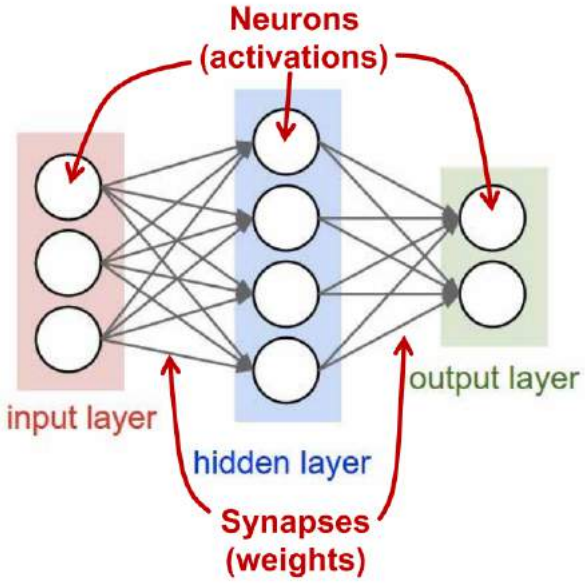
It was the work of McCulloch & Pitts (1943) that first developed a computational model for a neuron. Later, Rosenblatt (1958) built a learning machine based on the neuron model that learns from a set of training data. This machine is called a perceptron and is the kick-off point to artificial neural networks. According to Theodoridis (2015), neural networks are learning machines comprising a large number of neurons connected in a layered fashion. Figure 5 exemplifies a neural network system.

Figure 4 – Relationship between Deep Learning and Artificial Intelligence



Source: (SZE et al., 2017)

Figure 5 – Neural Network system



Source: (SZE et al., 2017)

A Multi-Layer Perceptron (MLP) is a simple neural network composed of more than one layer of perceptrons. The neurons in the input layer receive the input values and propagate them to the middle layer or the "hidden layer" of the network. Each value is associated with a "weight", and each neuron's computation involves a weighted sum of the input values. The network propagates these weighted values through the hidden layers until the output layer, and the output layer will present the network outputs to the user, as states Sze et al. (2017).

For many years, shallow neural network models with few stages were the leading choice

for brain-inspired computing. However, by the early 1990s, deep neural networks (composed of many layers) had become an explicit research subject and have been vastly explored (SCHMIDHUBER, 2015). Several studies define deep learning as the area inside the neural networks where there are more than three layers (more than one hidden layer). The number of layers in a Deep Neural Network (DNN) is usually between 5 and 1000, according to Sze et al. (2017). The learning process in a DNN, also called training, involves determining the value of the weights in the network. Inference runs the network model with given weight factors.

In a multi-layer feed-forward neural network, every node in a layer connects to every other node in the neighboring layer (SVOZIL; KVASNICKA; POSPICHAL, 1997). The input values propagate through the layers, and the neurons connect to each other through weights. Each neuron receives information from the precedent neurons and produces an output by passing the weighted sum of those signals through an activation function (SAZLI, 2006). The name feed-forward network indicates that information flows forward from the input to the output layer (THEODORIDIS, 2015).

The optimization process used in network training is called "gradient descent". In this process, the partial derivative of the loss related to each weight determines the updated weight value, as explained by Theodoridis (2015). The process is repeated at each iteration to reduce the overall loss. The algorithm uses backpropagation to compute the partial derivatives of the gradients, i.e., weight values are passed backward through the network to compute the weight-dependent loss function. The difference between the actual and the desired network output defines the loss. Thus, the latter must be available for training the network, implying a supervised learning technique (SAZLI, 2006). A series of methods as, for example, batch, supervised learning, reinforcement learning, and fine-tuning, improve the training performance and efficiency.

According to Svozil, Kvasnicka & Pospichal (1997), a multi-layer feed-forward neural network can operate in two modes: training and prediction. The training process adjusts the weight factors at each iteration to reduce the error (loss), starting from initial arbitrary values. Each iteration is called an epoch, and, usually, numerous epochs are necessary to complete the training. The weights will converge to a set of values considered the local optimum as the iterative process continues.

During prediction, the model receives an image as input and outputs a vector of scores, one for each class. The vector contents indicate the probability of the object belonging to that class. Usually, the highest score represents the most likely class, according to Sze et al.

(2017). Also, we can describe the loss as the gap between the correct ideal scores and the scores computed by the DNN based on its current weights. Therefore, the DNN training goal is to determine the weight factors that maximize the correct class score or that minimize the average loss over an extensive training set. The resulting error is an estimate of the quality of prediction of the trained network (SVOZIL; KVASNICKA; POSPICHAL, 1997).

According to Voulodimos et al. (2018), in the last decade, there has been a series of developments in deep architectures and deep learning algorithms. Among the factors that contributed to these improvements is the appearance of large and publicly-available datasets empowered by Graphics Processing Unit (GPU) computing. In addition, new regularization techniques and powerful frameworks accelerated the deep-learning revolution.

Nowadays, some of the applications that use deep learning include health care, visual data processing, social network analysis, and audio and speech processing (HASAN; YUSUF; ALZUBAIDI, 2020). In addition, deep learning techniques achieve good performance in various computer vision problems, like object detection, motion tracking, action recognition, human pose estimation, and semantic segmentation, as states Voulodimos et al. (2018). One of the most relevant types of deep learning models for computer vision and image applications are the CNNs, to which we turn in the following section.

2.3 CONVOLUTIONAL NEURAL NETWORKS

Applications in image classification and object detection increased with the development and improvement of deep learning algorithms. According to Pathak, Pandey & Rautaray (2018), object detection methods using deep learning techniques based on CNN have been extensively applied. This type of neural network performs well when processing data that come in the form of multiple arrays, as many media data modalities are, as described by LeCun, Bengio & Hinton (2015).

In the 1980s, studies in Neuroscience concluded that the brain has different regions for distinct tasks. Therefore, the brain has a hierarchic and localized organization. Fukushima (1988) published the first computational model based on the human brain and its local connectivities. However, the term CNN came into use only in the 1990s, with the research of Lecun et al. (1998), which involved using a neural network to recognize characters in images.

A CNN is a deep learning algorithm that takes an image as input, assigns learnable weights to various objects in it, and can differentiate one from another. In a CNN, only the last layer

is fully connected, whereas in an Artificial Neural Network (ANN), each neuron is connected to every other neuron (A. et al., 2019). CNNs are composed of multiple convolutional layers. The network generates a feature map in each layer consisting of an increasingly higher-level abstraction of the input data that preserves essential yet unique information.

By definition, a CNN uses a single network to learn several features of a given image and perform its classification, tasks previously performed separately. The visual system's structure inspired this idea. Today, CNNs achieve very good results in pattern recognition, as stated in Voulodimos et al. (2018).

The principal components of a CNN are convolutional layer, activation function, subsampling layer, and fully-connected layer. A typical CNN architecture consists of repetitions of sequences of several convolutional layers and a pooling layer, followed by one or more fully-connected layers (YAMASHITA et al., 2018). We describe these components in the following sections.

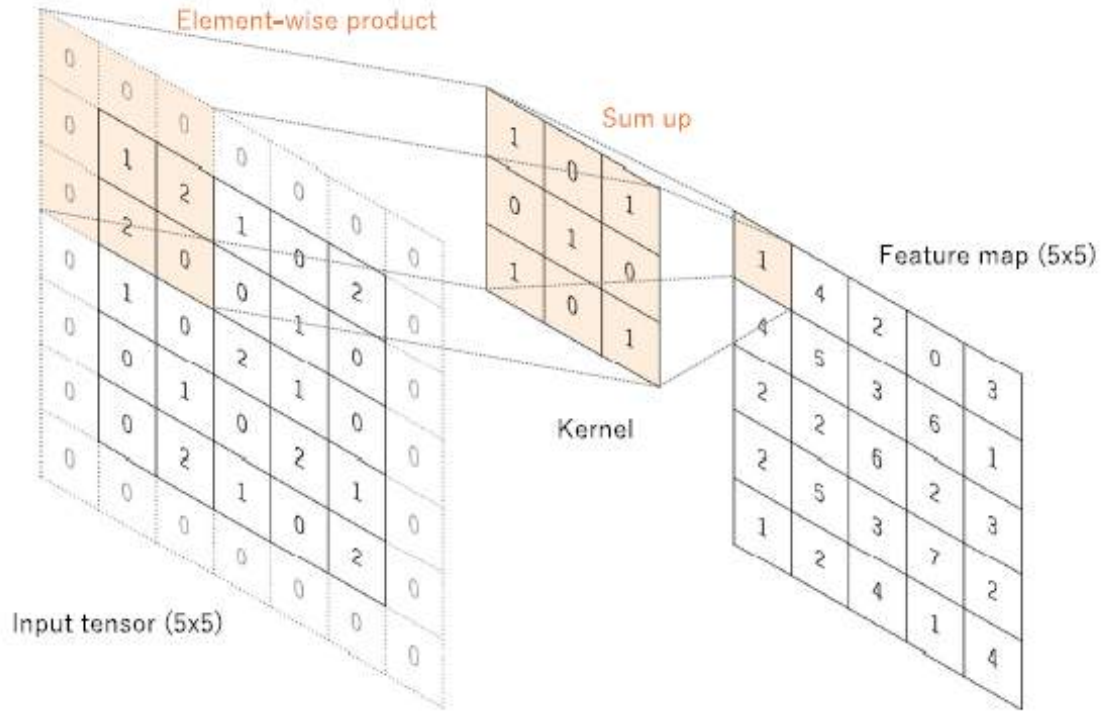
2.3.1 Convolutional Layer

In a CNN, each convolutional layer is responsible for performing feature extraction and generating a successively higher-level abstraction of the input data, called "feature map". This abstraction preserves essential yet unique information about the input image, as affirmed by Sze et al. (2017). The feature extraction consists of a combination of linear and non-linear operations, as, for example, convolution and activation functions (YAMASHITA et al., 2018).

The convolutional layer in a CNN, as opposed to MLP networks, preserves the spatial structure of the input image. According to Yamashita et al. (2018), it receives the input as a three-dimensional matrix and then convolves a three-dimensional filter (also called a kernel) with the image (called a tensor). The filter slides over the image spatially, computing element-wise products. The operation performed by the filter at each position is a matrix convolution between the input image and the filter, and, as output, it produces an activation map that is composed of the image features. Figure 6 shows an illustrative example of a convolution operation.

Each convolutional layer from a CNN includes several filters, and each filter will generate a different feature map, learning a specific attribute from the image (YAMASHITA et al., 2018). The filter depth must be equivalent to the current input depth. As CNNs are composed of various convolutional layers, and each layer includes several filters, many feature maps combine

Figure 6 – A convolution operation



Source: (YAMASHITA et al., 2018)

to classify an image (RAWAT; WANG, 2017).

Because of its behavior, the characteristics learned by the filter are robust to translation, as states LeCun, Bengio & Hinton (2015). In other words, the filter, which acts as a pattern detector, can identify these patterns in any image location. A learned filter applies to any network location since the parameters are shared.

The basic parameters from each convolutional layer are the filters size, the number of filters, the padding, and the stride (YAMASHITA et al., 2018). The padding is related to the image's edge size, and the stride is the displacement that the filter will move each time it slides through the image. The convolution in Figure 6, for example, has zero padding and a stride of 1. Another possible parameter is the expansion rate. An expansion rate exceeding one implies a dilated kernel, which causes the filter to lose its location characteristic to some degree.

2.3.2 Activation Function

After the convolution layer, an activation function is added for the network to module non-linearities, according to Yamashita et al. (2018). Depending on the function, it will scan

the network and allow some values to be replicated. In other words, the pixels which are not necessary will be deactivated, and only the essential pixels are kept (A. et al., 2019).

Among the most used activation functions, there are: binary step, Sigmoid, Hyperbolic Tangent (TanH), Rectified Linear Units (ReLU), and Softmax. Each one will generate a different output activation map.

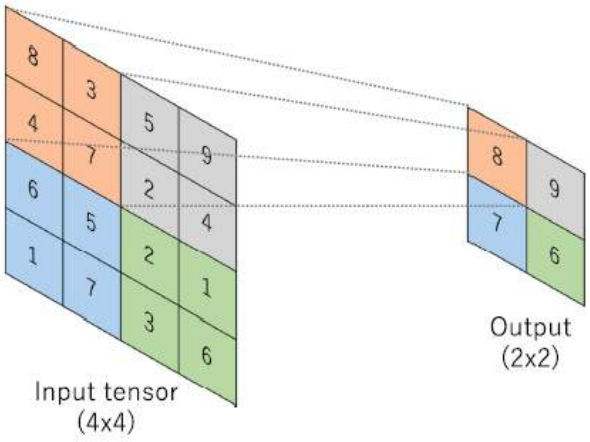
2.3.3 Subsampling Layer

LeCun, Bengio & Hinton (2015) states that the subsampling layer, also called the pooling or downsampling layer, is responsible for generating smaller representations from the feature maps produced in the previous layers, to create a representation with smaller computational cost and work against overfitting. According to Voulodimos et al. (2018), the pooling layer reduces the spatial dimensions of the input volume for the next convolutional layer. In addition, it also decreases the number of learnable parameters (YAMASHITA et al., 2018). Even though it does not affect the depth dimension of the volume, it leads to a certain loss of information (VOULODIMOS et al., 2018). However, another characteristic of this layer is that it reduces the sensibility to small distortions in the image.

The main used strategies to subsampling are:

- Max Pooling: Replicates the maximum value from the pool size.
- Average Pooling: Generates the average value among the ones present in the pool size.

Figure 7 – An example of max pooling operation



Source: (YAMASHITA et al., 2018)

Figure 7 shows an example of a max pooling operation with a filter size of 2x2, no padding, and a stride of 2.

2.3.4 Fully-Connected Layer

The output layer from a CNN is a basic neural network fully-connected layer, where all neurons are fully connected to all the neurons in the previous layer (A. et al., 2019). As stated by Voulodimos et al. (2018), this type of layer performs the high-level reasoning in the neural network. The output layer converts two-dimensional feature maps into one-dimensional vectors. These feature vectors constitute the direct classification result or allow further processing. The main idea of this layer is the same one used in a MLP. It comprises a classifier and a computing unit for calculating loss function, acting as an output layer (CUI, 2018).

In a general overview, a CNN combines all these components to optimize image processing and classification (LECUN; BENGIO; HINTON, 2015). Each layer may appear several times and may be combined in many different ways, depending on the application. Therefore, a specific architecture exists for any task, enabling feature extraction and classification. The training of the CNN determines the weights from the fully-connected layers, the biases from the activation functions, and the filters from the convolutional layers.

2.4 CNN TRAINING

As mentioned in Section 2.3, the CNN training is the process that determines some values and parameters involved in the neural network, namely the filters from the convolutional layers (described in 2.3.1), the biases from the activation function (explained in 2.3.2), and the weights from the fully-connected layers (described in 2.3.4). The learning of these values happens through a process that involves multiple repetitions from the whole CNN sequence and the loss calculation after each one of them. The loss is propagated from the last to the initial layer (the reason why it is called back-propagation), and the weights and parameters are adjusted depending on the loss value to minimize it (YAMASHITA et al., 2018).

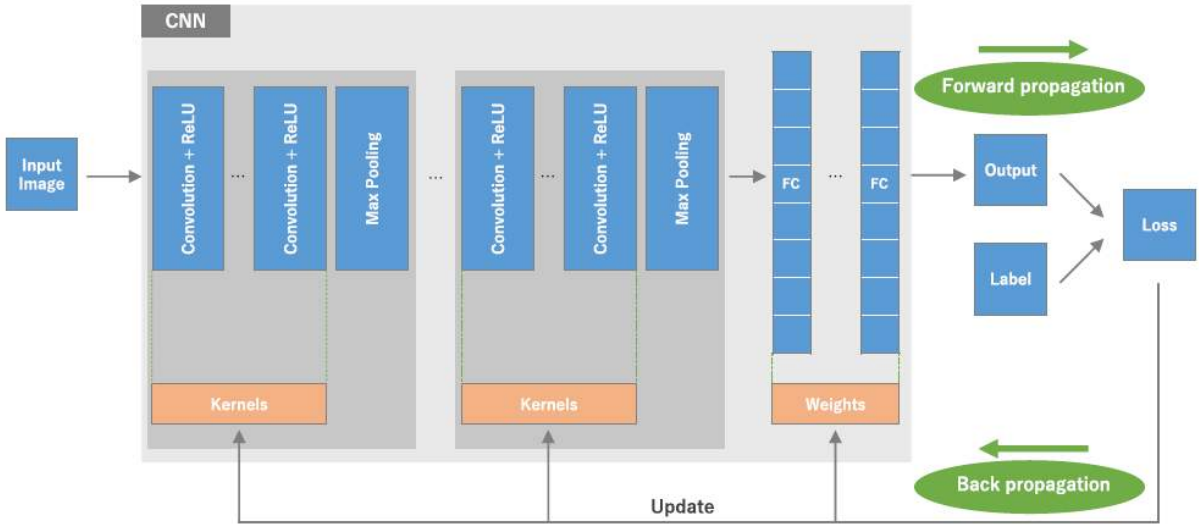
Nowadays, there are various algorithms to optimize this training. One of the most known and used is the Mini-Batch Gradient Descent, described in Li et al. (2014). The initial values of the parameters to be learned are randomly determined. First, it selects N data samples (in this case, N images), where N corresponds to the associated batch size. Then, the mini-batch

is propagated through the CNN to calculate the training loss. Gradients are calculated in the next step via retro-propagation through the network. Finally, the parameters are updated using the gradients. The process repeats for all mini-batches, and then the whole process is repeated for a determined number of times (called epochs), or until the loss reaches a minimum value, indicating that the CNN has achieved a good performance.

Different possible optimization algorithms may substitute the frequently used Stochastic Gradient Descent (SGD). Some examples are: SGD using also the Momentum (another hyper-parameter from the CNN), RMSProp or Adam. The choice of the algorithm depends on the application and the situation.

Figure 8 represents the complete training flow of a basic CNN, as described in Yamashita et al. (2018). The input image initially goes through various blocks, each one composed by a sequence of convolutional layers and activation functions (in this case, represented by ReLU function), followed by a max-pooling layer. Then, the generated feature maps go through a series of fully-connected layers, generating the output from the network. This output is, in turn, compared with the ground-truth label for this data, and the calculated loss is back-propagated through the network to update the weights and the convolution kernels. The order and number of layers from each type will vary, depending on the chosen CNN architecture. In the following section, we will discuss some considerations about the training in a CNN.

Figure 8 – CNN Training Process



Source: (YAMASHITA et al., 2018)

2.4.1 Stages in CNN Training

The first step in the training process is the data pre-processing. There are various manners to pre-process the input data before the training process. One strategy to offer greater flexibility to the model and make it less sensitive to changes in weights is to normalize it (PATRO; SAHU, 2015). This step ensures that each input parameter has a similar distribution, which allows the algorithm to work more in the central data region from the N-dimensional space. In addition, the centralized data simplifies the representation from this information and minimizes the impact in the separation frontier.

The next step takes the pre-processed dataset and splits it into subsets, each one with a different role in the CNN training process. According to Yamashita et al. (2018), typically, the available data is split into three sets:

- The training set is needed to train the network. Forward propagation computes the loss, and back-propagation updates the learnable parameters.
- The validation set is needed to evaluate the model during the training process and fine-tune the hyperparameters.
- The test set allows to evaluate the performance of the final model after training.

The initialization of the weights associated with each neuron in the network, as described in Section 2.3, follows the data preparation for training. As described by Narkhede, Bartakke & Sutaone (2021), there are two possible strategies for this process: perform the initialization with new weights, or use the weight values from a pre-trained model (transfer learning technique). The first type of initialization can generate random weight values, perform a data-driven initialization, or a hybrid one, that combines the two methods. The initialization using pre-trained weights, also called transfer learning, improves the generalization by learning quality features from the data (ERHAN et al., 2010). The idea is that, from a model trained in a more generic database, for example, ImageNet (DENG et al., 2009), we can use their weights in different ways, depending on the dataset size. The possibilities include training (i) the entire model (both convolutional and fully-connected layers), (ii) some convolutional layers (leaving others frozen), or (iii) the fully-connected layers freezing the convolutional base.

Next, the neural network model uses the back-propagation algorithm, as explained in Section 2.4. According to Yamashita et al. (2018), the main objective at this point is to find

kernels in convolution layers and weights in fully-connected layers, which minimize differences between output predictions and given ground-truth labels on a training dataset. The particular values of convolution kernels and weights determine the model performance. Forward propagation generates a loss function on a training dataset, and an optimization algorithm updates the learnable parameters according to the loss value. For binary classifications, one of the most used loss functions is the binary cross-entropy loss (RUBY; YENDAPALLI, 2020), a particular class of cross-entropy loss where the two prediction targets are 0 and 1. The most common loss function in multi-class classification problems is the categorical cross-entropy loss (described in Koidl (2013)), which measures the dissimilarity between the true and predicted label distribution.

As expressed by Rawat & Wang (2017), a commonly experienced problem when performing image classification is overfitting, which means that the model achieves a poor performance on a held-out test set after training. Overfitting implies that the model did not learn the ability to generalize on unseen data. One technique to recognize overfitting is to monitor the loss and an evaluation metric on the training and validation sets (YAMASHITA et al., 2018). This routine will check if the model performs too well on the training set compared to the validation set, indicating that overfitting has occurred.

Some strategies help to mitigate this problem. One of the most common ones is to perform data augmentation, as affirmed by Krizhevsky, Sutskever & Hinton (2012). This approach artificially enlarges the dataset using label-preserving transformations. Such transformations include image translations and horizontal reflections or intensity alterations of the RGB channels in the training images.

2.5 CNNS AND IMAGE CLASSIFICATION

Since the early 2000s, researchers have successfully applied CNNs for the detection, segmentation, and recognition of objects and regions in images (LECUN; BENGIO; HINTON, 2015). Some of the most common tasks include traffic sign recognition, segmentation of biological images, and detection of faces, pedestrians, and human bodies in natural images. However, it was only at the ImageNet competition in 2012 that the use of CNNs grew exponentially. Training sets composed of millions of images combined with the efficient GPUs and the development of new approaches caused a revolution in computer vision.

Nowadays, most works and studies that use CNNs for image classification apply the transfer

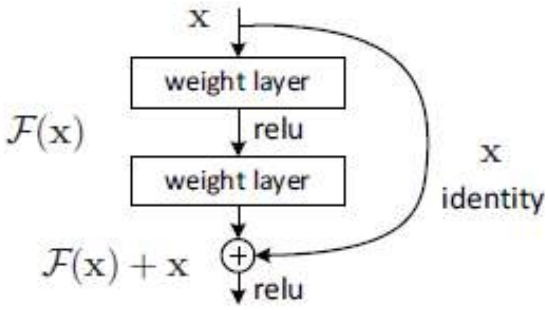
learning technique, using pre-trained models over the ImageNet dataset. Between the most famous of these architectures, we can cite AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), GoogLeNet (SZEGEDY et al., 2015), MobileNet (HOWARD et al., 2017), VGG-Net (SIMONYAN; ZISSERMAN, 2015) and Inception-V3 (SZEGEDY et al., 2016). However, since this work employs the ResNet architecture, we will describe it in more detail in the following section.

2.5.1 ResNet Architeture

He et al. (2016a) describe the first developed ResNet architecture, in 2016. With the evolution of CNN techniques, other versions of this model have been implemented. One of the most used ones is the ResNet50V2, explained in He et al. (2016b).

The ResNet architecture introduces a deep residual learning framework that addresses the degradation problem by explicitly letting the stacked layers fit a residual mapping. For instance, Figure 9 shows an example of a building block in the residual network that is essentially a feed-forward neural network with "shortcut connections". Since these connections perform identity mapping, they do not add extra parameters or computational complexity. The strategy allows to implement and train them in the same way as the previous ones.

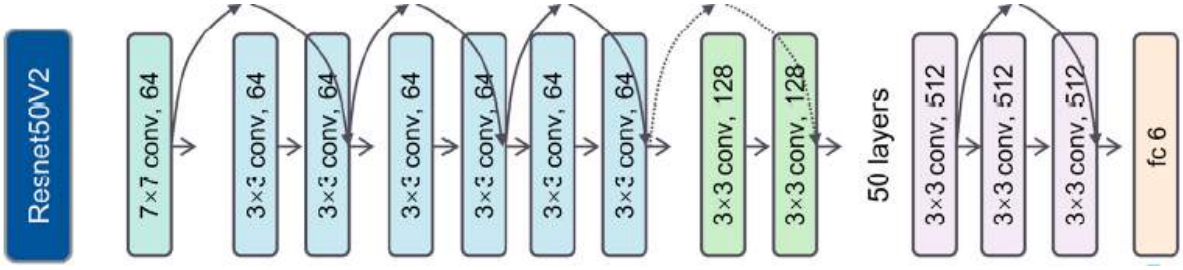
Figure 9 – Residual learning: a building block



Source: (HE et al., 2016a)

The idea behind ResNet50V2 goes even further. According to He et al. (2016b), it creates a direct path for propagating information, not only within a residual unit but through the entire network. Figure 10 displays the ResNet50V2 architecture, as shown by Liao et al. (2021).

Figure 10 – Architectures of typical deep neural networks: ResNet50V2



Source: (LIAO et al., 2021)

2.6 MULTI-LABEL CLASSIFICATION

Image classification via CNNs frequently explores multiple labeling. The multi-label learning paradigm emerges from the consideration that one real-world object might have several semantic meanings (ZHANG; ZHOU, 2014). Consequently, one solution to account for all of them is to assign a set of labels to the object that explicitly expresses its semantics. In this approach, a label set connected to a single instance represents each object.

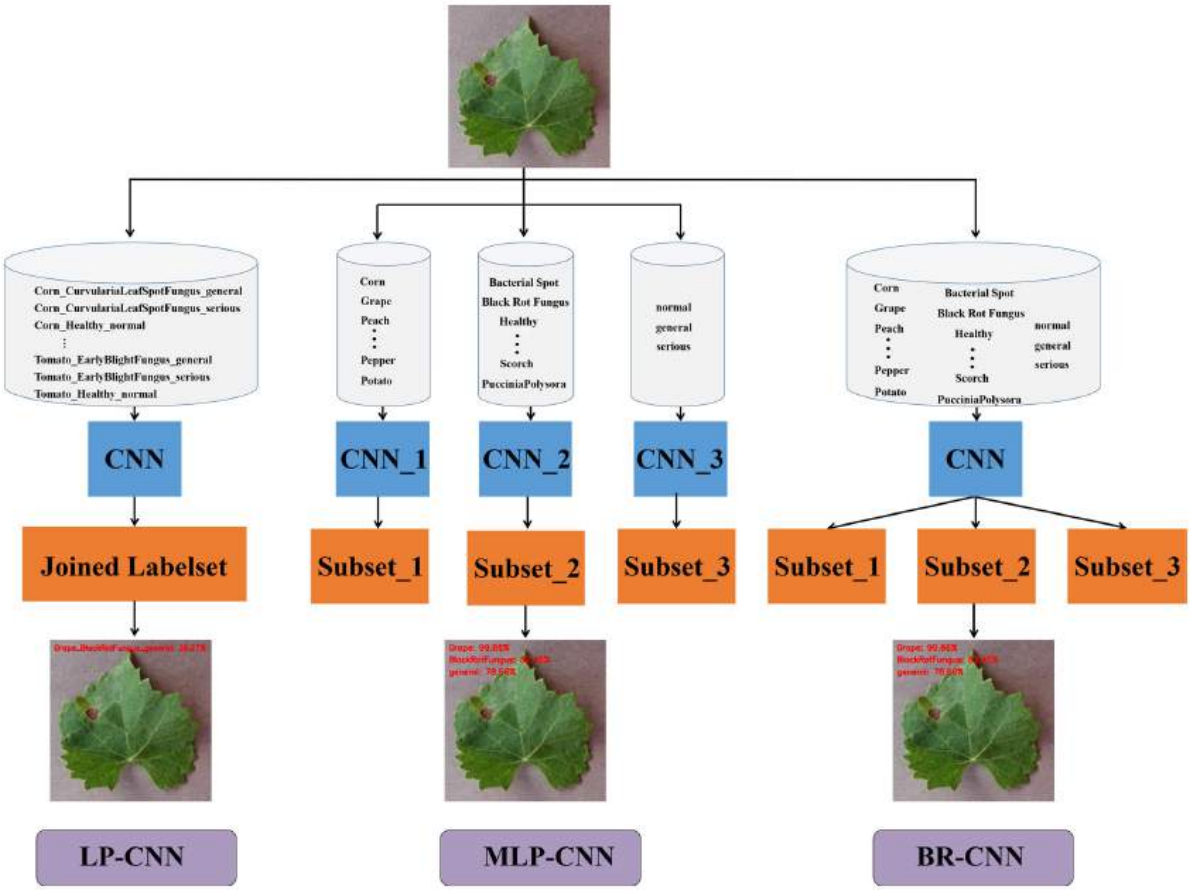
According to Zhang & Zhou (2014), the first researches on multi-label learning focused on text categorization. However, the technique has evolved during the past decade and attracted attention from the machine learning community applying the method to diverse problems. Nowadays, many real-world applications rely on multi-label classification, such as gene classification in bioinformatics, medical diagnosis, document classification, music annotation, and image recognition (TAWIAH; SHENG, 2013).

As stated by Tawiah & Sheng (2013), multi-label classification algorithms employ two basic approaches: problem transformation and algorithm adaptation. Problem transformation is to transfer multi-label classifications into multiple single-label classifications, especially binary classifications. In other words, we can transform object classification into various questions where we ask if the object is an instance of a given class or not. In the second approach, algorithm adaptation, the algorithm learns the structure and correlations between labels to perform the classification.

Ji et al. (2020) presents and discuss some multi-label algorithms, which are displayed in Figure 11. The Label Powerset (LP) method transforms a multi-label problem into a multi-class single-label classification problem. The MLP-CNN technique (proposed by Ji et al. (2020)), transforms the multi-label learning problem into an ensemble of multi-class classification problems, where each component learner in the ensemble is accompanied by one CNN model.

The Binary Relevance (BR) technique, one of the most common in the literature, transforms a multi-label problem into multiple binary problems (one binary single-label problem for each label).

Figure 11 – Illustration and comparison of different multi-label classification frameworks



Source: (JI et al., 2020)

Since the multi-label approach involves the classification of samples regarding several different contexts, various studies have tried to establish a connection between the labels when using this technique, as the ones by Song et al. (2018) and Yan et al. (2019). The methods for finding this correlation usually exploit the architecture of CNNs, use regional dependencies in the pictures, or employ the use of Reinforcement Learning (RL), to cite some examples.

One of the first works with this proposal was written by Wang et al. (2016). The authors propose a framework responsible for learning a joint image-label embedding to model the semantic relevance between images and labels. In more recent work, Li et al. (2020) proposes the use of an adaptive label correlation graph to model label dependencies. The graph learns label correlations with word embeddings and uses graph convolutional networks to map this graph into object classifiers, that are later applied to image features.

2.7 EVALUATION METRICS

Classification experiments include several evaluation metrics to analyze the model prediction performance. According to Hossin & Sulaiman (2015), the evaluation metric can be described as the measurement tool that measures the performance of the classifier. Each different metric evaluates a distinct characteristic of the classifier, but they are calculated based on the confusion matrix elements. This section describes their definition and general meaning.

2.7.1 Confusion Matrix

The confusion matrix displays the number of test samples predicted right and wrong. In a binary classification example, it can be represented by Figure 12 and is composed of the following elements:

- True Positives (TP): Number of objects positively classified that belong to the positive class.
- True Negatives (TN): Number of objects negatively classified that belong to the negative class.
- False Positives (FP): Number of objects positively classified that belong to the negative class.
- False Negatives (FN): Number of objects negatively classified that belong to the positive class.

Figure 12 – Confusion Matrix for Binary Classification

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True positive (*tp*)	False negative (*fn*)
Predicted Negative Class	False positive (*fp*)	True negative (*tn*)

Source: (HOSSIN; SULAIMAN, 2015)

Several metrics result from the confusion matrix's components for analyzing the model performance.

2.7.2 Accuracy

The metric measures the proportion of correct predictions among the total number of examined elements and relates to the frequency of times that the classifier was error-free. It is an appropriate choice for benchmarking when the test set is well-balanced. Equation 2.1 computes the accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \ . \tag{2.1}$$

2.7.3 Precision for Positive Class

The metric measures how much the classifier gets it right when it gives a positive classification. It is an appropriate choice when we want to be sure of the prediction for the positive class or when the False Positives cost is high. It is computed using equation 2.2:

$$Precision_{Pos} = \frac{TP}{TP + FP} \ . \tag{2.2}$$

2.7.4 Precision for Negative Class

The metric measures how much the classifier is right when it classifies the input as negative. It is an appropriate choice when ensuring the negative class prediction. It is computed using equation 2.3:

$$Precision_{Neg} = \frac{TN}{TN + FN} \ . \tag{2.3}$$

2.7.5 Recall

The metric is also called "True Positive Rate (TPR)". It measures which proportion of real positives is correctly classified. In other words, it is an appropriate optimization target when we want to identify every possible positive. It is useful when we don't want to miss any positive element or when the False Negatives cost is high. Equation 2.4. computes the recall:

$$Recall = \frac{TP}{TP + FN} \ . \tag{2.4}$$

2.7.6 Specificity

The metric is also called "True Negative Rate (TNR)". It calculates the proportion of negatives that are correctly classified. Optimizing the specificity value will guarantee a large number of correctly identified negatives. Its calculation, using the equation 2.5, is equivalent to $1 - Recall$:

$$Specificity = \frac{TN}{TN + FP} \ . \tag{2.5}$$

2.7.7 F1-Score

The metric compiles the harmonic average of Precision and Recall quantifying the mutual trade-off. A good value of the F1-Score means that the classifier correctly identifies real threats, but it is not disturbed by "false alarms". It is calculated using the equation 2.6:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \ . \tag{2.6}$$

3 RELATED WORKS

An increasing number of works and studies on plant disease detection build on new technologies in agriculture and progress in artificial intelligence and machine learning techniques. This chapter presents some with a similar scope, which we analyzed during the research and development of this dissertation. The chapter concludes with a comprehensive comparison between selected works and summarizes the possibilities to contribute to the development of this domain.

3.1 THE LITTLE WE KNOW: AN EXPLORATORY LITERATURE REVIEW ON THE UTILITY OF MOBILE PHONE-ENABLED SERVICES FOR SMALLHOLDER FARMERS

Baumüller (2018) presents an exploratory literature review on the impact of mobile phone-enabled services in the life of smallholder farmers from developing countries. The paper gives an overview of the development of mobile phone technology and the possibilities they bring to smallholder farmers. This technology creates the opportunity to reach remote, dispersed, and poorly serviced farmers, overcoming the challenges of countryside life. The author also introduces the "m-services", or services offered through mobile phones.

The work reviews the empirical literature on agriculture-related services offered through mobile phones. The author identifies 23 relevant publications and analyzes them according to the four categories of m-services: services that disseminate information, financial services, services that facilitate access to inputs, and services that enable access to output markets.

The first m-service category is related to information and learning. The author cites agricultural practices, weather reports, or disease outbreaks as relevant information for farmers. These data not only assist farmers in introducing innovations in their work but help them understand and manage the risks. Some of the described studies also report the impact of voice-based information services. Others, for example, disseminate recorded training modules to the farmers via mobile phones. In general, the evaluated works reported that, according to the users, they had improved their knowledge of farming practices, increased yields, and reduced costs.

The second category is related to financial services, including payment services, banking, loans, and insurance. According to Baumüller (2018), they allow farmers to pay for innovations

and associated inputs and to sell their produce. Banking services also assist farmers in managing and earning interest on their savings. The surveyed studies show that m-services help farmers improve their commercialization, input use, and income.

The third m-service category is associated with agricultural inputs. They include information dissemination on input suppliers or prices and facilitated access to water and electricity. However, according to the author, m-services are not yet widely used for this purpose.

The author also explores m-services that provide information on market prices for crops and livestock. These services help farmers link to alternative buyers or markets and facilitate transactions. The presented studies conclude that these m-services help producers secure higher crop prices.

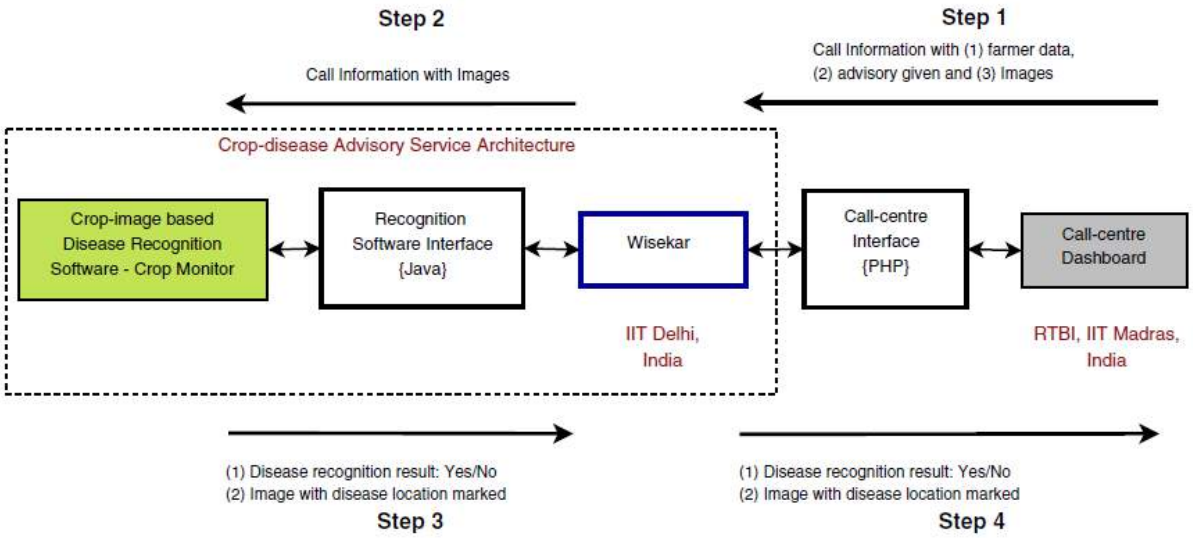
The paper also identifies research gaps in the literature, as, for example, the behavioral factors that influence farmers' willingness and ability to use m-services. The author emphasizes the need to compare between different m-service designs, as, for instance, their prizes or how the m-service is delivered and accessed. Another research gap is the lack of comparisons between mobile phone services and alternatives with a similar scope.

3.2 AUTOMATION OF AGRICULTURE SUPPORT SYSTEMS USING WISEKAR: CASE STUDY OF A CROP-DISEASE ADVISORY SERVICE

In addition to the development of recognition and classification systems based on AI, it is also essential to develop the infrastructure that allows the communication and consulting service to the users. Sarangi, Umadikar & Kar (2016) presents a framework of an agricultural advisory system that uses a call-center approach. The paper states the importance of developing an information and communication system that interconnects the stakeholders (farmers and experts). The work also identifies automating disease recognition as an enabler for increased diagnosis efficiency.

The authors describe the complete system as a service provider for farmers. The farmer uses his mobile phone to capture and send crop pictures to the call center. The farmer may also call the call center to obtain information about the disease management or mitigation, which are available to the farmers in a dashboard and given by experts. Therefore, the system is composed of a database and an interface for providing support services to farmers. The focus of this work is an automated crop-disease advisory service. Figure 13 shows the corresponding architecture.

Figure 13 – Flow of communication in the Automated Crop-Disease Advisory Service (ACAS)



Source: (SARANGI; UMADIKAR; KAR, 2016)

According to Sarangi, Umadikar & Kar (2016), Wisekar is an Internet of Things (IoT) repository designed to store data from multiple domains in a homogeneous structure. When a farmer contacts the system, the call information (data and images) is passed from the dashboard to Wisekar through the interface. A crop monitor requests the raw image and associated data. The monitor processes the image and sends it, along with disease details, back to Wisekar, where the user can access the obtained results via a dashboard. The crop monitor returns the detected crop, the predicted disease, and the location in the image. The prediction is currently in the process of being improved. The round-trip time between the raw image submission and the reception of the results back in the dashboard is the essential measure to evaluate the system performance. This response time correlates with the transferred data size. The evaluation includes different scenarios.

3.3 MACHINE LEARNING BASED PLANT DISEASES DETECTION: A REVIEW

Pardede et al. (2020) presents a review of solutions based on machine learning architectures to detect plant diseases. The authors state that crop losses due to plant diseases correspond to 10 to 50% of the total crop production annually. Therefore, early and correct disease detection could avoid such losses. The reviewed studies, covering both shallow and deep architectures, show how plant disease detection advanced with machine learning techniques.

The paper presents a typical machine learning system focused on plant diseases detection.

The first step is the collection of a sufficient amount of data. Supervised training of plant disease detection systems requires a correct annotation of the training data must be correctly annotated, which is often executed by experts in the domain. Pre-processing of images removes noise during data acquisition and collection. Image classification occurs according to plant disease and employs shallow or deep architectures. For each one of them, the work summarizes the results of several studies that apply it for disease detection.

In shallow machine learning architectures, such as SVM (BOSER; GUYON; VAPNIK, 1992), Naïve Bayes (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997), and k-NN (FIX; HODGES, 1989), the chosen features are very important for the performance. Therefore, Pardede et al. (2020) relates the extracted features for each reported study, as well as the crop and the classifier used in the detection task (usually SVM). One of the reported challenges relates to data dimensionality. Since the images are high dimensional, classifier optimization is demanding. This issue is solved using feature selections and reductions.

The other strategy presented by the authors is employing deep learning architectures, among which CNN is the most widely used to work with image data. Several architectures have been proposed, and some of them are often used in plant diseases detection task, as AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), GoogLeNet (SZEGEDY et al., 2015), MobileNet (HOWARD et al., 2017) and ResNet (HE et al., 2016a), for example. The work summarizes some studies that use deep learning architectures for this purpose, as well as the chosen crops, the results, and the CNN algorithms. The negative side of deep learning techniques is that they require large training datasets, which are not always available. Some workarounds for minimizing this issue are fine-tuning, transfer learning, and data augmentation.

In the conclusion, the authors present the dominant methods in plant diseases detection: SVM and CNN. According to them, SVM has capability to find the best solution if good features can be designed (PARDEDE et al., 2020). In recent years, however, CNN has exponentially grown, being considered nowadays as the "universal approximator", meaning that enough training data can make the CNN learn potentially any function. However, it is still not robust against data and environmental variations, and therefore there is still room for improvement.

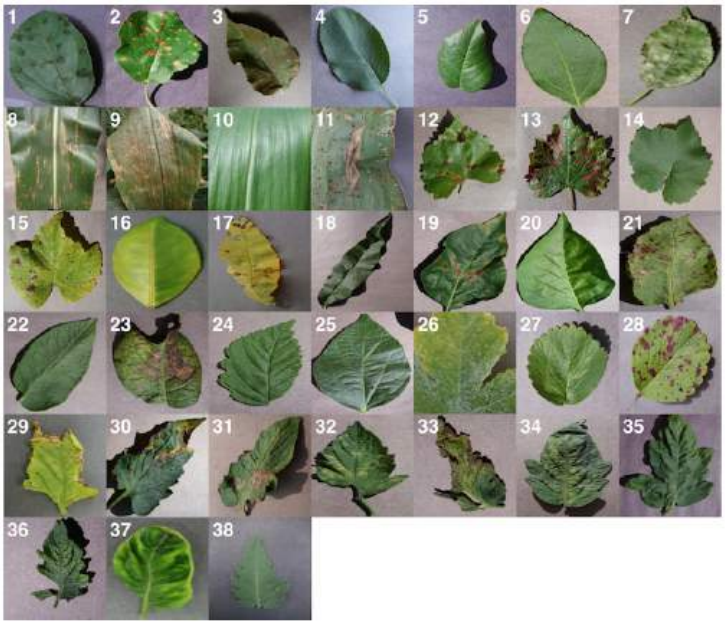
3.4 USING DEEP LEARNING FOR IMAGE-BASED PLANT DISEASE DETECTION

In a pioneering work, Mohanty & Salathé (2016) suggests using a deep learning approach based on image classification to identify selected plant diseases through leaf images. To this

extent, the authors created a dataset for disease classification (MOHANTY, 2016 (accessed March 1, 2021)) composed of 54306 images and trained a Convolutional Neural Network (CNN) to classify them.

The PlantVillage dataset (MOHANTY, 2016 (accessed March 1, 2021)), which is now open-source and used in many works with the goal of disease identification, is composed of plant leaves images from 14 crop species showing signs of 26 diseases (or healthy leaves). The dataset consists of 38 possible classes, each indicating a crop-disease pair. Figure 14 shows some examples of leaf images from the dataset, showing different crops and diseases.

Figure 14 – Example of leaf images from the PlantVillage dataset, representing every crop-disease pair used



Source: (MOHANTY; SALATHé, 2016)

Mohanty & Salathé (2016) also implemented and trained a neural network to correctly identify the pair crop-disease for each input image of a plant leaf. To find the configuration that leads to the best results, they performed experiments with different pre-trained models, different visual representations of the image data, and different dataset divisions. They also compared the results using transfer learning or training from scratch. All the experimental configurations run for a total of 30 epochs each.

The pre-trained models used were AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) and GoogLeNet (SZEGEDY et al., 2015), and GoogLeNet performed consistently better in the experiments. The three versions of the dataset were in color, gray-scale, and segment (just the leaf, without any background). The models performed better with the colored version of the dataset. The experiments also included different percentages of the complete dataset used

during training. Values of 20%, 40%, 50%, 60%, and 80% for the training set (rest for testing) were employed. The best result, as expected, was obtained for 80%. Another expected result was an improved performance for transfer learning compared to training from scratch.

Finally, the authors describe the results from the performed experiments. In the best configuration, the system achieved an accuracy of over 99% when applied over test images from the same dataset. However, the authors took all leaf pictures in a controlled environment presenting similar illumination, focus, contrast, position, and background conditions, which is not the reality in the field. The trained model was also used to classify 121 images collected in a crop field, achieving only an overall accuracy of 31.40%.

3.5 A DEEP-LEARNING-BASED REAL-TIME DETECTOR FOR GRAPE LEAF DISEASES USING IMPROVED CONVOLUTIONAL NEURAL NETWORKS

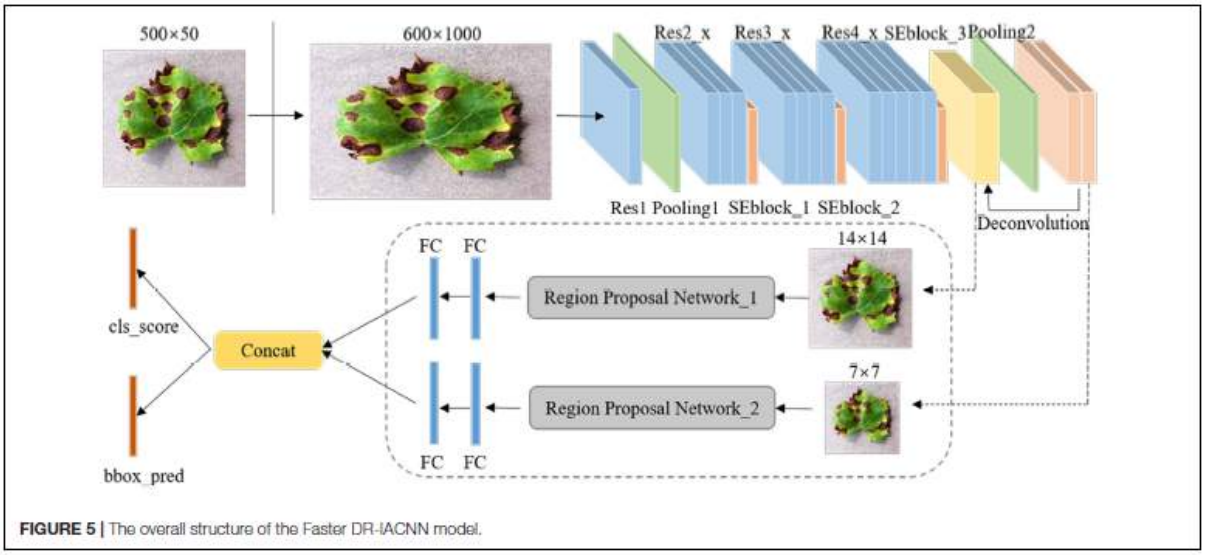
According to Xie et al. (2020), effective detection of grape leaf diseases is essential to ensure its healthy growth. Given the importance of early identification, the work proposes a deep-learning detector based on improved CNNs to monitor grape leaf diseases in real-time. With this purpose, the authors established a grape leaf disease dataset, proposed a CNN model, and applied it to real-time detection of grape leaf diseases.

Grape leaf images from a laboratory and a real grapery compose the dataset. The authors took 4449 pictures covering four disease categories and various climate conditions. The original leaf pictures were augmented and annotated by experts.

For disease detection, Xie et al. (2020) proposes the Faster DR-IACNN model, which consists of three parts. The first one is a pre-network for extracting disease image features, created from a modified ResNet (HE et al., 2016a) backbone network and called INSE-ResNet. It also uses a double-Region Proposal Network (RPN) structure, which has an improved feature extraction ability. It locates and predicts the bounding boxes of the diseased spots. Fully connected layers for classification and regression constitute the third part. Figure 15 shows the overall Faster DR-IACNN model.

In the performed experiments, training employed 60% of the dataset and validation and testing the remaining 40%. 62.286 images composed the extended dataset. Using various feature extractors, the authors benchmarked the proposed technique against previous state-of-the-art methods. The authors also compared the Mean Average Precision (mAP) and detection speed for each experiment.

Figure 15 – Overall structure of the Faster DR-IACNN model



Source: (XIE et al., 2020)

The obtained precision and mAP obtained from the various methods and feature extractors confirmed the better performance of the proposed technique in almost all of the classes, achieving a mAP of 81.1%.

3.6 FACTORS INFLUENCING THE USE OF DEEP LEARNING FOR PLANT DISEASE RECOGNITION

Barbedo (2018) discuss the importance and achievements in plant disorders detection. Moreover, he talks about the limitations and use of different Machine Learning techniques in disease recognition. The main goal of this study was to analyze the main factors that affect the performance of deep learning-based tools for plant disease detection and recognition over field images. This goal is achieved by performing some experiments with CNNs and analyzing its results.

The author performs the experiments over the Digipathos dataset, created by Embrapa (2014 (accessed August 31, 2021)), composed of almost 50000 images of 171 diseases affecting 21 plant species. Dividing the original images pictures into smaller frames focusing on individual lesion spots or localized symptom regions increased the dataset. Training of the CNNs employed three versions of the dataset (original images, with background removed, subdivided images). The author also used image augmentation over the pictures.

The author carefully analyses each misclassification produced by the models and associates

them with a specific causal factor. According to Barbedo (2018), relative differences between the trained neural networks are principal indicators of some of the main factors that affect the effectiveness of their use for plant disease recognition.

The author exposed intrinsic and extrinsic factors that affect disease recognition. Insufficient size and variety of datasets, symptom representation, covariate shift, image background, and different conditions in image capture constitute the extrinsic factors. The intrinsic factors include symptom segmentation, symptom variations, multiple simultaneous diseases, and disorders with similar symptoms.

Figure 16 represents the table from the paper that shows the number of samples and accuracy achieved for each dataset used in the experiments. However, the work mainly focuses on the analyzed factors and their impact on the classification.

Figure 16 – Accuracies obtained using CNNs trained with different datasets

Dataset	# Training samples	Accuracy
Original	1584	76%
Background removed	1584	79%
Subdivided (full)	100,608	87%
Subdivided (reduced)	1584	81%

Source: (BARBEDO, 2018)

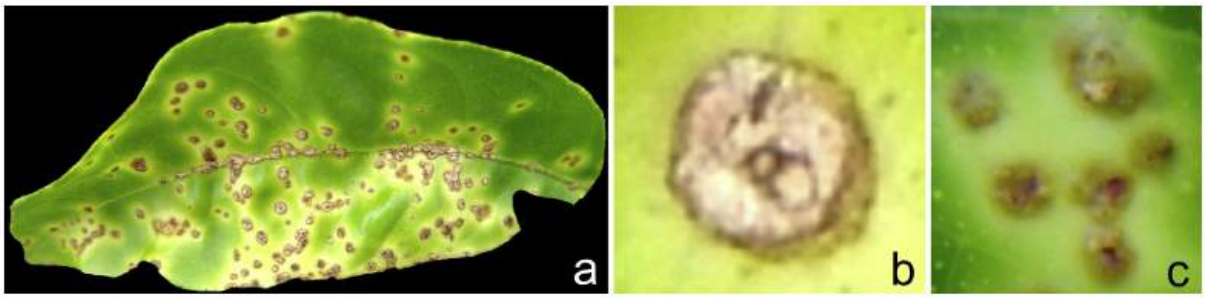
3.7 PLANT DISEASE IDENTIFICATION FROM INDIVIDUAL LESIONS AND SPOTS USING DEEP LEARNING

Barbedo (2019) proposes a system for plant disease recognition using CNNs. It states the limitations of using a dataset created in a laboratory without any variations in the pictures and proposes a dataset that overcomes this limitation. He also segmented the images into individual lesions and spots. The described approach increased data diversity to identify multiple diseases affecting the same leaf. The resulting accuracy values obtained for each crop varied from 75 to 100%.

The experiments employed the Digipathos dataset, created by Embrapa (2014 (accessed August 31, 2021)) and composed of images taken using different sensors. The dataset contains about 60% of pictures captured under controlled conditions and 40% under field conditions. Initially, it included 1567 images representing 79 diseases affecting 14 plant species, including diverse conditions and varying symptoms. The experiments used only pictures containing plant

leaves with the background removed. The images were then divided into individual lesions, creating the expanded dataset. The author identified five different signs and symptoms, depending on their characteristics. Figure 17 displays an example of a diseased plant leaf image from the Digipathos dataset, followed by an isolated lesion and a cluster of lesions, the last two being part of the original picture.

Figure 17 – Example of scattered small symptoms (a), isolated lesion (b), and cluster of lesions (c)



Source: (BARBEDO, 2019)

Barbedo (2019) applied Transfer Learning to a pre-trained GoogLeNet CNN (SZEGEDY et al., 2015), and chose the training parameters using a grid search technique. He performed two groups of experiments. The first group was focused on determining the origin of a symptom already located. The original pictures, pictures with the background removed, and augmented images constituted three different datasets. The author investigated the training dataset size and the impact of severe class imbalance. The second group centered on detecting disease signs that need further classification. To this extent, the author analyzed the inclusion of healthy samples and the generation of a new test dataset. Using a sliding window to generate new cropped pictures enabled the classification of the disease level.

The first group of experiments revealed that the performance varied among different crops and dataset sizes, especially when using the expanded dataset. The paper analyzed the resulting confusion matrices for the investigated crops. Based on the second group of experiments, the author concluded that including healthy samples had little impact on the model's effectiveness. Moreover, the efficacy of the detection approach is related to the prominence of symptoms in the image.

3.8 MULTI-LABEL LEARNING FOR CROP LEAF DISEASES RECOGNITION AND SEVERITY ESTIMATION BASED ON CONVOLUTIONAL NEURAL NETWORKS

Ji et al. (2020) proposes a system composed of a series of networks to automatically recognize crop leaf diseases and estimate their severity based on images. The authors state the importance of correctly diagnosing plant diseases and discuss state-of-the-art approaches.

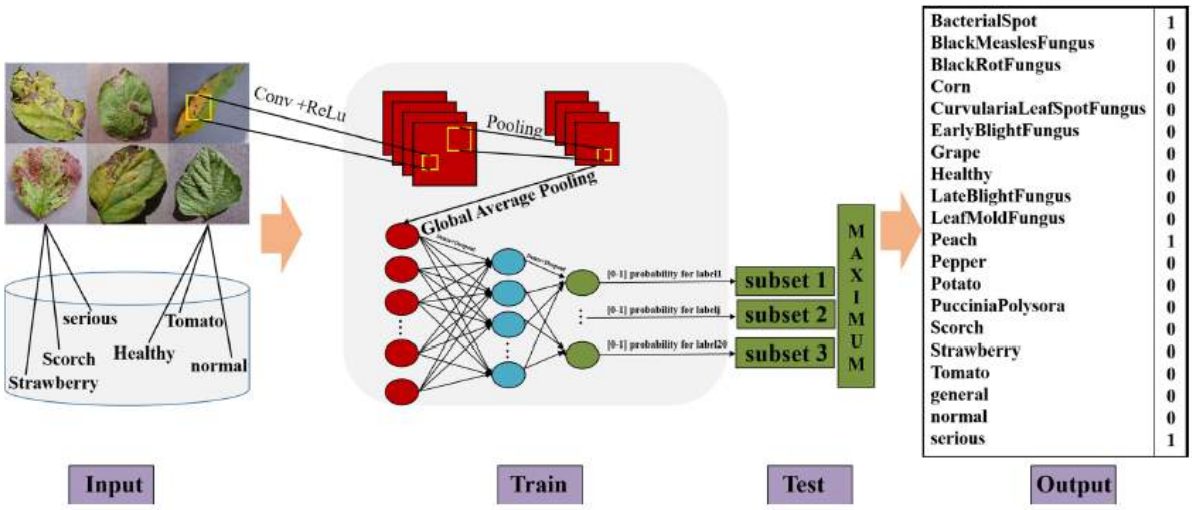
The recognition process described in the paper employs images of crop leaves. Therefore, the dataset used to train the classification models is the AI Challenger (HUGHES; SALATHE, 2015), composed of 12691 images that include 31 categories and 20 unique labels. Division into train-validation-test subsets follows the proportion 80%, 10% and 10%, respectively.

The authors combined several state-of-the-art CNNs with multi-label learning algorithms to perform the diseases recognition and severity estimation. Among the tested architectures are GoogLeNet (SZEGEDY et al., 2015), ResNet (HE et al., 2016a), DenseNet (HUANG et al., 2017) and NasNet (ZOPH et al., 2018). The work trained the CNNs using a transfer learning strategy to improve the classification performance. In addition, a technique that transforms a multi-label problem into multiple binary problems, one for each label. is developed. According to this approach, called "BR", each binary model predicts the relevance of one of the labels.

Figure 18 displays the pipeline of the work proposed by the authors. The three picture labels are crop, disease, and severity. One classifier is trained for each label, performing a binary classification in "positive"and "negative"examples. A deep CNN with one output probability for each label models the classifier. As test metrics, the authors report the probabilities of each subset and the maximum value.

The authors present and discuss the results obtained in the experiments. The CNN based on DenseNet achieved better results in crop diseases recognition. However, for crop diseases severity estimation, the CNN based on ResNet obtained a better performance. The work also shows that the multi-label technique used, BR-CNN, outperforms other multi-label methods in crop leaf diseases recognition and severity estimation.

Figure 18 – Pipeline of the proposed BR-CNN for crop leaf diseases recognition and severity estimation



Source: (JI et al., 2020)

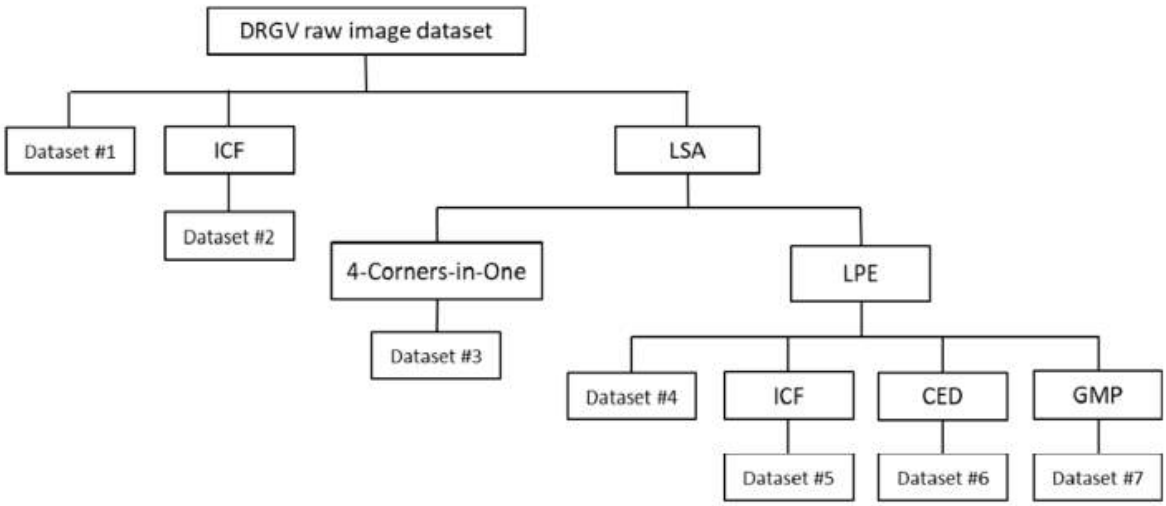
3.9 DEEP LEARNING TECHNIQUES FOR GRAPE PLANT SPECIES IDENTIFICATION IN NATURAL IMAGES

Several factors may cause harvest variations within vineyards from year to year, as, for example, soil conditions, diseases, pests, climate, and pesticides. Additionally, some vineyards have more than one grape variety. Variety identification becomes crucial to assure a more targeted treatment since some grape varieties are more susceptible to certain pests and diseases. This identification is a challenge mainly due to the high visual similarity of the leaf images on different grape varieties. Literature on this area usually uses image datasets created in laboratories, and the approaches that use pictures collected in the field face several environmental issues. Given this scenario, Pereira, Morais & Reis (2019) evaluates the performance of transfer learning and fine-tuning techniques when applied to the identification of grape varieties.

The generation of an augmented pre-processed dataset for image classification includes image processing over the raw dataset, followed by a data augmentation over the pre-processed dataset. Two hundred twenty-four images from six different grape varieties were collected over two years, composing the raw dataset. Next, the authors pre-processed different subsets of the complete dataset. First, the modification of seven subsets employed diverse image processing methods such as signal processing techniques, leaf vein extraction strategies, noise detection, image rotation, and warping methods.

Pre-processing of another group of subsets applied data augmentation techniques helping

Figure 19 – Flowchart of the pre-processed dataset generation process



Source: (PEREIRA; MORAIS; REIS, 2019)

to reduce overfitting in the training process. "Fake samples"were introduced by Zhang et al. (2019) and are obtained by three different means: one-pixel image translation with a given factor, horizontal image reflection (mirroring), and image rotation. The new samples composed a dataset of 10 times the original data. The flowchart in Figure 19 illustrates the generation process of the pre-processed datasets. Data preparation concludes with splitting the pre-processed datasets into the train, validation, and test sets. Images also need to be resized to a fixed size.

The proposed plant species identification employs a Convolutional Neural Network (CNN). The used architecture was AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), pre-trained over a subset of the ImageNet database (DENG et al., 2009). The authors focused on transferring the pre-trained AlexNet architecture to the specific task of grape variety classification in natural images (PEREIRA; MORAIS; REIS, 2019). The work used the transfer learning technique with fine-tuning and fixed feature extractor schemes. The performed experiments used different subsets of the complete dataset, different dataset splitting ratios, and different data augmentation and fine-tuning techniques. The maximum test accuracy achieved was 77.30%. The authors also evaluated the results regarding the individual varieties. The best accuracy value was 89.1%, and the worst one was 65.65%.

3.10 COMPARATIVE ANALYSIS BETWEEN RELATED WORKS

We presented two types of related works. The first group is composed of reviews of studies in the area of technology and agriculture, that analyze them and point to some relevant characteristics in the domain. The second group includes works with similar approaches to the present one, using image classification in a specific application related to agriculture. In order to establish a comparison between the related works, we will analyze the two groups separately.

Since the first group was composed of overview studies and reviews, we will not compare them using a table. Instead, we will only highlight the main conclusions and analysis relevant to the development of our work.

- Baumüller (2018) executes a review and analysis on services offered to farmers through mobile phones. The first category described by the author relates well with our objective of assisting farmers by offering information.
- Sarangi, Umadikar & Kar (2016) performs a case study that proposes a framework as an agriculture advisory system for crop-disease detection. However, the work does not implement the crop-disease detection system or the technologies used for the model training.
- Pardede et al. (2020) reviews machine learning technologies used as solutions to detect plant diseases, as well as their advantages and disadvantages. Our work proposes the use of a deep learning technique (CNN) to perform the proposed task.

For each work of the second group, we will analyse the following characteristics:

- **Objective:** The main achieved objective;
- **Dataset:** The used image dataset, including size and collection information (field versus laboratory);
- **Strategy:** The deep learning strategy and neural network architecture used in image classification;
- **Variations:** The evaluation properties and changes proposed by the paper;
- **Results:** The main obtained results.

Table 1 establishes the comparison between the related works from the second group.

Table 1 – Comparative analysis between Related Works

Related Work	Objective	Dataset	Strategy	Variations	Results
(MOHANTY; SALATHé, 2016)	Identify crop and disease through leaf image	PlantVillage, 54306 images, Lab	CNN	Creation of new dataset, Architectures, dataset versions, Dataset split ratio, learning type	99% accuracy over PlantVillage dataset, 31.4% accuracy over field images
(XIE et al., 2020)	Real-time detection for grape leaf diseases	Custom, 4449 images, Lab and Field	Faster DR-IACNN	New architecture, INSE-ResNet, double-RPN	81.1% mAP
(BARBEDO, 2018)	Discover factors influencing plant disease recognition	Digipathos, 50000 images, Lab and Field	CNN	Extended dataset, 3 datset versions, Intrinsic and extrinsic factors	87% accuracy over subdivided dataset
(BARBEDO, 2019)	Plant disease identification from individual lesions and spots	Digipathos, 1567 images, Lab and Field	CNN (GoogLeNet)	Extended dataset, 3 datset versions, Lesions and spots separation	82% accuracy over original dataset, 94% accuracy over extended dataset
(JI et al., 2020)	Crop leaves disease recognition and severity estimation	AIChallenger, 12691 images, Lab	CNNMulti-label	Architectures, multi-label multi-binary technique	94.71% precision 94.70% recall 94.70% F1-Score
(PEREIRA; MORAIS; REIS, 2019)	Identify grapes variety through leaf images	Custom, 224 images, Field	CNN (AlexNet)	Image processing methods, Data augmentation, Transfer Learning, Fine-tuning	77.3% avg. accuracy 89.1% max accuracy for particular variety

Source: the author (2021)

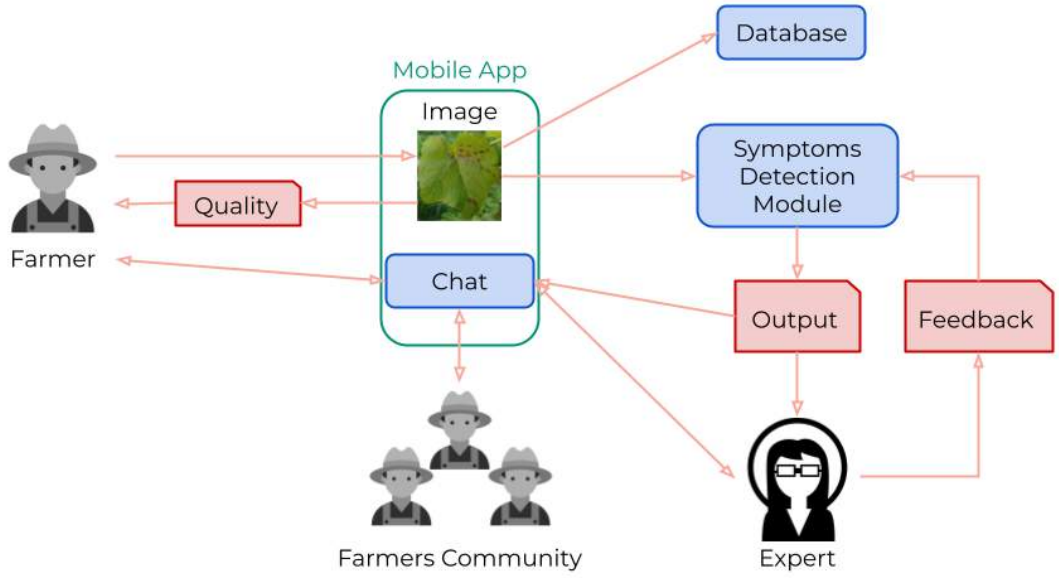
4 PROPOSED APPROACH

This work implements a system capable of detecting disease symptoms over plant images. Therefore, the developed system architecture described below addresses a real-world problem.

4.1 SYSTEM OVERVIEW

As mentioned in Section 1.3, the proposed work aims to improve a system implemented to assist the Phytosanitary Clinic of Pernambuco (CliFiPe). The clinic's general intention is to help farmers monitor their plantations and take measures to correct their problems. The clinic attempts to inform the producers of disease outbreaks by early detection of symptoms and pests and aid in disease treatment. The developed symptom classifier amplifies the possible services of a digital platform designed for communication between farmers and clinic experts. Figure 20 below gives an overview of the complete system.

Figure 20 – General architecture of the complete system composed by a mobile app and a digital assistant for a crop clinic



Source: the author (2021)

The proposed platform is accessible to farmers through a mobile application developed with this purpose. The user selects a crop from a predefined collection and takes a leaf picture. The system analyses the picture quality and automatically asks the user to retake it if it is

not enough to detect the disease. Otherwise, images are automatically divided into segments and submitted to the classification system to detect the probability of being diseased. The system's feedback is sent both to the farmer (displayed in the app) and to an expert from the clinic, whose feedback about the output helps improve the classification. All images are stored in a database and are used to fine-tune and continuously optimize the classification systems' performance. The app also displays general disease information such as the most common symptoms, prevention, and generic treatment measures. However, more specific advice on disease control is left to direct communication with a phytopathology expert. The knowledge base is an information database created by experts from CliFiPe. The mobile app allows the farmer to directly communicate with the experts, exposing doubts and receiving assistance with problems in their crops. It also enables the creation of a community, granting an opportunity for farmers to help each other and share knowledge. The initial version of the platform and the mobile app have already been implemented and are currently in use by initial users and experts from the clinic.

This work, however, is focused on the Symptoms Detection Module. One of the specific objectives of this work is to identify whether the plant is diseased. For this purpose, the system uses machine learning and computer vision techniques to analyze leaf images and detect whether they show or not disease symptoms. The following Section describes the module in more detail.

4.2 SYMPTOMS DETECTION MODULE

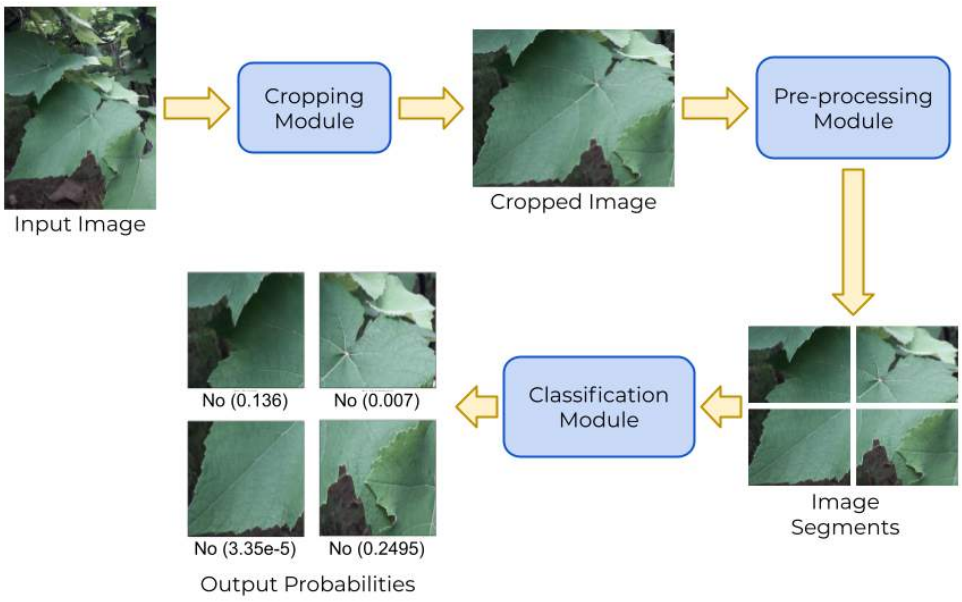
As described in Sec. 1.3, the main goal of this work is to identify the presence of symptoms caused by pests and diseases over plant images. More specifically, the developed system aims to detect disease symptoms revealed by leaf images submitted by the users. Since the farmers take the pictures presented to the module, the field environment will influence the detection result. Possible influence factors include variations of lighting, position, and focus. Moreover, pictures may present shadows, intense sunlight, or several leaves. We classify images hampered by these conditions as low-quality pictures. The detection of disease symptoms in low-quality images is especially challenging for the system. Consequently, one of the specific objectives of this work is to identify the quality of a taken picture to guarantee that it is enough to perform symptoms detection. As described in Section 4.1, the farmer defines the image crop before taking the picture, and so the AI system is not intended to perform this classification.

Sec. 3 delineates some examples of works that perform similar tasks. However, usually, pictures are taken under supervised conditions using a standard background. Available training datasets, such as the PlantVillage, described in Mohanty (2016 (accessed March 1, 2021)), and the Digipathos dataset, described in Embrapa (2014 (accessed August 31, 2021)), do not include images with variations in lighting, size, or framing, and thus does not reflect actual field conditions. As a result, according to Mohanty & Salathé (2016), models trained over such datasets achieve only an average precision of 31% when applied to authentic field images. The work described here uses photos taken in the field to train the neural network model. Therefore, the training dataset shall exhibit similar picture quality and detail variations as pictures taken with a mobile phone by inexperienced users. A CNN trained in this way will recognize disease symptoms more reliably.

In addition, we collected the images used in the training and inference experiments in a rural region of the state of Pernambuco. Thus, the training dataset includes images with similar characteristics to the ones submitted by the local farmers. Consequently, the trained model will better perform in the classification. We used images of grape leaves, one of the most common crops cultivated in the state, for the experiments described in this work.

Fig. 21 shows the general architecture of the digital assistant. The system takes the raw images to be classified as input and consists of three modules: the cropping module, the pre-processing module, and the classification module.

Figure 21 – Tasks performed by the digital assistant



Source: the author (2021)

4.2.1 Cropping Module

Since the images to be classified will be taken by different users and in varying conditions, they likely present distinct characteristics, including variations in size, framing position, and camera-to-leaf distance. Additionally, a visible background in an extended picture area confuses the classification network. Therefore, the first module will crop the input image, centering the leaf and eliminating parts of the picture background to mitigate this problem. Manually cropped images constitute the training and validation data sets to guarantee a certain standard, improving the classification performance of the CNN model. The pictures taken by the mobile app users will be used, in the future, to train an algorithm to crop inference images automatically.

4.2.2 Pre-processing Module

After cropping, each picture presents a standard that mainly displays the leaf. However, various disease symptoms do not distribute uniformly on the leaf surface. Most times, they appear in the form of spots or lesions that cover only a part of the leaf, according to Barbedo (2019). That may lead to an incorrect classification by the detection module. Hence, this work pre-processes the cropped picture before sending it to the classification module.

Firstly, in order to facilitate the detection of the symptoms when they cover only one part of the leaf area, the image is divided into segments, which serve as inputs for the classification module. This process involves dividing the pictures into rectangular segments of the same size, and the segment number depends on the frame proportions. Frames with a height-to-width ratio below a given threshold (in this case, 1.25) divide into four, otherwise six segments. The chosen threshold value guarantees that each picture segment shows a similar degree of detail. Moreover, this segmentation increases the size of the training data set, allowing to exclude parts with limited information. Also, it gives more flexibility in balancing the training data set by selected inclusion of segments of varying exposure conditions and quality. The Bilinear Interpolation technique (presented in Smith (1981)) resizes the generated picture segments to match the input shape of the neural network (256x256x3).

The leaves pictures segments can also be used in future works to identify the disease severity based on the leaf area covered by symptoms and the number of segments presenting them.

4.2.3 Classification Module

This module performs the main task described in this work. It uses Convolutional Neural Networks to classify the input segments. A CNN, as described in Sec. 2.3, is a deep learning technique widely used in pattern recognition that employs a single network to learn and classify the image features, according to Voulodimos et al. (2018). We based our model on the ResNet50V2 pre-trained model, proposed by He et al. (2016a) and described in Section 2.5.

Since the main objective of our work is to detect diseased leaves using their images, we developed a system where the Neural Network classifies the input segments as showing symptoms or not (classes "Symptoms"and "No Symptoms"). However, not all pictures present sufficient quality to perform the classification. Thus, we will also filter these images employing an additional classification scheme evaluating the photo quality for being adequate (or not) to detect disease symptoms.

Different classification strategies were developed and experimented to find the optimal performance solution, including training a single model to classify both picture features, the symptoms' presence and the photo quality. We detail the image training dataset and the module implementation in Chapter 5. The digital assistant performs binary classification to detect the diseased leaves and forwards the segments with the most pronounced symptoms (highest probability) for further analysis with a phytopathology expert. Selecting the picture segments that most clearly demonstrate the disease symptoms is a valuable step in disease identification and agent recognition, either by human experts or another machine learning model.

4.3 SYSTEM INTEGRATION

The Symptoms Detection Module, described in the previous section and the focus of this work, is part of the general system presented in Section 4.1 and developed along with other students. The module is inserted in a digital platform that acts as an image database and interfaces with phytopathology experts. It receives from the mobile application the photos taken by the farmers, along with other relevant information (location, farmer identification, date and time, crop).

The platform will apply the trained model over the photo to identify if its quality is good enough to detect symptoms. If not, the system will send a message to the mobile application,

asking the user to take another picture. Otherwise, the disease identification module output will be displayed to the expert (in the interface) and the farmer (in the mobile application). The expert will then further inspect the image, in case it shows symptoms, and will assist the farmer directly through the chat in the application. He will also provide feedback about the classification (confirming or correcting the output), that will be used to improve the system performance dynamically.

Future works include the conclusion of this integration, which is under development in the current implementation stage.

5 METHODS

This chapter presents and describes the methods used during the development of this work. First, Section 5.1 introduces the baseline experiments in the literature for comparison. Section 5.2 details the creation of the dataset used in the study. Section 5.3 describes the performed experiments. Section 5.4 specifies the technical details of the implementation. Finally, Section 5.5 defines the experimental setup used in this work.

5.1 BASELINE EXPERIMENTS

The first step in developing the proposed system was to apply a state-of-the-art classifier to the challenge at hand, essential for validating the improvements suggested in the present work. Moreover, the application requires assessing the availability of suitable training and testing data.

The experiment chosen as the baseline is the disease identification proposed by Mohanty & Salathé (2016). It performs an image classification according to the crop and the disease, using a CNN technique. The authors propose the use of the PlantVillage Dataset, available in Mohanty (2016 (accessed March 1, 2021)). The dataset contains 54306 images divided into 14 crop species and 26 diseases, including pictures of healthy leaves for comparison. The work described in the paper evaluates different CNN architectures, different versions of the dataset, different ratios between training and testing dataset, and also the difference between the CNN learning from scratch and using transfer learning.

We adapted the experiment to our application scenario. We chose to use only the images from the crop "grape", resulting in 4062 pictures taken from the PlantVillage dataset and belonging to 4 different classes (3 disease classes and the "healthy"class). Furthermore, the pictures were divided into segments, resulting in the following number of images to be used in the experiments:

- "Black Rot"= 4720 images;
- "Black Measles"= 5532 images;
- "Leaf Blight"= 4304 images;
- "Healthy"= 1692 images;

After separating the dataset for train and test, we obtained 12998 images (80% of the dataset) for training and 3250 images (20% of the dataset) for validation and testing.

The classification performed in this experiment used a multi-class CNN. We used a pre-trained ResNet50V2 model and applied Transfer Learning over it. Before the training, the images were resized to the size 224x224x3, randomly flipped, and normalized. It was used Pytorch as the framework, 32 as batch size, sigmoid as activation function, and binary cross-entropy as loss function. The chosen optimizer was Adam, with a learning rate of 0.001. We focused on the F1 metric (Equation 2.6) during validation and test.

5.2 LOCAL DATASET CREATION

As demonstrated in the baseline experiment, there are currently some available online datasets successfully employed in plant disease experiments, e.g., the PlantVillage, created by Mohanty (2016 (accessed March 1, 2021)), and the Digipathos dataset, created by Embrapa (2014 (accessed August 31, 2021)). However, their images are usually taken in a lab and do not present a pronounced background or variations in lighting, size, or framing, thus, not reflecting the field conditions. Therefore, we manually built a local dataset for the CNN training to improve the classification performance when applied to images taken in the field.

The dataset employed in this study is composed of images we collected in the region of the Siriji valley, in the countryside of Pernambuco. There are several smallholders plantations in the area, and we collected images from plant leaves at some of them. Among the cultivated crops, the most important ones for the region's economy and production include grape, banana, and sugar cane. In order to start the dataset creation, we chose to collect images of grape leaves. Figure 22a shows a picture taken in one of the plantations where the images were collected, and Fig. 22b shows the map location from the area of the images collection.

The image collection happened in April 2021. All the images were collected in the morning, between 9:00 and 12:00, on a sunny day. The images were gathered using the cameras of four different smartphones, with an average camera resolution of 13 Mega-pixels. The pictures were taken from an average distance of 30 cm from the leaves, trying to center them in the frame. Since the objective of this collection was to create a diverse dataset, we included images both presenting or not sunlight and shadows. We also incorporated some photos out of focus or taken against the sunlight. All the pictures show a natural background, with the possibility of including ground, other leaves, or the sky. We included all these variations due to the objective

Figure 22 – Image collection for local dataset

(a) Plantation used for image collection

(b) Map location from image collection

Source: the author (2021)

of training a classification model that will be applied over images taken in the field.

For this study, we trained a CNN to identify disease symptoms in grape leaves images, but the extension to other crops is straightforward. First, we took pictures of healthy and diseased leaves in different growth stages and lighting conditions. Then, phytopathology experts from CliFiPe annotated the photos, identifying them as manifesting symptoms (class "Symptoms") or not (class "No Symptoms"). After the expert annotation, the dataset size was:

- 1987 images for the class “No symptoms"
- 1302 images for the class “Symptoms"

The collected pictures were then manually cropped and divided into segments, as described in Sec. 4. After this step, they are ready to be divided into subsets and used in the experiments.

We did not use all the images for the training of the neural network models. First, we separated a small number of representative images to illustrate how particular image characteristics influence the classification results of selected trained models. Twenty-four leaves images (twelve from the class "No Symptoms"and twelve from "Symptoms") were chosen for this final case study, exhibiting different illumination, focus, and contrast conditions. The choice ensured some variability and balance between picture classes and exposure conditions. No dataset for model training, validation, or testing included the selected images, and no performance statistics were derived. The final case study, reported in Sec. 6.2.3, only illustrates possible outcomes when the trained models are applied, emulating the challenges of a usage scenario

at a large scale. The case study images were segmented, generating a total of 104 segments (54 from the class "Symptoms"and 50 from "No Symptoms").

Secondly, 80% of the remaining images constituted the training and 20% the testing dataset as described in Section 2.4.1. Repeating the random division into train and test datasets allowed the training and verification of distinct variations of neural network models. We will use this approach to estimate the expected performance variability when applying these models for the digital assistant in a crop clinic. Moreover, we gain insights into the composition of a well-balanced training dataset.

After cropping and segmentation of the collected images, the complete dataset contained 4556 images for the class "Symptoms"and 6662 images for the category "No symptoms". The previously described division into datasets for training and testing can occur before or after segmentation. After the division, the set sizes are:

- Train set: 8974 images (3644 for the class "Symptoms"and 5330 for "No Symptoms")
- Test set: 2224 images (912 for the class "Symptoms"and 1332 for "No Symptoms")

The neural network model training relies on the generated train subset. For this reason, the set must be composed of a sufficient number of images covering a significant variety of segments. The test set is used for the trained model to perform predictions. It allows estimating the performance when used as a digital assistant, classifying pictures taken by the mobile app users.

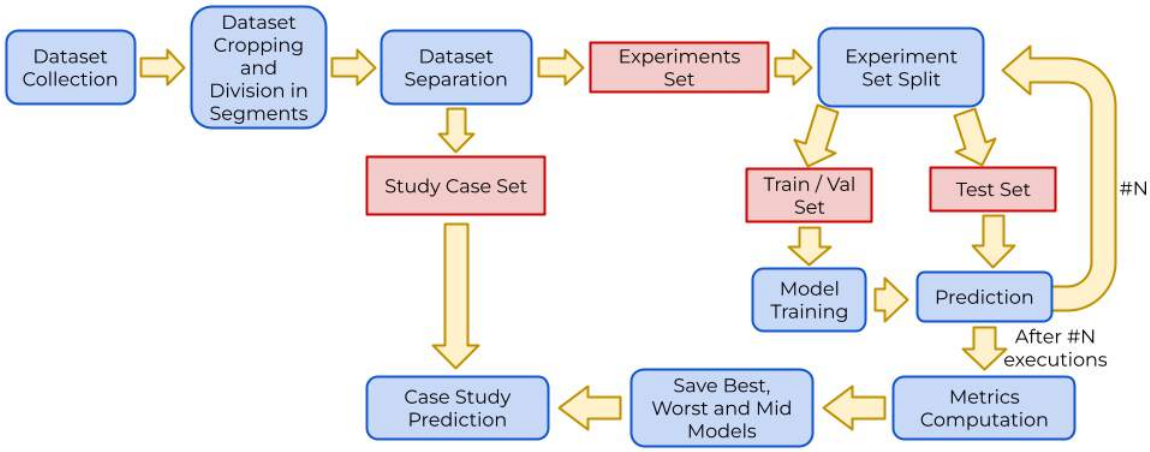
5.3 PERFORMED EXPERIMENTS

5.3.1 Experiments Pipeline

Before experimenting the different classification scenarios, we tested and evaluated different dataset division strategies (into train and test set) and hyperparameter value options, in order to determine the ones that maximize the model performance. Among the hyperparameters, we tested different values of dataset size, different weight values for both classes, different number of repetitions of the training pipeline, different evaluation metrics, and different classification thresholds. In addition, we also compared if the dataset split worked better before or after dividing the pictures into segments.

Fig. 23 shows a diagram that summarizes the experiment flow. We implemented a script to repeat the complete assessment (training and testing) a certain number of times (N) while performing a grid search for high-performance models. For each repetition, the dataset is randomly divided (maintaining the ratio of 80/20). Then, we apply the trained model to the respective test dataset of the repetition and compute the evaluation. Calculating the mean and standard deviation of the predicted recall metric from each execution concludes each of the N assessments. Although not representing a systematic cross-dataset validation, the approach will identify flyers, i.e., models that show atypical performance due to an accidentally introduced bias in the randomly selected training dataset. The script also indicates the models giving the best, the worst, and the recall closest to the mean. In addition, the observation of several prediction measures guarantees consistent model behavior complementing the optimization of a specific evaluation metric that defines the experiment goal. For all experiments, we also observed the accuracy metric (Equation 2.1) since it involves all the elements of the confusion matrix. The accuracy computation acts as a complementary metric, ensuring that the optimization of a given metric does not decrease the other ones.

Figure 23 – Flow diagram of the performed computational experiments



Source: the author (2021)

Applying these models to the case study dataset, composed equally of healthy and diseased leaves images with different quality levels, gives further insights into the strategy for finding a well-performing model. A suitable training dataset balances the number of pictures belonging to separate classes. In addition, pictures of high and low quality need to be equilibrated to account for image variability. The training of several distinct models helps understand the expected network performance, capability, and limitations. Moreover, the assessment of the training impact becomes possible.

5.3.2 Symptoms Detection

As described in Section 1.3, the main goal in this work is to develop a system that can detect the presence of disease symptoms in pictures of plant leaves. We conducted a first classification experiment using the collected and pre-processed local dataset. We trained a neural network to classify if an input image of a leaf shows symptoms or not. Since it is a binary classification, we considered the images that reveal disease symptoms as belonging to the class "1", otherwise to the class "0". For this experiment, since the FN cost is high, we optimized the recall evaluation metric to avoid erroneously missing any image that displays symptoms.

The experiment relied on the local dataset described in Section 5.2. Since the dataset contains pictures taken in a crop field, the model will learn the classification with varying exposure conditions present. That includes sunlight, shadows, different backgrounds, and other objects that may also appear in the image. Such an approach is required, since the system shall classify pictures taken by farmers in the field in real-time. However, the described steps were taken during dataset preparation to ensure a certain level of standardization of the photos.

After the training process, we applied the trained models over the previously separated case study set to evaluate the experiment performance in detail. Since the set is composed of images not used during training, it allows us to emulate an application of the system over new field images. This process also enables the visualization of the results for each picture in the dataset. Such visualization may point to possible challenges in the classification.

Complementing the analysis of the detection of the symptoms, we created a filter over the misclassified images from our dataset that progressively removes these from the test set. This investigation highlighted the need for a strategy to filter the pictures that the system will most likely consider challenging to classify.

5.3.3 Pictures Quality Classification

The observed results from the performed experiments revealed the importance of the dataset used to train the model. Since the images originate from a crop field, they are subject to characteristics that may affect the classification, e.g., a dominating background, too much sunlight, or a picture out of focus. It was possible to notice how the variability and the quality of the images interfere with the classification. Therefore, we trained another neural network

model that acts as a filter and decides if the picture quality is enough to grant symptom classification.

The next group of experiments implemented a system to classify each segment picture according to its quality. Similarly to the previous group, a CNN performed the binary classification. Since our goal is to filter the pictures of low quality, there are two possibilities of classes: "Field"and "Low". The "Field"class corresponds to the images taken in a crop field with good enough quality to be classified. The "Low"class is related to the pictures whose quality may disturb the classification. The idea is to apply this model to the photos taken by the user. If such a photo classifies for "Field"quality, then the symptoms detection model is executed over it. However, if the predicted class is "Low", the mobile application informs the user and asks him to take another picture.

This experiment fully exploits the local dataset, capitalizing on the included crop field images with "low quality". According to our interpretation of low-quality characteristics, we separated the pictures into two quality classes. We classified as "low quality"images the ones that are out of focus, or the leaf is hidden in the shadows (and the contrast between sunlight and shadows makes it challenging to distinguish the leaf), or there are other objects in front of the leaf, preventing the identification of its contours. It is important to emphasize that these features constitute our interpretation of challenging pictures, and, therefore, this is a subjective criteria.

The complete dataset includes 11218 image segments, 9740 from the class "Field"and 1478 from "Low". The training set employed 80% of the dataset (8974 images that were later balanced) and the validation set, the remaining 20% of the dataset (2244 images). For this application, the weighted precision metric (weighted average between equation 2.2 and 2.3) was most closely observed among the prediction results, along with the accuracy (Equation 2.1).

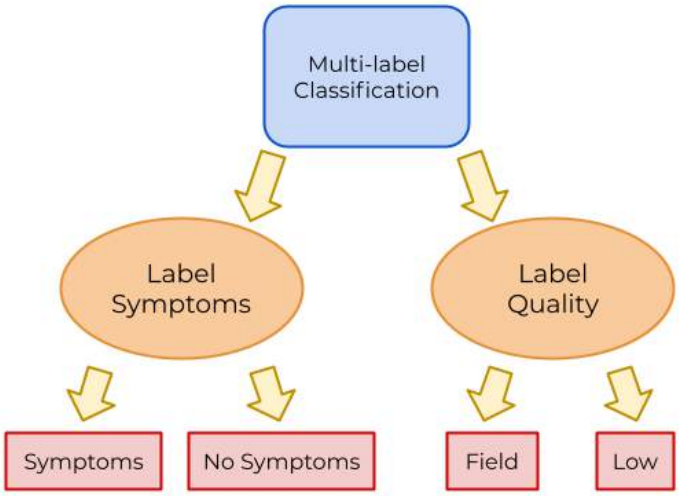
The trained classifier sorts the images according to their quality. After that, we applied the previously trained models described in Section 5.3.2 (performing disease symptoms detection) over the two subsets (high and low quality). The experiment showed the performance improvement generated by this filter.

5.3.4 Multi-label Experiments

As explained in the previous section, the quality and variability of the images play a decisive role in their classification. Therefore, we developed a filter to determine the quality of the image before the detection of the symptoms. However, another possibility is to use a single neural network model to perform both classifications.

The multi-label technique explained in Section 2.6 enables the training of a single model that classifies the input into multiple labels at once. A multi-label CNN has the advantage of being faster and using less space than several separate models, maintaining the classification performance. The training process is also simplified since it comprises a single neural network. In the present study, a multi-label CNN simultaneously classifies the input leaf image according to the quality ("Low"or "Field") and to the presence of symptoms ("Symptoms"or "No symptoms"). Figure 24 represents this idea, displaying the labels and the classes related to each one.

Figure 24 – Diagram representing the multi-label approach



Source: the author (2021)

We needed to change the dataset labels and the implementation code to apply the multi-label technique. The annotation of each dataset image comprises the two tags "Symptoms"and "Quality". Again, training employed 80% of images and validation and testing the remaining 20%. The full dataset size is 11218 picture segments that are divided into classes the following way:

- Label Quality: 9740 images for class "Field"and 1478 images for class "Low"

- Label Symptoms: 4556 images for class "Symptoms"and 6662 images for class "No Symptoms"

We also applied the CNN models trained using the multi-label approach over the case study dataset for the same reason as in the previous experiment. Finally, we compared the results obtained from the multi- and single-label approaches.

5.3.5 Expanded Dataset

As expected and observed in the experiments, the dataset size influences the classification performance. A small dataset will not present enough samples and variability for training the model to classify the images. However, the local dataset used in the experiments has size limitations, since it was manually collected and annotated.

Therefore, we also performed experiments using an online dataset created in a laboratory combined with our dataset containing crop field images to increase the training dataset size and optimize the model performance. The enlarged dataset shall ensure more samples, maintaining the variability that enables the model to learn the field images characteristics.

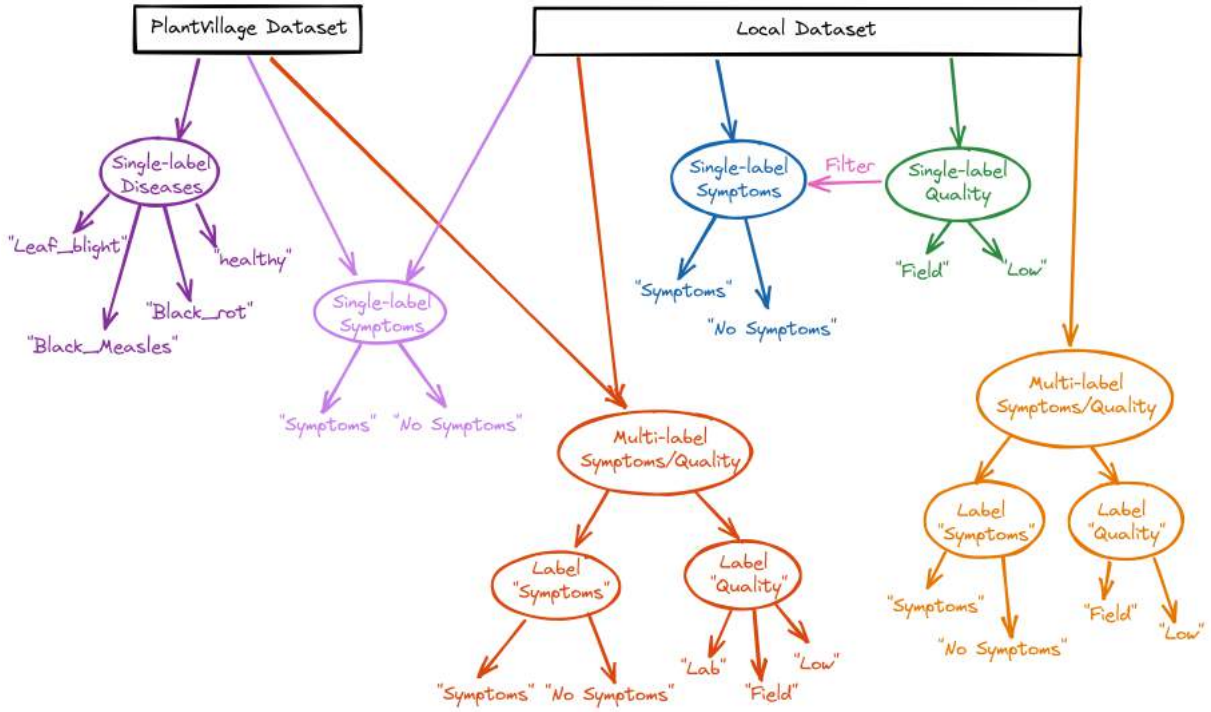
In this experiment, we added the images from the grape crop from the PlantVillage dataset available in Mohanty (2016 (accessed March 1, 2021)) to the local dataset used in the previous experiments. To preserve our standard, we divided the pictures of the PlantVillage dataset into segments, following the same logic used in the local dataset preparation. Since the PlantVillage images use a disease annotation, we mapped their classes into the ones used in our classifier, i.e., “Symptoms” or “No Symptoms”. After the preparation, the expanded dataset comprises 30372 images, being 20306 from the class "Symptoms"and 10066 from "No Symptoms". The expanded dataset was used in experiments both using the single-label and the multi-label technique.

Since the multi-label approach classifies the images regarding their quality, it is necessary to separate the images created in a lab from images collected in the field. Therefore, we included another class, called “Lab”, under the label “Quality”. This class contains all the images created in a laboratory. Their picture quality does not compare to the one exhibited by crop field pictures. The picture number from the PlantVillage dataset amount to 16248, 1692 from the class “No Symptoms” and 14556 from “Symptoms”.

5.3.6 Experiments Summary

In summary, Figure 25 displays the performed experiments in this work. The figure shows the datasets used in each experiment and the output classes from the neural network model. The colors used in the representation of each experiment are the same ones used in the resulting distribution histograms in Chapter 6.

Figure 25 – Diagram representing the performed experiments



Source: the author (2021)

5.4 NEURAL NETWORK IMPLEMENTATION

In all the performed experiments described in Section 5.3, we used the deep learning technique known as CNN. It trains a Convolutional Neural Network, often used to classify images, as in the present application.

During the training, the model might face learning problems of varying severity. In extreme situations, the model exhibits under- or overfitting. Thus, the optimization algorithm monitors the learning progress by validating the model after each epoch (each training iteration updating the internal network parameters). Therefore, a certain percentage of the training dataset serves as a validation set. The model performance over this subset, evaluated after each epoch, will

indicate potential fitting problems. We used 20% of the training set for validation.

We carefully balanced the remaining 80% of the images used for training according to the image number of each class by image rotation and flipping. Augmentation of the picture class with fewer images followed the ratio between the size of the two categories. This technique was used both for symptoms detection and for quality classification.

Table 2 displays the parameters of the CNN used for the single-label experiments (symptoms detection and picture quality classification).

Table 2 – Network Implementation Technical Details

Parameter	Value
Pre-trained Model	ResNet50V2
Activation Function	Sigmoid
Loss Function	Binary Cross-Entropy
Optimizer	Adam
Learning Rate	0.0001
Batch Size	128
Max. Epoch Number	100
Early Stopping	3 epochs
Framework	Tensorflow (SL) / Pytorch (ML)

Source: the author (2021)

It was employed a sigmoid activation function in the output layer. During training, binary cross-entropy served as a loss function. The chosen optimizer was Adam, with a learning rate of 0.0001. We trained the model using a batch size of 128 and a maximum number of epochs of 100. However, an early stopping criterion was employed to avoid overfitting, with a threshold of 3 consecutive epochs with no progress. We implemented the classification module using the Tensorflow framework for the Single-label experiments and the Pytorch framework for the Multi-label experiments. All the implementation technical details are summarized in Table 2.

For performance evaluation, the trained model classified the images of the test set (not used in training). For the symptoms detection experiment, the prediction for each model returns a probability between 0 and 1, indicating the likelihood of the segment showing disease symptoms. A threshold value of 0.5 separates the classes, i.e., a prediction below 0.5 indicates an image segment with “No Symptoms", otherwise with “Symptoms". We derived several performance metrics from the predicted confusion matrix: accuracy, precision (both for the positive and negative classes), recall, specificity, F1-score, and average precision for the performance assessment. However, in general, we optimized the recall metric.

Since quality classification is also a binary classification problem, the predicted output probability varies between 0 and 1, mapped into the two classes using a threshold. Class "0"indicates high-quality images belonging to the class "Field", and class "1"represents the low-quality pictures or the class "Low". In this case, the optimized metric was precision. For both experiment scenarios, we also observed the accuracy metric, which involves all elements from the confusion matrix.

In the case of simultaneously classifying symptoms and quality, we adapted a CNN classifier to conduct multi-label classification. The neural network originated from a ResNet50 pre-trained model by connecting two linear fully-connected layers, one for each label. The chosen activation function was sigmoid, the loss function was binary cross-entropy (for both labels), and the batch size was 128. The optimizer was Adam, with a learning rate of 0.001. The maximum number of epochs for each pipeline execution was 75. The complete flow was repeated for $N = 50$ iterations, ensuring model variations.

The output probability from each fully-connected layer maps each inference image into the predicted classes. The total loss follows from the losses for each label. After the prediction, we computed the confusion matrix for each picture label and determined the evaluation metrics accuracy, precision, recall, specificity, and f1-score. However, each picture label demands a particular metric for optimization, the recall metric for the “Symptoms” label and the precision metric for the “Quality” label.

5.5 EXPERIMENTAL SETUP

We executed all experiments on a Standard NC6s v3 virtual machine hosted at Microsoft Azure, comprising six vCPUs and 112 GB of memory. The processor is an Intel Xeon CPU E5-2690 v4, with a clock of 2.60 GHz. It also has a Tesla V100 PCIe 16GB GPU. We performed model training and testing on the same platform.

Regarding the executed single-label experiments, we performed a total of 30 tests, including the initial hyperparameter tuning analysis. Each experiment implemented a pipeline script that repeated the train and test flow for N times (usually $N = 50$). Concerning the multi-label approach, we executed a total of 10 experiments. Similarly to the single-label technique, we also used a pipeline script that repeated the flow for $N = 50$ times. Each experiment flow took about 30 minutes to complete, so a complete experiment usually required more than 24 hours to finish, on average.

6 RESULTS

The following chapter presents the results obtained from the performed experiments. Section 6.1 shows the results from the baseline experiments executed for comparison, Section 6.2 presents the results from the experiments using single-label technique, and Section 6.3 describes the outcomes from the Multi-label experiments. Finally, Section 6.4 summarizes and discusses the obtained results.

During the performed experiments, we used three different techniques to evaluate them. As explained in Section 5.3, for each experiment, we implemented a pipeline that trained $N = 50$ different variations of the same model, randomly splitting the dataset in different ways. In order to evaluate and compare their performances, we used three different techniques. First, after the training and application of each variation over its validation dataset, we organized them in a ranked list according to their prediction recall value. The evaluation metrics from the variations that produced the best, mean, and worst recall are displayed on a table. This enables not only the comparison between three different model variations, but also the correlation between some metrics at the expense of others. Another technique was to display all the $N = 50$ values of a chosen evaluation metric in a single histogram, in order to observe the distribution for different value ranges. This analysis was performed for the relevant metrics (in the symptoms detection experiment, they were recall and accuracy, and in the quality classification, precision, and accuracy). This technique was used in two cases. The first one was when each variation was applied over its own validation dataset, which generates more realistic values. The second one is related to the application of the $N = 50$ model variations over the complete local dataset. Even though these results cannot be considered the actual performance of the variants (since they include data used in the training), they allow a comparison of different experiments over the same dataset.

6.1 BASELINE EXPERIMENTS

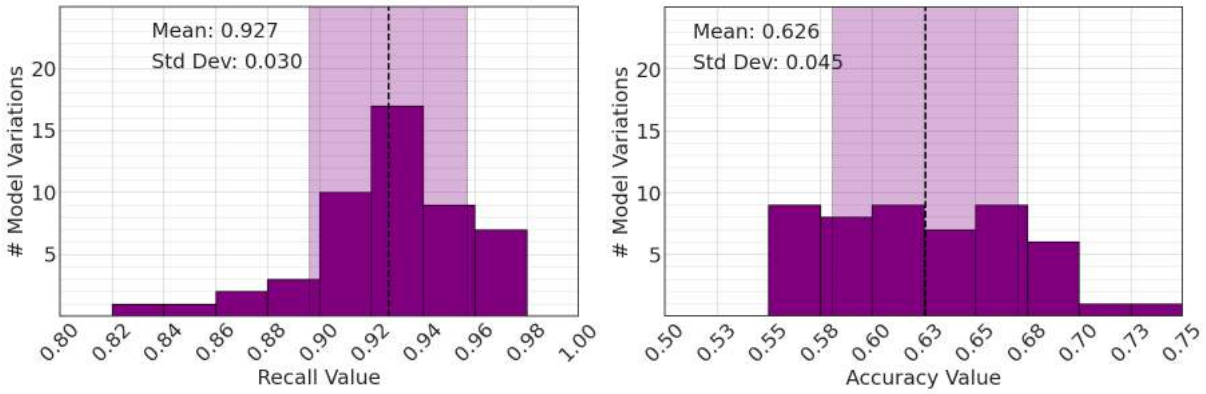
In the first performed experiment, described in Section 5.1, the model learned to identify three different plant diseases and distinguish them from healthy leaves. The train-test pipeline was executed for $N = 50$ iterations, performing a random train/test dataset separation at each of them, using the PlantVillage dataset.

After the 50 executions, we analyzed the F1-score obtained at each one, and we calculated the best, the worst, the mean, and the standard deviation for this metric. The best model variation in this ranking produced a value of 100%, the closest to the mean obtained 94.4%, and the worst achieved a 77.78% F1-score value for grape diseases classification. The mean value for all the $N = 50$ variations was 96.67%, and the standard deviation was 4.4%.

After the complete pipeline process, all the 50 variants were applied to classify the full local dataset, assessing performance with authentic field images. Since the annotation of the dataset does not include disease classes, we mapped the predicted disease class into one of the two classes of the label "Symptoms Presence"("Symptoms"and "No Symptoms").

The model predictions were used to calculate the recall and the accuracy over the local dataset. Figure 26 shows a histogram that represents the distribution of the recall value (26a) and the accuracy value (26b) of the 50 model variations over the complete local dataset.

Figure 26 – Distribution histograms from baseline experiment over complete dataset



(a) Recall distribution histogram

(b) Accuracy distribution histogram

Source: the author (2021)

The best result for the recall over the complete local dataset was 97%, and the worst was 83.45%. The mean value for the 50 variants was 92.7%, and the standard deviation was 3.0%. Among the accuracy results, the maximum value was 73.14%, and the minimum was 55.55%, obtaining a 62.6% mean and a 4.5% standard deviation for the distribution. The low accuracy has to be expected because the variations did not learn the characteristics of crop field images. The result emphasizes the need for a training dataset closer to reality.

6.2 SINGLE-LABEL EXPERIMENTS

6.2.1 Hyperparameters Tuning

The main goal of the following experiment was to classify the images in one of the two classes, "Symptoms"or "No Symptoms". In the first step, before the classifier training, it was performed a hyperparameter tuning, following the pipeline presented in Figure 23. For clarity purposes, we divided the results of this process into two different parts.

Table 3 summarizes the findings of the first group of assessments. A minimum number of repetitions of dataset splits and model training are required to explore the complete dataset reliably. We found that $N = 50$ is sufficient for the collected data. The mean recall and standard deviation do not change significantly for larger N, but the computation time increases. Another interesting finding concerns the dataset division strategy. Data division into train and test sets can occur before or after separating the leaves pictures into segments. The former approach (before segmentation, defined in the table as the split category “Leaves") ensures that all the parts from a given picture end up in the same group. However, constructing the training set on segment level (splitting after segmentation, defined in the table as the split category “Segments") increases flexibility, reducing the standard deviation and the gap between maximum and minimum recall. Finally, the first group of assessments also evaluated the desirable dataset size. We tested the use of the complete dataset, as opposed to the use of a partial one. As shown in Tab. 3, a training set that is too small leads to a reduced recall mean and maximum value, independently of the division strategy. However, the 11218 images collected in a dedicated field expedition are sufficient to train models with good performance and high evaluation metrics. The table also shows the mean values of the precision, accuracy, and F1-score measured in the experiments.

Table 4 summarizes the findings of the second group of assessments, which are related to the optimization of the training algorithm. Inside the second group, some experiments determine the best weight for each of the two classes during training. Choosing different weights for the classes allows balancing without resampling. In other words, weights can tell the model to pay more attention to the instances of a particular category. Tab. 4 shows that attributing weight “1"to class “0"(“No symptoms") and weight “2"to class “1"(“Symptoms") gives a higher recall mean, and therefore we chose these values to be used in the experiments. This result is plausible because the importance to the class “Symptoms"is increased, and we

Table 3 – Assessment results - Group 1

Exec. #	Split Category	Dataset Size	Mean Recall	Max Recall	Min Recall	Mean Prec.	Mean Acc.	Mean F1
10	Segments	11218	0.907	0.945	0.876	0.785	0.856	0.836
50	Segments	11218	0.916	0.949	0.862	0.791	0.86	0.841
50	Leaves	11218	0.912	0.979	0.834	0.743	0.813	0.805
50	Leaves	3650	0.863	0.923	0.770	0.927	0.858	0.882
50	Segments	3650	0.863	0.910	0.811	0.931	0.871	0.894

Source: the author (2021)

are mainly trying to avoid false negatives.

Table 4 – Assessment results - Group 2

Weight "0"	Weight "1"	Mean Recall	Std. Dev. Recall	Max Recall	Min Recall	Mean Prec.	Mean Acc.	Mean F1
1	2	0.928	0.024	0.969	0.857	0.7875	0.847	0.839
2	1	0.905	0.027	0.957	0.836	0.818	0.863	0.849
1	1	0.911	0.026	0.962	0.832	0.819	0.861	0.849
1	1.34	0.919	0.024	0.955	0.844	0.829	0.873	0.859

Source: the author (2021)

In addition, we analyzed the evaluation metric and the classification threshold, not shown in the table. Some models learned using the accuracy metric instead of the recall metric. However, they produced too many false negatives. This constitutes an undesirable finding, since segments classified showing no symptoms will not be called for further inspection by a human expert. We also tested different threshold values for the classification, but found that the standard choice of 0.5 gives the most reliable prediction, producing better recall and accuracy values.

6.2.2 Symptoms Detection

After identifying the parameter values that produce the best results for the proposed application, we performed the training and prediction of $N = 50$ model variations in the determined specific conditions. In this case, the digital assistant shall forward to experts image segments that possibly show symptoms for further inspection. Thus, ideally, picture segments revealing no disease symptoms will be classified correctly (TN) and not analyzed. Still, any picture seg-

ment possibly showing disease symptoms need to pass through inspection, i.e., should not be classified erroneously (FN). Therefore, during training, an essential evaluation metric is the recall metric, computed as shown in equation 2.4, since it must be used when the FN cost is high. The results from the application of the trained variants over its validation datasets were ranked, and the best, mean (variation whose recall value is the closest to the mean), and worst variations were determined based on the recall values. Nevertheless, other evaluation metrics complemented the analysis, such as accuracy, precision, F1-score, and average precision. Table 5 below shows the predicted metrics for the best, the mean, and the worst model variations obtained in this experiment.

Table 5 – Prediction Results - Symptoms Detection

Model ID	Recall	Acc.	Prec. "0"	Prec. "1"	TNR	F1 Score	Avg. Prec.
43	0.965	0.815	0.965	0.704	0.707	0.814	0.694
16	0.924	0.857	0.937	0.777	0.808	0.844	0.750
1	0.870	0.883	0.911	0.843	0.892	0.856	0.785

Source: the author (2021)

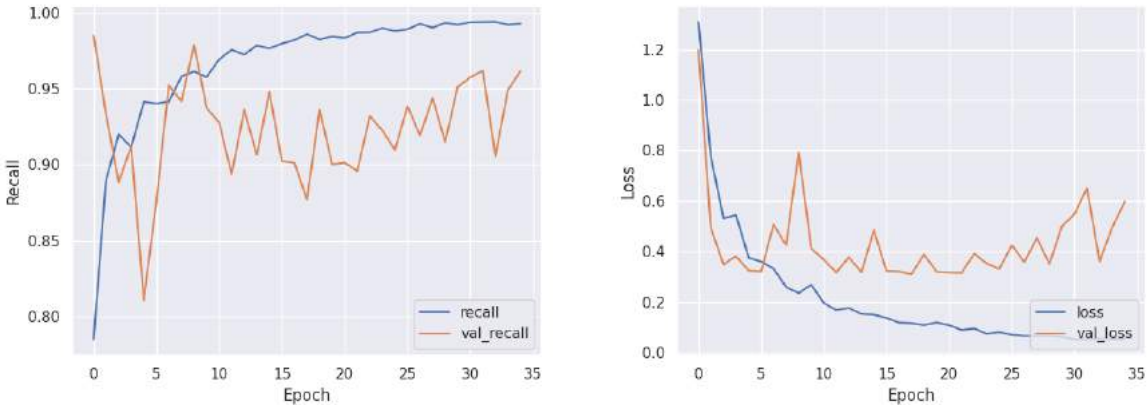
It is possible to notice in Tab. 5 the relationship between the performance metrics and how they are interconnected. When the metrics based on the positive class produce better results (recall and precision for positive class), the metrics related to the negative class yield a poorer performance and vice-versa. An expected outcome since the model was more likely to classify an input image as belonging to the positive class.

Our goal is to find the model variation that maximizes the recall metric, since we aim to identify all possible positive samples. In the case of this experiment, it was the one identified as 43. Figure 27 displays the recall (27a) and loss (27b) graphs obtained in the train process for the model variation 43.

Figure 28 displays the histograms that represent the metric distribution in this experiment. Each value corresponding to the application of the trained model variation over its respective validation dataset is distributed in the histogram. Histogram 28a is related to the recall, chosen as the metric that we want to optimize, and histogram 28b relates to the accuracy, calculated using equation 2.1 and involving all the elements in the confusion matrix.

We also decided to apply all trained variations over the complete local dataset for classifying all pictures. The goal is to establish a standard comparison among all the trained variants in this study (independent of the used datasets), although train and test pictures are not well

Figure 27 – Training results for best model variation from symptoms detection experiment



(a) Training recall (b) Training loss

Source: the author (2021)

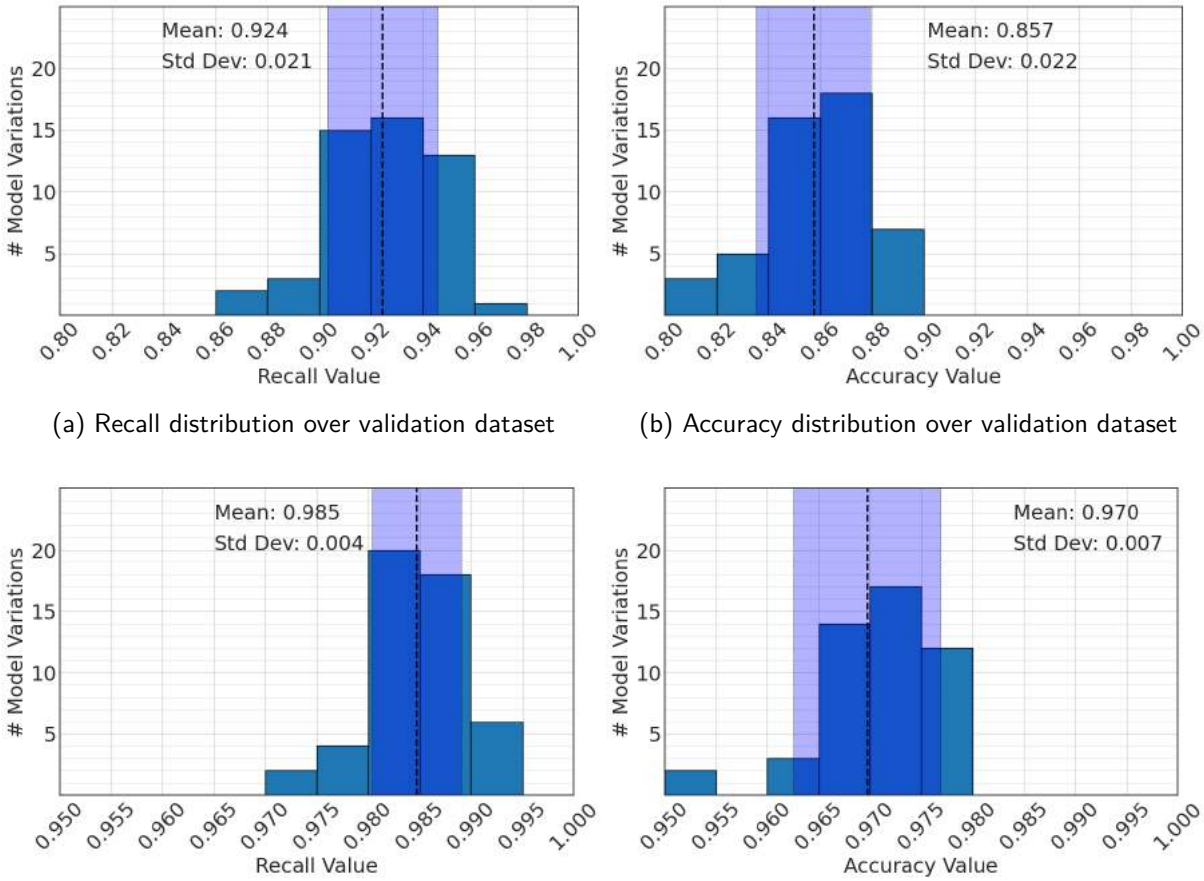
separated. Histogram 28c shows the recall distribution and histogram 28d shows the accuracy distribution when the model variations are applied over the complete local dataset. It is possible to notice that both the recall and the accuracy distribution are over 95%, showing the variants' consistency. When compared to the results obtained in the baseline experiment, we can observe an increase of 22% in the best accuracy, and of 35% in the mean one.

6.2.3 Case Study Experiments

As stated in Section 5.2, the case study dataset is composed of images from both classes (showing symptoms or not) for various light, focus, and background conditions, as well as different severity levels of symptoms. There are 104 images in this dataset, 54 from the class "Symptoms"and 50 from "No Symptoms", not used in any training. Applying the models from different experiments over this dataset allows a more specific evaluation. This analysis involves both the model characteristics and the features in the test dataset that are important for the classification. It will also allow a closer inspection of field image prediction results.

We applied the model variations trained in the experiment described in 5.3.2 over the case study dataset. Initially, the analysis of the predictions revealed some characteristics present in the inference images that influence the classification accuracy. For example, when the area occupied by the background is small, it is easier for the model to classify the leaf segment correctly. However, the classification is more challenging for the model when the sunlight is too bright; or the leaf image does not have enough contrast. Thus, a carefully composed and

Figure 28 – Histogram of prediction metrics for symptoms detection



(a) Recall distribution over validation dataset

(b) Accuracy distribution over validation dataset

(c) Recall distribution over complete dataset

(d) Accuracy distribution over complete dataset

Source: the author (2021)

sufficiently large training dataset is essential.

The experiment also enabled a more focused analysis of the neural network characteristics. We applied all the $N = 50$ variations (trained to detect disease symptoms) over the images selected for the study case dataset. The mean recall obtained was 0.923, and the standard deviation was 0.028. Table 6 show the entries to the confusion matrices, as well as the recall and accuracy metrics, for three selected variants. They give the best, the closest to the mean, and the worst recall, respectively, for the case study dataset.

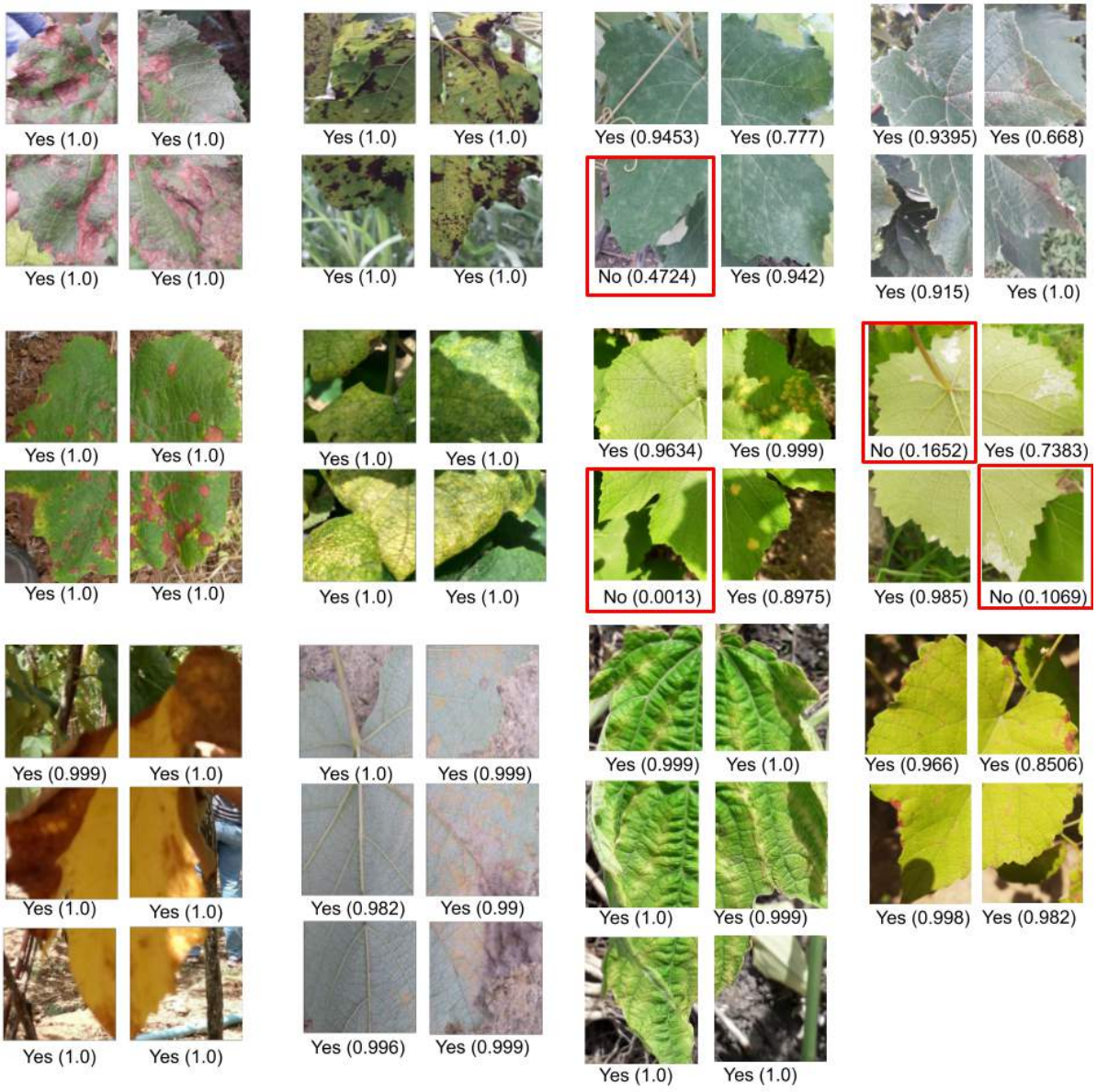
Table 6 – Prediction Results - Symptoms Detection over Case Study Dataset

Model ID	TP	FN	FP	TN	Recall	Accuracy
11	52	2	21	29	0.963	0.779
0	50	4	17	33	0.926	0.798
20	46	8	14	36	0.852	0.789

Source: the author (2021)

We chose the model variation identified as 0 (that produced the mid recall) to be observed in more detail when applied over the case study set. Figure 29 displays the prediction results from this model over the images labeled as showing symptoms. We marked the FN in red for better visualization. Since we aim to increase the recall, we want to find as many "positive samples"as possible, i.e., do not miss any image showing symptoms. In the results, it is possible to notice that most segments were correctly classified, and the probability varies with how apparent the disease symptoms are in the picture. It is also possible to see that, in all the cases where the model misclassified the segments, the symptoms covered only a minor part of the frame.

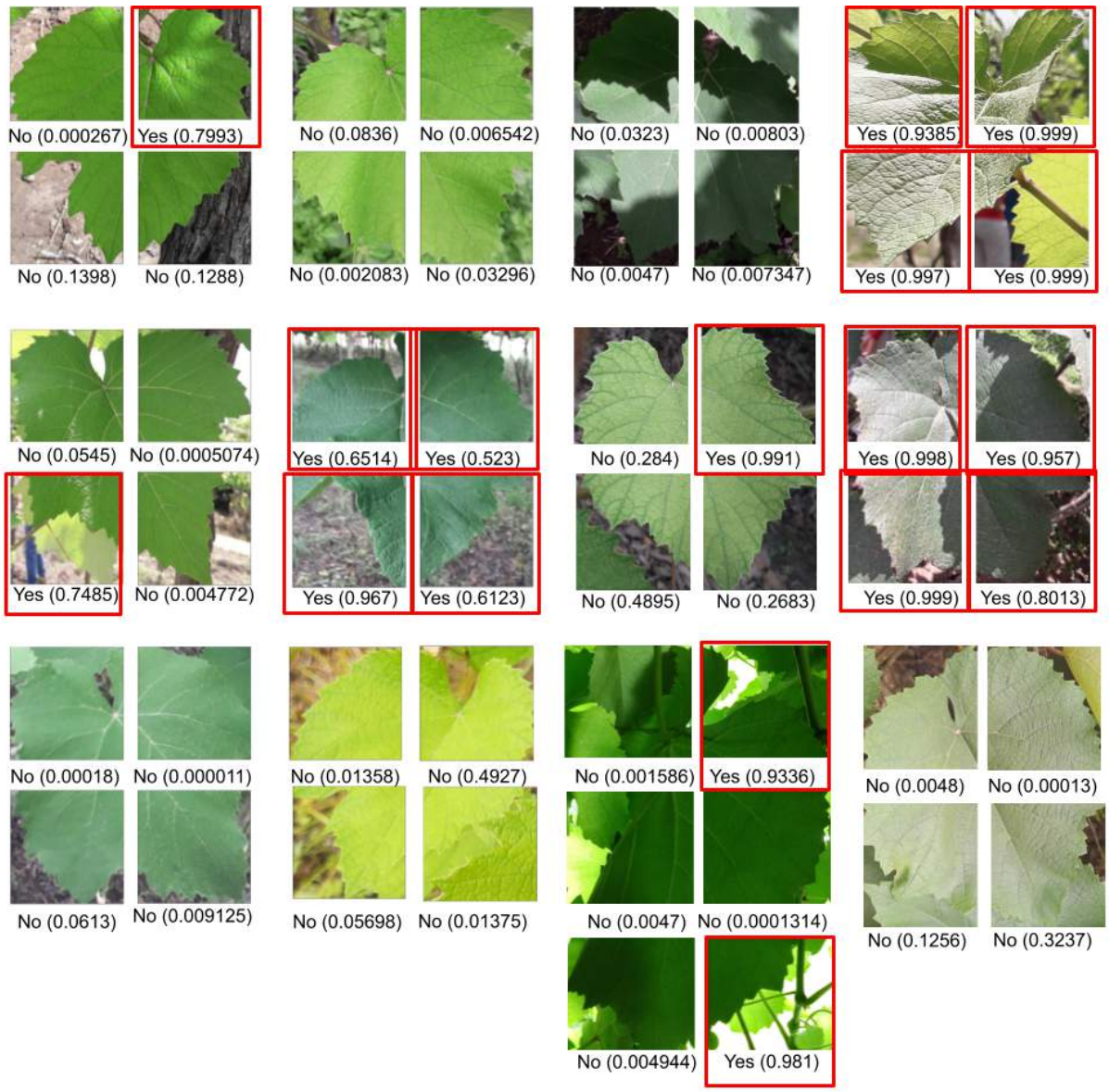
Figure 29 – Case Study prediction results for class "Symptoms"



Source: the author (2021)

On the other hand, Figure 30 shows the results when the same variation classifies the images from the class "No Symptoms", with FP marked in red. The results are worse than the positive class, but it is possible to notice specific characteristics that produced errors in the images. For example, they often show a dominant background area or too much sunlight and shadows.

Figure 30 – Case Study prediction results for class "No Symptoms"



Source: the author (2021)

The outcomes from this experiment enabled some analysis related to the images used in the dataset. The first point is the need for a diverse and large dataset. An ideal dataset balances the classes and includes sufficient picture variations, which will allow the neural network model to interpret the pictures in various conditions.

Another inference we made from the examination is that not all the pictures present enough quality for classifying the presence of the symptoms. Some of them may be out of focus, for example, and the classification may not be accurate. Therefore, it is necessary to filter the images of low quality.

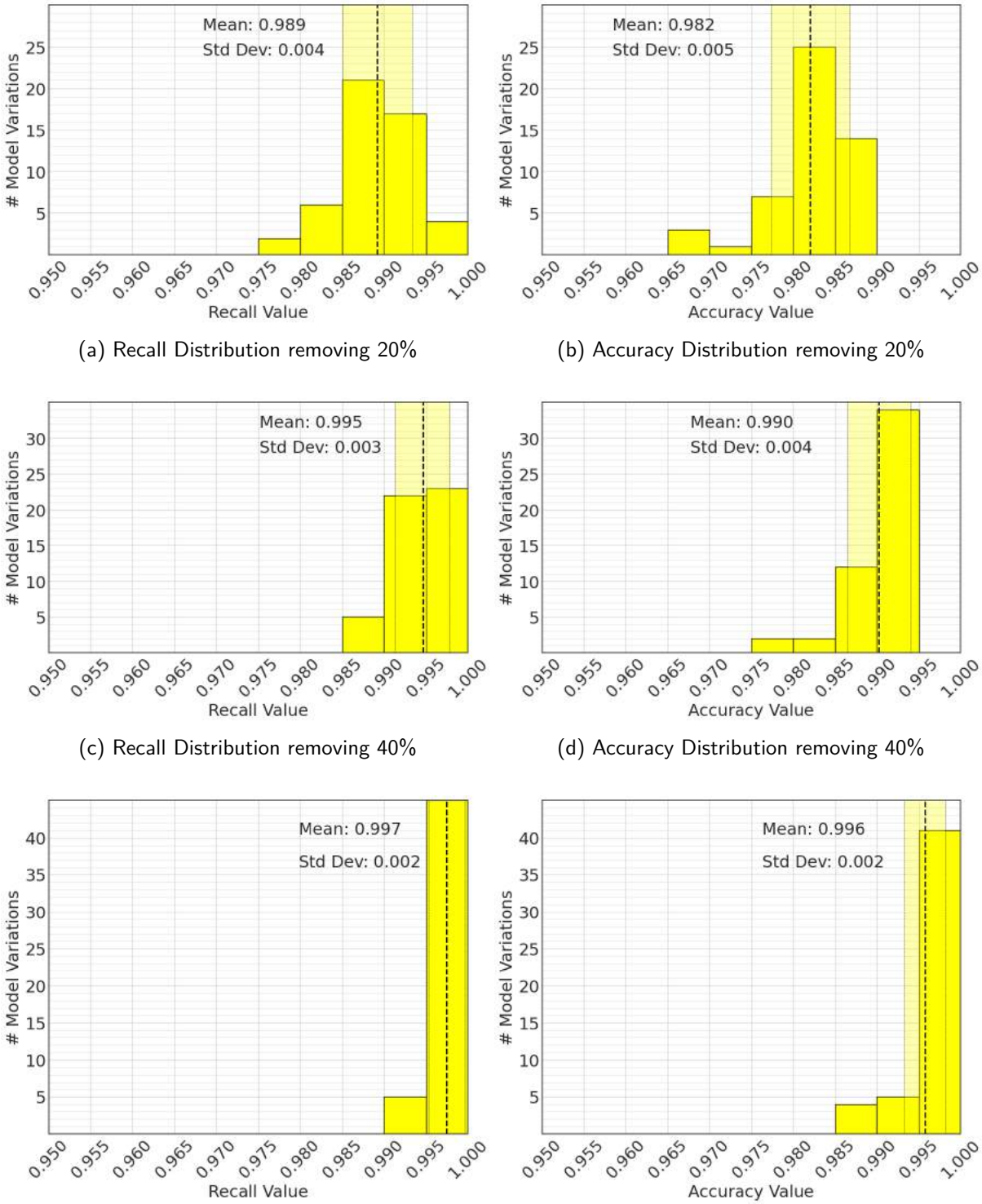
6.2.4 Misclassified Images Filter

As a consequence of the previously described experiments, we decided to analyze the images that pose a challenge for classification. The case study dataset allowed us to observe these images more specifically. Figures 29 and 30 explicitly indicate the FN and FP, respectively, and we can notice that they present in general some common characteristics. However, to identify the challenging pictures in the complete dataset, we sought the incorrectly classified images and noted how many models committed the mistake. This way, we could determine the leaf pictures most often misclassified. The objective, in this case, was to evaluate the model behavior when these challenging pictures were removed from the test dataset and discover a way to filter these kinds of images. For the complete local dataset, classification yielded 2921 FNs or FPs. From this group, 2254 images, or around 77% of them, are FP, which means they do not present symptoms. The remaining 667 pictures, or 23%, show symptoms, being classified as FN. This result shows that most misclassified images are FP, which is what we expected since the risk, in this case, is less severe.

Figure 31 displays the recall and accuracy distribution histogram over each group of images evaluated. Firstly, we identified the 20% of the pictures that were most times wrongly labeled. This group consisted of 584 images, that were removed from the complete local dataset. Then, all the $N = 50$ variations were applied over the remaining images, whose recall results are shown in Figure 31a and accuracy in Figure 31b. The same sequence was applied removing 40% of the misclassified images, which results in 1168 photos. The results can be seen in Figure 31c (recall) and Figure 31d (accuracy). In the next step, the 60% worst images (or 1752 pictures) were removed, resulting in the graphs 31e and 31f. Then, 80% of the false images (2336) were separated, and the histograms are displayed in Figure 31g and Figure 31h. Finally, when all the 2921 images were eliminated, as expected, the results are as shown in Figure 31i and Figure 31j.

The resulting histograms shown in Figure 31 show the expected outcome of this experiment. The mean values for recall and accuracy increase, and the standard deviation decreases when

Figure 31 – Histogram of prediction metrics removing false images



(a) Recall Distribution removing 20%

(b) Accuracy Distribution removing 20%

(c) Recall Distribution removing 40%

(d) Accuracy Distribution removing 40%

(e) Recall Distribution removing 60%

(f) Accuracy Distribution removing 60%

Source: the author (2021)

removing the most challenging images from the test dataset. Figure 32 displays a graph showing the variation of the mean recall value over the different groups (32a), and the variation of the standard deviation of the recall value (32b). It allows us to endorse performance improvement

Figure 31 – Histogram of prediction metrics removing false images

(g) Recall Distribution removing 80%

(h) Accuracy Distribution removing 80%

(i) Recall Distribution removing 100%

(j) Accuracy Distribution removing 100%

Source: the author (2021)

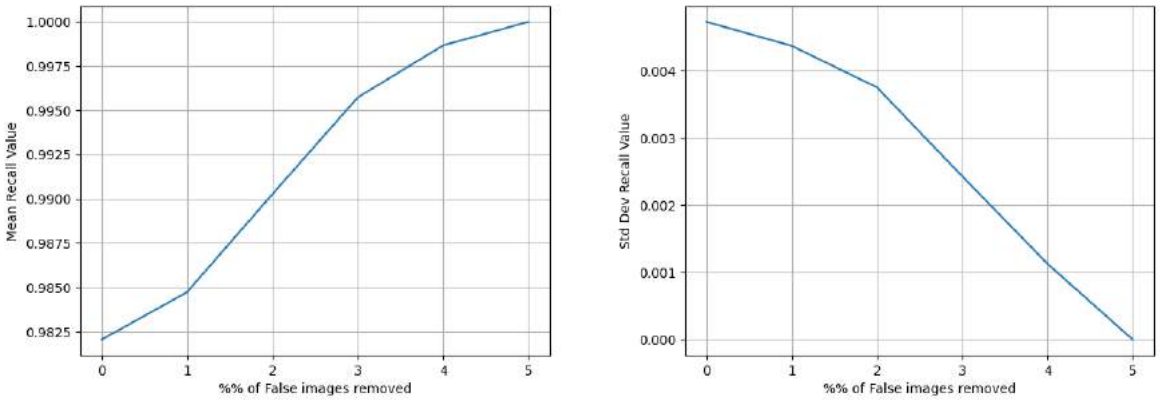
when removing the challenging image.

6.2.5 Pictures Quality Classification

The previous Section 6.2.4 revealed the importance and the potential of filtering the challenging pictures before the classification. However, the system application that is classifying the crop field images does not know which images are the demanding ones. Thus, we tried to filter the low-quality leaf pictures before executing the disease symptoms detection to guarantee that the photographs to be classified are not challenging to the system.

Therefore, the next performed experiment was the picture classification regarding their

Figure 32 – Recall values variation over different groups



(a) Mean recall variation (b) Standard deviation recall variation

Source: the author (2021)

quality, as described in the Subsection 5.3.3. We performed ourselves the image annotation into high or low quality, so it is essential to highlight the use of a subjective criterion. We tried to encompass the field images into these two classes ("Field", with high-quality images, and "Low", with low-quality images) for classification. In order to standardize this classification, we created a pattern for which image characteristics constitute low-quality features. We decided that images out of focus, presenting the leaf entirely on shadows, or taken against sunlight must be classified as low-quality pictures, belonging to the class "Low". Consequently, the remaining ones are classified as high-quality, belonging to the category "Field".

We repeated the same pipeline used in the previous experiments, which executed the training and prediction for $N = 50$ iterations. For each iteration, the obtained trained model variation predicted the classes of the validation set. Since the objective here is the picture quality prediction, we chose precision as the most relevant metric for optimization and analysis, but we also computed the other evaluation metrics.

Table 7 shows the evaluation metrics obtained in the prediction for the best, the mean, and the worst variation from this experiment, respectively. The variants ranking is based on the precision metric, which depends on a weighted average between the two classes.

This experiment shows the discrepancy between the samples from the two classes. The metrics related to the negative class (in this case, class "Field", containing crop field images of good quality) scored higher values. Metrics associated with the positive picture class (in this case, class "Low", containing crop field images of low quality) achieved poorer results. This observation is a direct consequence of the fact that the created local dataset is composed, in

Table 7 – Prediction Results - Pictures Quality Classification

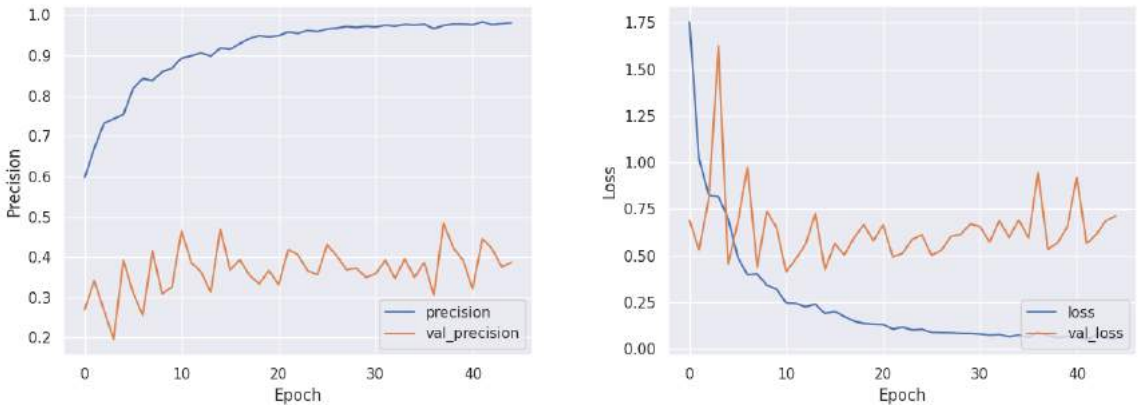
Model ID	Precision	Accuracy	Recall	TNR	F1	Avg. Prec.
0	0.870	0.812	0.652	0.836	0.478	0.292
42	0.854	0.802	0.500	0.851	0.415	0.248
2	0.830	0.799	0.563	0.835	0.428	0.253

Source: the author (2021)

its majority, of images of good quality. The classification of picture quality was not yet a topic when the students and phytopathology experts initially collected the pictures. Thus, there was an effort to include only high-quality leaf pictures in the dataset.

Nevertheless, the analysis of the resulting values allowed us to choose a model variation to represent this experiment. In this case, it was the one that produced the best "precision"value, identified as variant 0. The graphs in Figure 33 display the precision (33a) and loss (33b) produced during the training of this variation.

Figure 33 – Training results for best model from picture quality classification experiment



(a) Training precision (b) Training loss

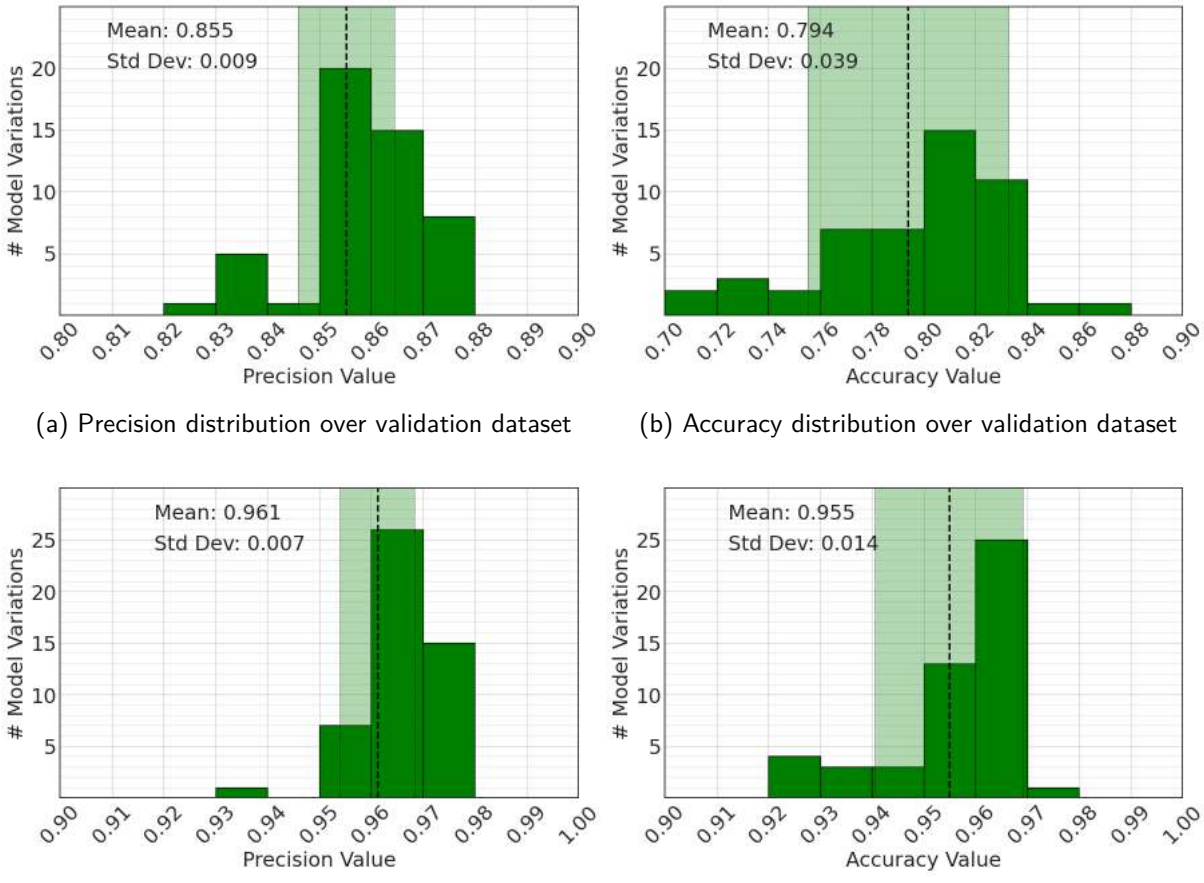
Source: the author (2021)

Another result related to the present experiment is the histogram that displays the distribution of the precision from the $N = 50$ variations, each one over its validation dataset (Figure 34). Figure 34a shows the distribution of the weighted precision (chosen as the key metric), and figure 34b shows the distribution of the accuracy value, whose computation involves also the other elements from the confusion matrix. The resulting histograms show that the measured precision metric closely follows a normal distribution. Even though the accuracy distribution is more sparse, all values exceed 70% accuracy.

Finally, the model variations trained in this experiment were also applied over the complete

dataset to compare their performances over the same data. Figure 34 also shows the distribution of the precision (34c) and the accuracy (34d) metrics in this scenario. It is possible to see that both the precision and the accuracy distribution achieve better results than when applied over the validation datasets, maintaining the distribution format. Therefore, the results values in both cases behave as expected, enabling the comparison between the variations. The ones that produce the best, the worst, and the mean precision values are the same from the application over the validation datasets.

Figure 34 – Histogram of prediction metrics for quality classification



(a) Precision distribution over validation dataset

(b) Accuracy distribution over validation dataset

(c) Precision distribution over complete dataset

(d) Accuracy distribution over complete dataset

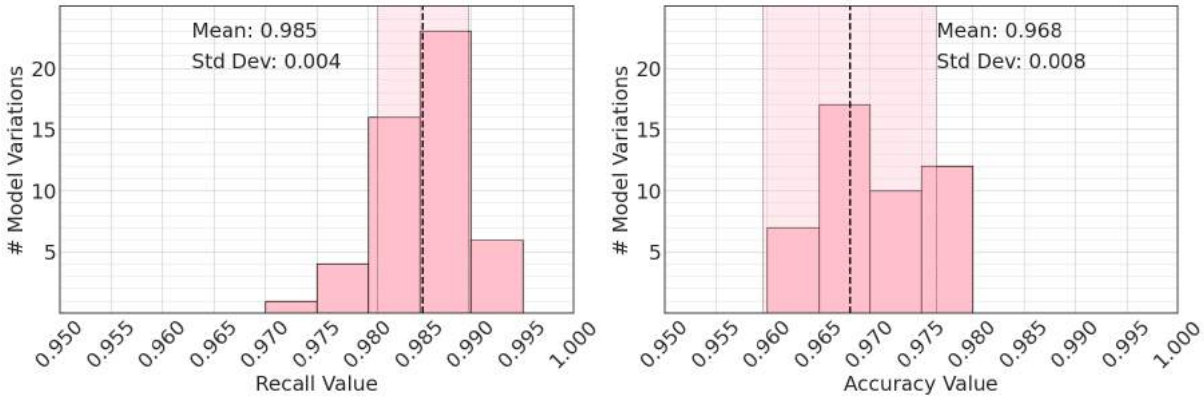
Source: the author (2021)

6.2.6 Quality Classification Filter

After the division of the local test dataset into two classes ("Field"and "Low"), the previously developed classifier analyzed the images from each quality class regarding the presence of disease symptoms separately.

Initially, only the images with better quality (class "Field") were used as test dataset, and the $N = 50$ model variations trained in the experiment described in 5.3.2 were applied over them. In this case, the test set contained 9740 images, 3281 from the class "Symptoms", and 6459 from the class "No Symptoms". Figure 35 shows the distribution histogram of the obtained results, both from the recall metric (Figure 35a), and from the accuracy metric (Figure 35b).

Figure 35 – Histogram of prediction metrics over high quality field images

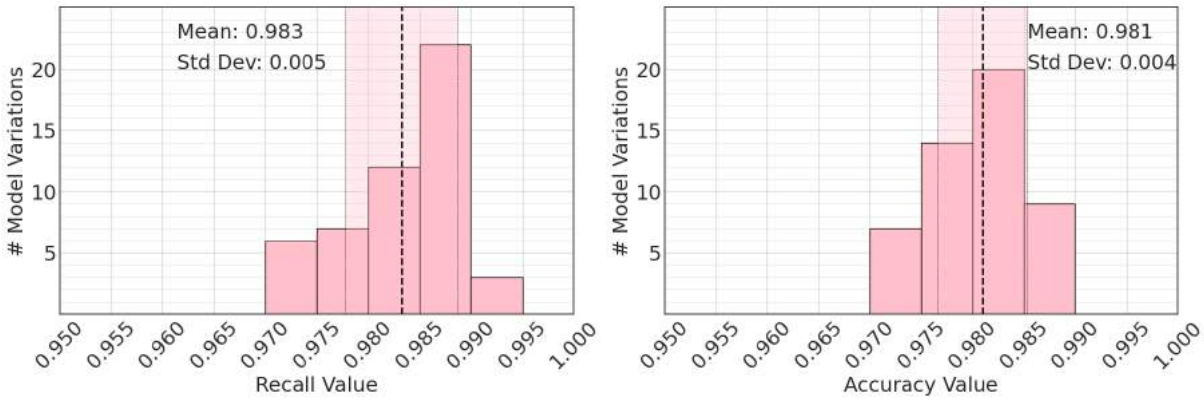


(a) Recall distribution (b) Accuracy distribution

Source: the author (2021)

Then, in a similar way, we applied the same variations over the images classified with lower quality (class "Low"). This set contains 1478 images, being 1275 from the class "Symptoms"and 203 from the class "No Symptoms". Figure 36 displays the evaluation metrics distribution, both from the recall values (36a) and from the accuracy values (36b).

Figure 36 – Histogram of prediction metrics over low quality field images



(a) Recall distribution (b) Accuracy distribution

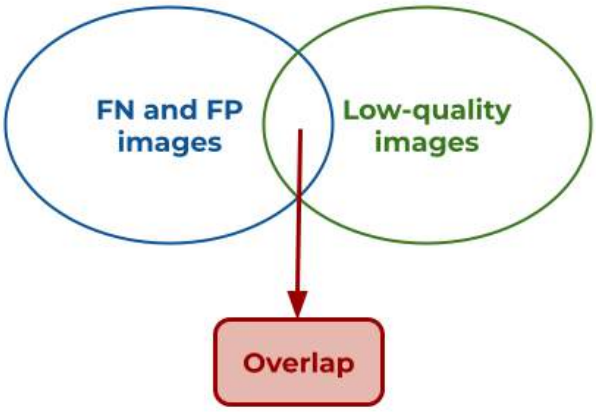
Source: the author (2021)

The results corroborate increased metric values in both cases compared to the values

obtained for the complete local dataset. However, contrary to the expected, the low-quality class does not necessarily generate worse results than the high-quality class.

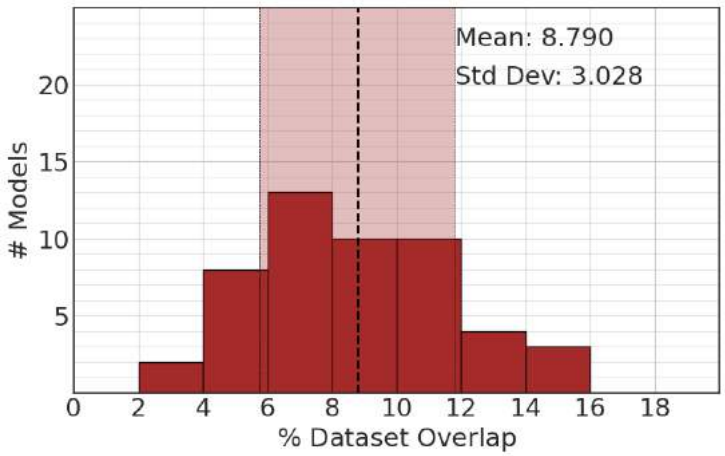
To better understand the reasons, we analyzed the images that the neural network classifies to be of low quality, trying to correlate them to the leaf pictures erroneously classified by the system described in 6.2.4. For each of the $N = 50$ repetitions, we calculated the overlap between the images labeled as having low quality and the previously obtained FNs or FPs. Figure 37 shows a diagram explaining which images are represented in this overlap.

Figure 37 – Diagram representing overlap images



Source: the author (2021)

Figure 38 – Overlap distribution



Source: the author (2021)

Figure 38 shows the histogram that represents the distribution of these percentage values. The mean overlap value, as observed in this histogram, is 8.79%. Considering this value, that means that, if we use the quality classification model as a filter, we eliminate about 9% of the misclassified images. In this case, we expect a worse improvement in symptoms detection than

the one expressed in graphs 31a and 31b, that represent a removal of 20% of the FN and FP. Since this overlap is small compared to what we were expecting, the result points to the fact that we can still improve the quality annotation of the leaf pictures.

6.2.7 Expanded Dataset

One of the main results obtained from the performed experiments is the importance of a large and diverse dataset. In this work, we created our dataset, collecting, annotating, and preparing leaf pictures obtained from a crop field. This dataset enabled the training process to learn the variations and characteristics of photos taken in an actual crop field. However, since the manually built local dataset has size limitations, a more extensive dataset would generate better results.

To overcome this situation, we experimented using a combination of all available images as the dataset. That means that we used not only the pictures from the created local dataset but also the available PlantVillage dataset, created by Mohanty & Salathé (2016). These images originated from a laboratory, and therefore do not present the field conditions and distracting background, but they may help the system improve its performance.

The complete dataset used in this experiment has 30372 images, being 24297 training pictures and 6075 validation and test pictures. The experiment was performed the same way as the previous one. We trained $N = 50$ variations of a neural network model and applied each one over the respective test dataset. A comparison of the measured recall identified the best, the mean, and the worst model variation. Table 8 shows the prediction results for the three variants, respectively. These values can be compared to the ones in Table 5 (which refers to the symptoms' detection over the local dataset). Therefore, we can observe an increase of 2%, on average, for the Recall metric, and an average increase of 10% for the Accuracy metric.

Table 8 – Prediction Results - Symptoms Detection - Expanded Dataset

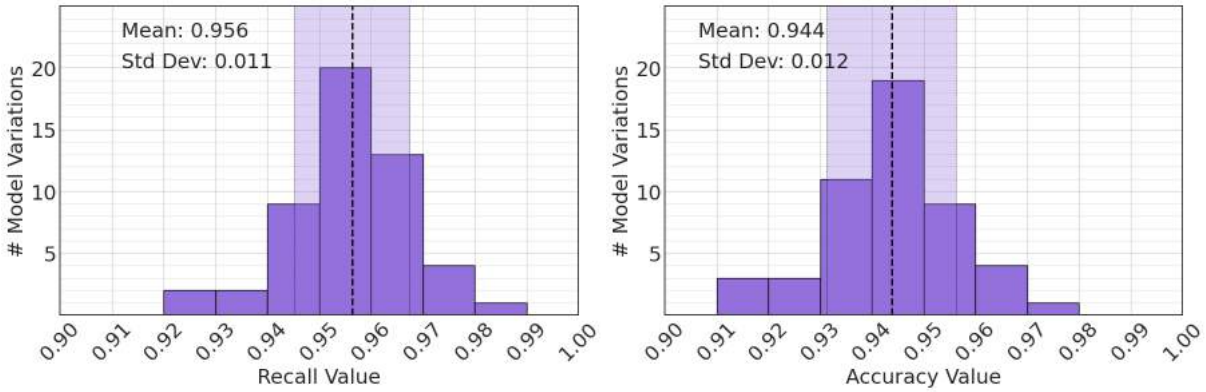
Model ID	Recall	Acc.	Prec. "0"	Prec. "1"	TNR	F1	Avg. Prec.
19	0.970	0.943	0.929	0.949	0.881	0.960	0.942
23	0.944	0.949	0.882	0.982	0.959	0.962	0.966
7	0.915	0.936	0.835	0.992	0.983	0.952	0.967

Source: the author (2021)

After the training, we applied the neural network model variations over the complete local

dataset (only the images collected in the field). A comparison with the initial experiment allows determining if the performance has improved. Figure 39 shows the distribution for the predicted evaluation metrics, both for the recall (39a) and the accuracy metric (39b). Comparing these histograms with the ones in Figure 28, it is possible to notice that, even though the values are close, the previous experiment presented a better performance (about 2% better, in average).

Figure 39 – Histogram of prediction metrics over complete dataset for training with expanded dataset



(a) Recall distribution (b) Accuracy distribution

Source: the author (2021)

6.3 MULTI-LABEL EXPERIMENTS

As explained in Section 5.3.4, we decided to use the multi-label approach to classify the input images concerning both the disease symptoms presence and their quality with a single trained model. In this application, the implemented neural network involves two labels: one to identify if the image shows disease symptoms (whose possible classes are “Symptoms” and “No Symptoms”), and the other one to classify the picture quality (whose classes are "Low"and "Field").

The experiments performed using this technique were similar to those executed using the single-label method, only adapted to this situation. Concerning the datasets, we initially employed only the local dataset, later adding pictures from the available PlantVillage dataset (MOHANTY, 2016 (accessed March 1, 2021)).

6.3.1 Local Dataset

Firstly, the training of $N = 50$ multi-label model variations over the local dataset followed the pipeline execution presented in Section 5.3.2. Section 5.3.4 describes the hyperparameters and variables used in the training process. Since the multi-label technique produces more than one output value (in this case, two), there is a need to observe more than one evaluation metric. We decided to continue optimizing the recall for the label "Symptoms Presence". Similarly to the single-label experiment, we chose precision optimization for the picture label "Quality". However, we computed all the metrics described in section 2.7 for both picture labels.

All the $N = 50$ variations were ranked according to their performance. Since the primary goal of this work is to detect disease symptoms, the ranking uses the recall value. Table 9 displays the result prediction metrics related to the label "Symptoms Presence" for the best, the mean, and the worst variation obtained in this experiment. In the last column of the table, we also display the precision obtained in the prediction from the label "Picture Quality" for each variant.

Table 9 – Multi-label Prediction Results - Symptoms Detection over local Dataset

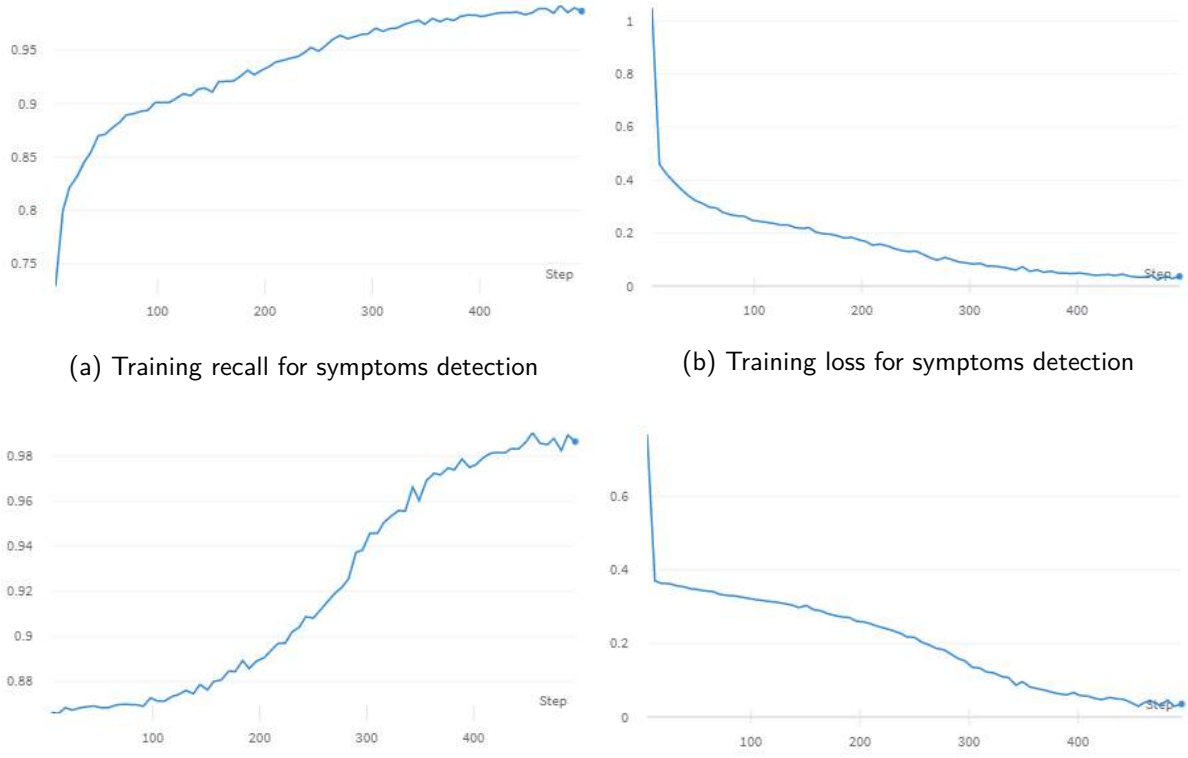
Model ID	Recall	Acc.	Prec. "0"	Prec. "1"	TNR	F1	Prec. Qual.
42	0.949	0.913	0.894	0.942	0.959	0.897	0.867
10	0.903	0.898	0.932	0.851	0.894	0.876	0.852
11	0.852	0.891	0.901	0.874	0.923	0.857	0.883

Source: the author (2021)

The values presented in Tab. 9 show the relationship between the different performance metrics. Model variation 42, for example, generates a better recall for symptoms detection, while model variation 11 obtains a better precision value for the photo label "Quality". The table also shows that the values for the second label are all above 85%, indicating that this technique generates consistent results for both picture labels.

As explained before, even though the model variations classify the inputs according to two different labels, we chose to use the recall from the label "Symptoms Presence"to detect the "best"variant. In this case, the best one was identified as variation 42. Figure 40 displays the training metrics graphs for the multi-label approach. It shows the training recall (40a) and loss (40b) for the label "Symptoms Presence", and also the training precision (40c) and loss (40d) for the label "Photo Quality", all related to model variation 42.

Figure 40 – Training metrics for Multi-label approach



(a) Training recall for symptoms detection

(b) Training loss for symptoms detection

(c) Training precision for quality classification

(d) Training loss for quality classification

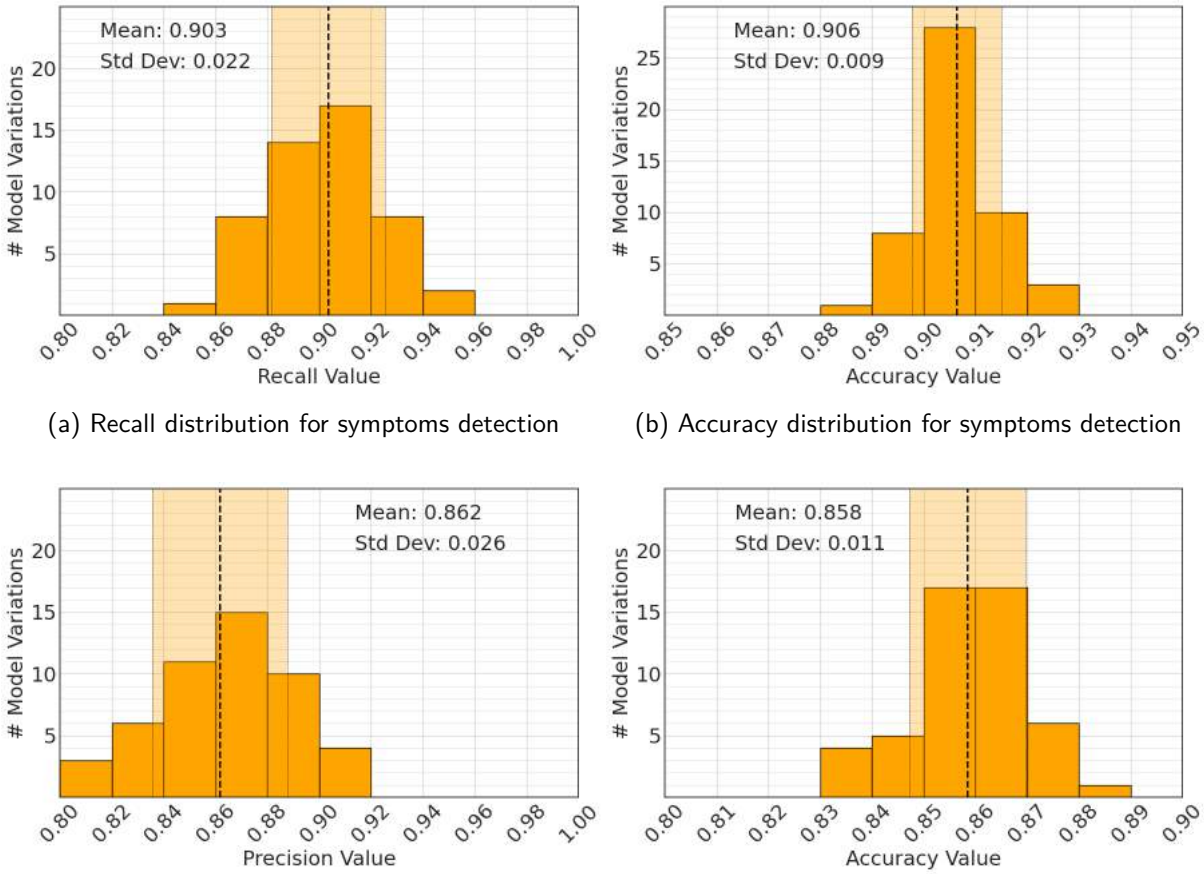
Source: the author (2021)

Figure 41 shows a histogram of the predicted evaluation metrics distribution obtained by the application of the $N = 50$ variations over their respective test sets. In a similar way to the Single-label experiment, Fig. 41a shows the distribution of the recall value related to the disease symptoms detection, and Fig. 41b displays the accuracy distribution. Regarding the quality classification, Fig. 41c shows the distribution of the precision value and Fig. 41d displays the accuracy distribution.

The histograms displayed in Fig. 41 indicate that the multi-label technique achieves better performance, in general, when compared with the single-label approach. For example, the mean recall concerning the label "Symptoms Presence" is smaller, but the accuracy metric (for symptoms detection) is 5% higher, on average, and the precision calculated for "Photo Quality" is also 5% higher (in the best model variation). This outcome means that the multi-label classification can achieve approximately the same or even better results using a single model to perform picture classifications of both labels.

Then, the trained $N = 50$ model variations were also applied to the complete local dataset, in order to establish a comparison over the same test dataset. Figure 42 shows the results distribution for this scenario, displaying the recall distribution histogram (42a) and the accuracy

Figure 41 – Histogram of prediction metrics for Multi-label Classification



(a) Recall distribution for symptoms detection

(b) Accuracy distribution for symptoms detection

(c) Precision distribution for quality classification

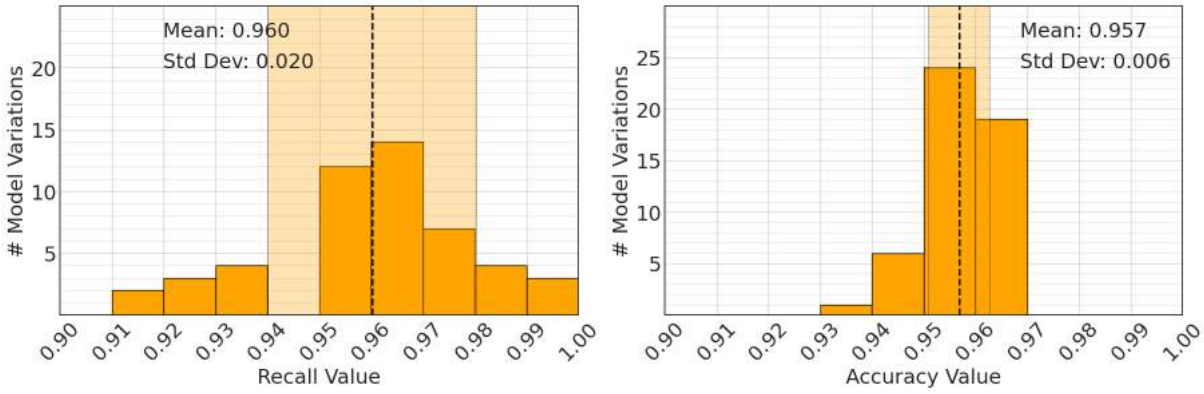
(d) Accuracy distribution for quality classification

Source: the author (2021)

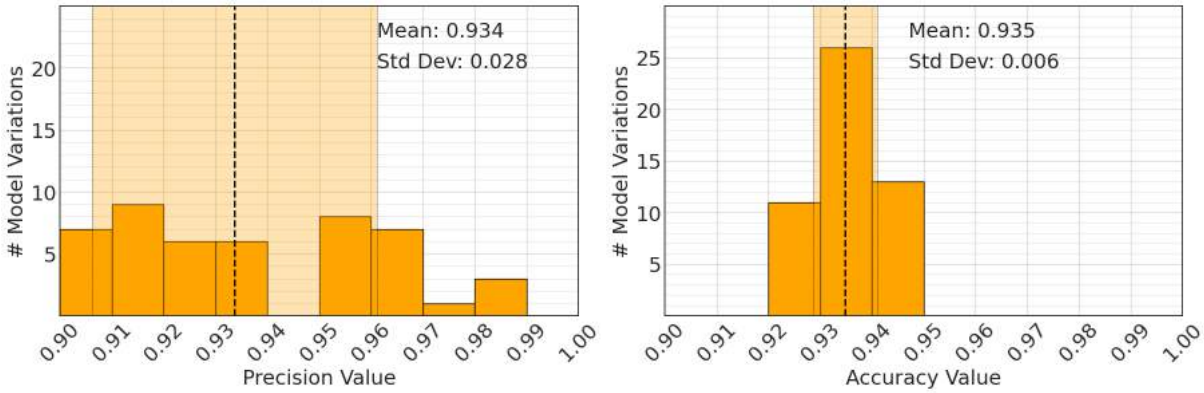
distribution (42b) for symptoms detection, as well as the precision distribution (42c) and the accuracy distribution (42d) for image classification.

Finally, we separated the images classified by the label "Picture Quality"as belonging to the class "Field"(crop field pictures that presented a high quality), and we applied the trained variations over it. The metrics distributions are shown in Figure 43. The recall distribution is displayed in Fig. 43a and the accuracy distribution in Fig. 43b. When compared to the previous results, shown in Fig. 42 (the same models applied over the complete local dataset), they reveal the improvement of the symptoms detection recall when this "filter"is applied. In other words, removing the images classified by the system as presenting low quality improves the symptoms detector's performance. In this case, both the recall and the accuracy metric increased for symptoms detection, the latter by over 3%.

Figure 42 – Histogram of prediction metrics for Multi-label Classification over complete dataset



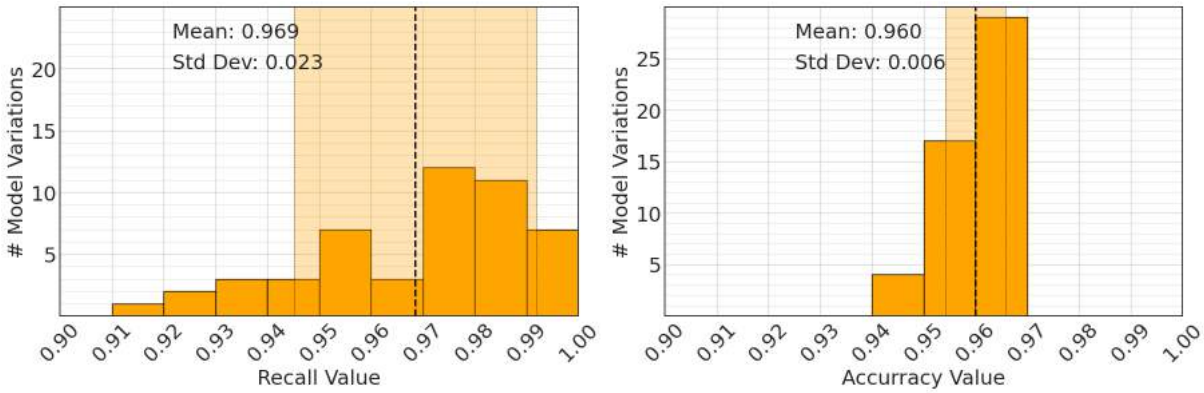
(a) Recall distribution for symptoms detection (b) Accuracy distribution for symptoms detection



(c) Precision distribution for quality classification (d) Accuracy distribution for quality classification

Source: the author (2021)

Figure 43 – Prediction results distribution for symptoms detection over high quality images



(a) Recall distribution (b) Accuracy distribution

Source: the author (2021)

6.3.2 Case Study Experiments

Following the same reasoning used with the single-label approach, we also applied the trained model variations over the Case Study dataset. The images are the same ones used

in the experiment reported in Sec. 6.2.3 and not used in any training. Sec. 6.3.1 describes the corresponding variations. The objective here was to analyze the model performance over different images in more detail, observing the resulting predictions of specific images.

The results from the application of $N = 50$ variations are shown in Table 10. The prediction results from the best, mean, and worst variants, respectively, are displayed in the table. Compared with the results displayed in Table 6, we can observe that the recall is lower, but the accuracy is higher.

Table 10 – Prediction Results - Symptoms Detection over Case Study Dataset using Multi-label Approach

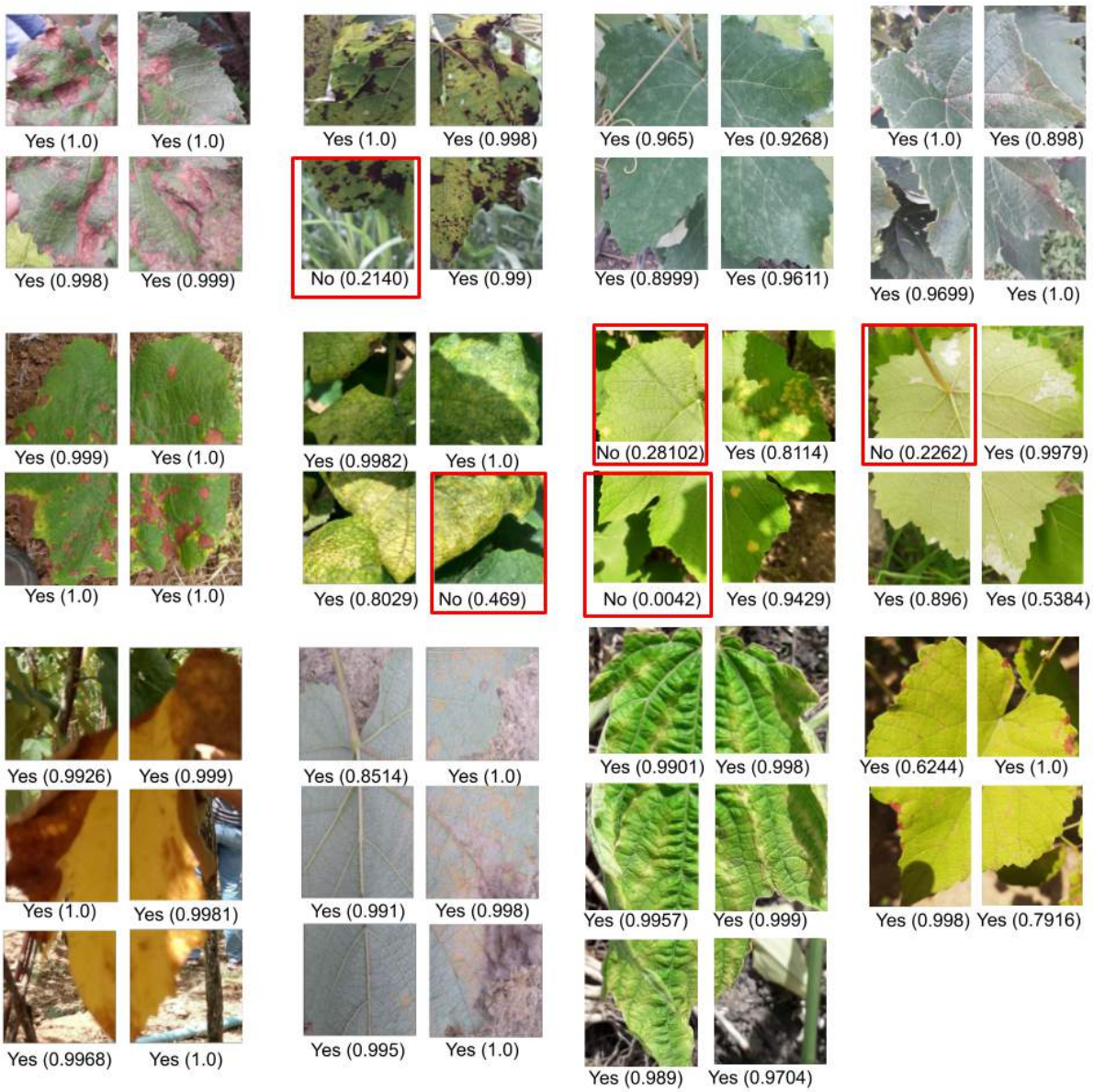
Model ID	TP	FN	FP	TN	Recall	Accuracy
45	49	5	7	43	0.907	0.885
5	47	7	9	41	0.870	0.846
27	43	11	12	38	0.796	0.779

Source: the author (2021)

We chose the best model variation (identified as 45) to discuss the results and the output probability generated for each case study image. Figure 44 shows the output results for every image in this set using the specified variation. Red frames highlight the FNs, and we can observe some conditions leading to false classification, e.g., bright sunlight, a distracting large-area background, or a limited leaf region showing symptoms.

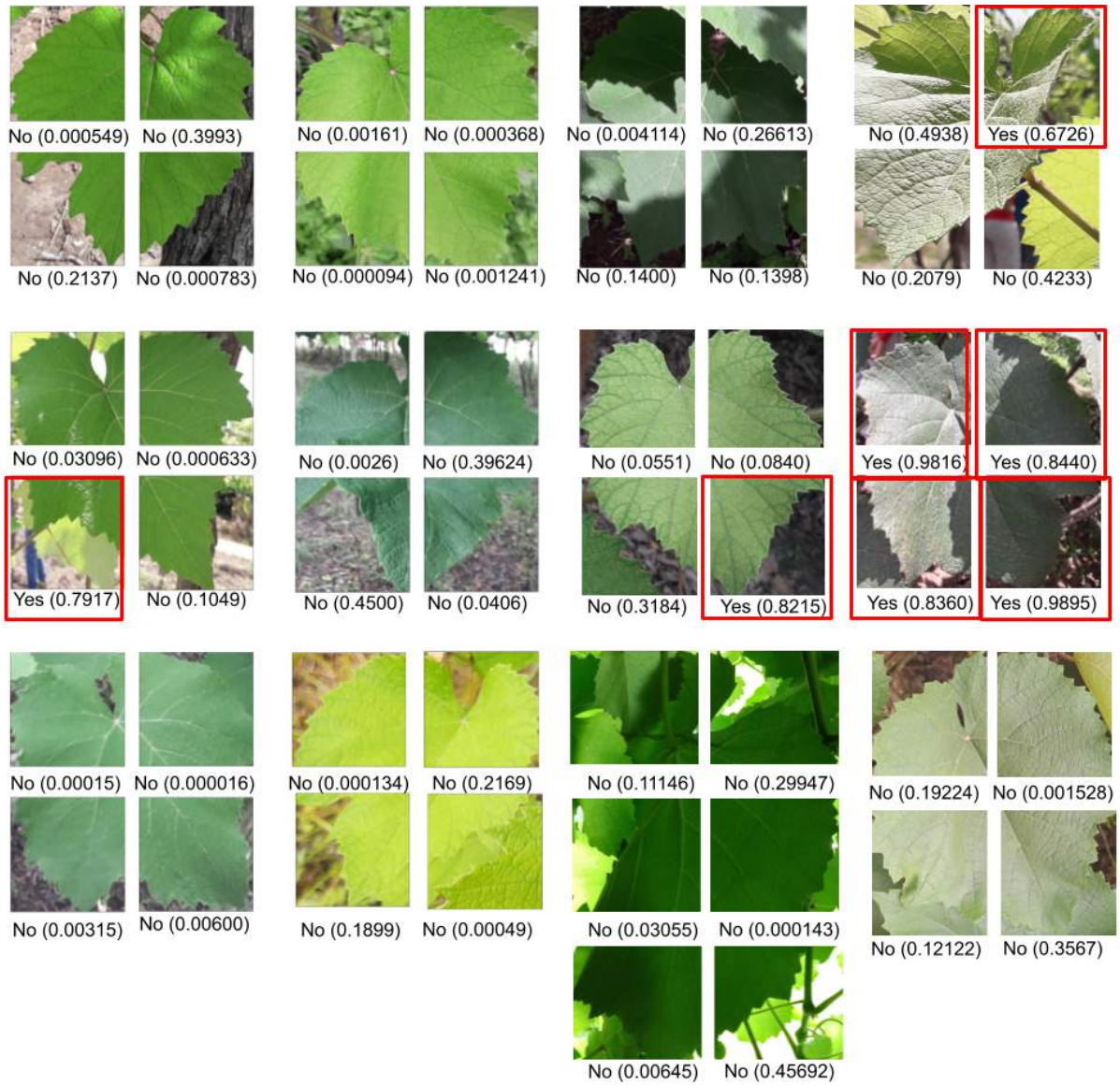
On the other hand, Figure 45 shows the results when applying variant 45 over the images from the class "No Symptoms". Red frames mark FPs. Comparing these results with the ones obtained using the single-label approach, the improvement for this class is clear, causing the accuracy to be higher in this case.

Figure 44 – Case Study prediction results for class "Symptoms"using Multi-label approach



Source: the author (2021)

Figure 45 – Case Study prediction results for class "No Symptoms"using Multi-label approach



Source: the author (2021)

6.3.3 Expanded Dataset

Since we are adding images created in a lab, we decided to add a new class to the label that classifies the image quality. This label, called "Lab", ensures that the system learns the difference between these images and those taken in a crop field. The training and testing procedure was the same as in the previous experiments.

Comparison of the $N = 50$ evaluation results at the end of the training gave the best, the mean, and the worst model variation considering the recall for the "Symptoms Presence"label. Table 11 show the prediction results for these three variants, respectively. We can infer from this outcome that the model trained with the expanded dataset performed better than the one trained with the local dataset only (Table 9).

Table 11 – Multi-label Prediction Results - Symptoms Detection over expanded Dataset

Model ID	Recall	Acc.	Prec. "0"	Prec. "1"	TNR	F1	Prec. Qual.
22	0.976	0.970	0.950	0.980	0.958	0.978	0.948
14	0.963	0.962	0.910	0.991	0.981	0.971	0.934
9	0.953	0.962	0.913	0.989	0.979	0.971	0.925

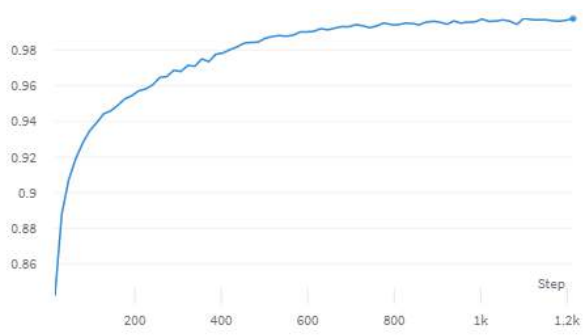
Source: the author (2021)

Figure 46 shows the training graphs from the model variation considered the best in this experiment (referred as variation 22). Fig. 46a (recall) and 46b (loss) are related to the symptoms detection label. Fig. 46c (precision) and 46d (loss) refer to the photo quality label. Comparing these graphs with the equivalent ones from the multi-label models using only the local dataset (40), we can notice the improvement in this situation.

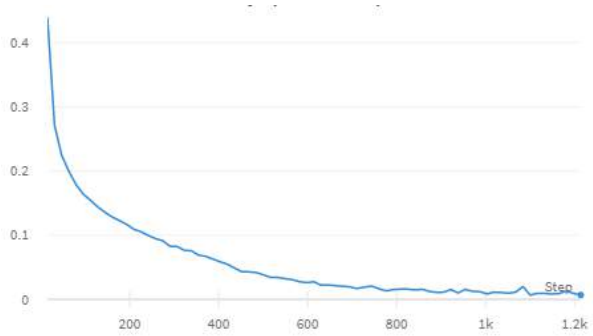
Then, in order to establish a comparison with the other experiments using the same test dataset, we applied the model variations trained in this experiment over the local dataset (only the images collected by us in the field). Figure 47 shows the histogram with the evaluation metric distributions for this scenario. Fig. 47a (recall) and 47b (accuracy) are related to the symptoms detection label. Fig. 47c (precision) and 47d (accuracy) refer to the photo quality label.

We can compare the histograms in Figure 47 with the ones in the previous section, in Figure 42. Regarding detecting disease symptoms, the mean recall increased by more than 2%. Concerning quality classification, even though the mean precision value decreased by 2%, the values are better distributed.

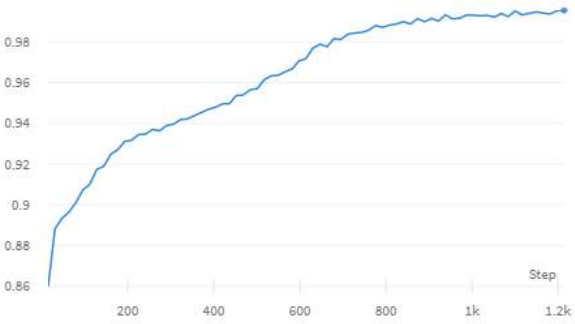
Figure 46 – Training results for Multi-label classification over expanded dataset



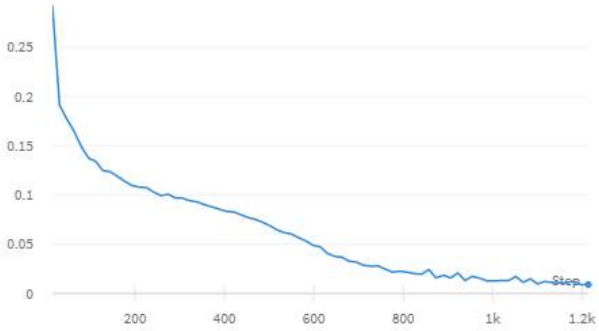
(a) Training recall for symptoms detection



(b) Training loss for symptoms detection



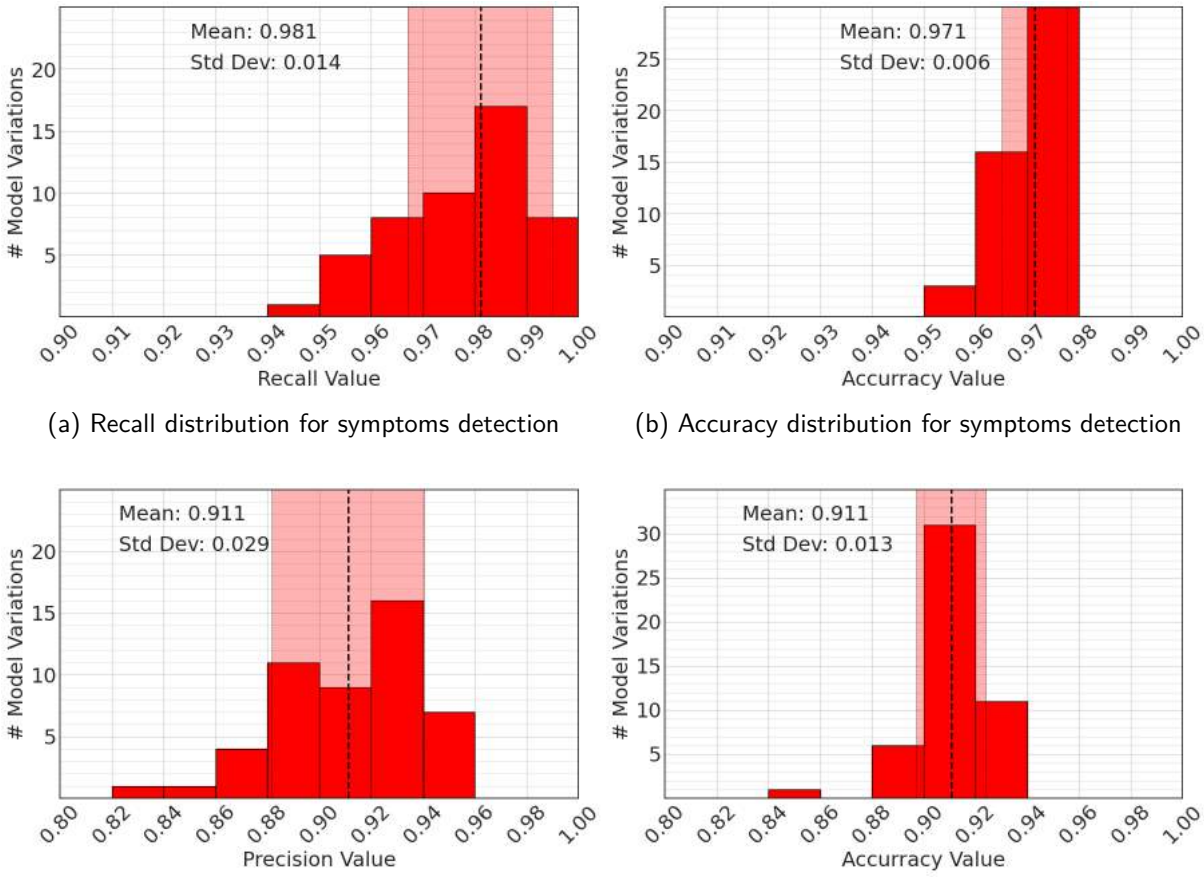
(c) Training precision for quality classification



(d) Training loss for quality classification

Source: the author (2021)

Figure 47 – Prediction results distribution for Multi-label classification over expanded dataset



(a) Recall distribution for symptoms detection

(b) Accuracy distribution for symptoms detection

(c) Precision distribution for quality classification

(d) Accuracy distribution for quality classification

Source: the author (2021)

6.4 EXPERIMENTS RESULTS SUMMARY

We conclude the exposure of the performed experiments and their obtained results by highlighting some of them. In this section, we will present the best models from four tests that best summarize the outcome of this work. These models obtained the best recall values in the ranked list of each one of these experiments: single-label (both using local and expanded dataset), and multi-label (both using local and expanded dataset). Table 12 presents the prediction evaluation metrics from these experiments. The quality classification results presented in the first line of the table (Single-label local experiment) refer to the single-label experiment to classify the pictures regarding their quality (not the same experiment as for symptoms detection).

Table 12 – Prediction Results - Experiments Summary

Experiment	**Recall Sympt.**	**Acc. Sympt.**	**Prec. Sympt.**	**TNR Sympt.**	**F1 Sympt.**	**Prec. Qual.**	**Acc. Qual.**
Single-Label Local	0.965	0.815	0.835	0.707	0.814	0.87	0.812
Single-Label Expanded	0.970	0.943	0.939	0.881	0.960	-	-
Multi-Label Local	0.949	0.913	0.918	0.959	0.897	0.867	0.877
Multi-Label Expanded	0.976	0.970	0.965	0.958	0.978	0.948	0.935

Source: the author (2021)

As expected, the training using the expanded dataset generated the best results, see Table 12. Moreover, comparing the single-label and multi-label approaches with the local dataset, we can observe that, even though the first one resulted in a better recall when detecting symptoms, the other metrics were more balanced in the multi-label experiments. However, when comparing the results using the expanded dataset, we can observe that the multi-label technique achieved better performance in all the evaluated metrics. In addition, regarding the quality classification experiment, the accuracy achieved with a multi-label proposal was higher (maintaining the precision value), whereas the multi-label experiment with an expanded dataset achieved better values for both metrics.

Thus, we can state that this technique presents more advantages in our application. From the complete analysis, we can conclude that the model trained with the multi-label approach

using the expanded dataset obtained the best performance, producing remarkable results in this application.

7 CONCLUSION

The monograph presented an approach based on deep learning and CNNs to detect disease symptoms in crop leaves images. The system is part of a digital platform developed in partnership with the Phytosanitary Clinic of Pernambuco (CliFiPe). The platform aims to establish communication and help phytopathology experts offer assistance to smallholder farmers. A central platform component is a mobile phone application that allows the producer to take pictures of plant leaves. These images are saved in our database and sent to the Symptoms Detection System. The picture segments revealing the most pronounced disease symptoms are suggested to the expert. The classifier shall also ask the farmer to retake a picture of insufficient quality and inform about preventive actions. Data provided by the farmer help improve the system classification performance. Specifically, this dissertation trains and develops the Symptoms Detection System.

The module developed here receives an input leaf image and aims to classify if it detects the presence of disease symptoms. Before the classification, the input image is cropped and centralized (to remove the surplus background) and then divided into segments. The objective here is to create a standard in the input images, force the system to learn the characteristics of different parts of a plant leaf, and augment the training dataset.

We created our local dataset since datasets are not available that include plant leaves pictures taken in a crop field exhibiting natural conditions. We collected 3289 grape leaves images from plantations located in Pernambuco state. Phytopathology experts from CliFiPe annotated the leaf pictures regarding the presence of disease symptoms. Pre-processing prepared the leaf pictures for network model training and prediction.

Initially, a CNN was trained and tested with the grape images from the PlantVillage dataset to implement a multi-class classifier for three grape diseases and healthy leaves. We applied the trained model over the local dataset, establishing a baseline control for the system to be developed.

We implemented a pipeline flow for model training to generate distinct models, performing a grid search to find the neural network hyperparameters that produce the best result. The determined parameters empowered a single-label neural network to detect disease symptoms over segments of grape leaf pictures. The trained models were applied over the respective test datasets, achieving a mean recall of 92.4% and a maximum of 96.5%. The mean obtai-

ned accuracy was 85.7%. In addition, the models classified a previously separated small case study dataset, helping to understand the impact of exposure conditions. The experiment revealed some characteristics of the field images that prevented the photo's correct classification. Sunlight, shadows, a significant background area, and pictures out of focus are some of the characteristics that disturb the work of the Symptoms Detection Module.

Filtering such "low-quality" pictures would mitigate the challenge and improve the classifier performance. Thus, we implemented a neural network model trained to classify the quality of crop field images (high or low). Training relied on an unbalanced dataset due to the limited number of low-quality images, possibly influencing the model performance. Nevertheless, the system was able to classify the field images on their quality, achieving a mean precision value of 85.5% and a mean accuracy value of 79.4%. The trained neural network model acted as a filter, and the symptoms classifier needs only run over the resulting images classified as presenting high quality. This experiment achieved a mean recall of 98.5% and a mean accuracy of 96.8%, confirming the improvement in the system performance.

Another measure experimented in this work was using a combination of both the local and the available PlantVillage dataset to train the symptoms detection module. When the models classified the validation dataset, the mean recall value was 94.4%, slightly larger than obtained when training with the local dataset. However, the mean accuracy value was over 94%, which is more than 8% higher than the previous experiment.

We also developed and applied another technique, the multi-label classification. A single CNN serves as a symptoms detector and picture quality classifier. Initially, the images used in training were only from the local dataset. This experiment achieved a mean recall value of 90.3% and an accuracy value of 90.6% for symptoms detection, a precision value of 86.2%, and an accuracy value of 85.8% for quality classification. The trained models were also applied over the case study dataset, obtaining better results than the single-label models. When trained over the expanded dataset (combined local and PlantVillage dataset), the mean recall was 96.3%, and the mean accuracy was 96.2%. The symptoms detection and the picture quality classification achieved a mean precision value of 93.4%. Comparing the results with the previous experiments, we can observe a substantial improvement in the classification performance.

One of the contributions of this work, as explained in Section 1.3.1, is the creation of a labeled local dataset composed of images from an important crop from the region (in this case, grape). The developed system also helps the phytopathology expert offer assistance to smallholder farmers by detecting diseased leaves and alerting the experts in this case.

Furthermore, identifying the symptoms also alerts the farmer of possible disease outbreaks, sharing information that may help other producers in the community. Moreover, the users of the mobile phone application act as citizen scientists. Smallholders take pictures of their crops and provide additional data, helping to build an extensive local database and empowering a wide range of high-performance agricultural classifiers.

The present dissertation resulted in an article about the disease symptom detection system, published in the 2021 Brazilian Conference on Intelligent Systems (BRACIS).

- Barros M..S. et al. (2021) "Supervised Training of a Simple Digital Assistant for a Free Crop Clinic". In: Britto A., Valdivia Delgado K. (eds) Intelligent Systems. BRACIS 2021. Lecture Notes in Computer Science, vol 13074. Springer, Cham. https://doi.org/10.1007/978-3-030-91699-2_12

Table 13 displays a comparison between the proposed approach and the related works presented in Section 3. Further development will aim towards a complete disease classification system. A prerequisite is a more complete and balanced dataset and additional experiments on implementation variations.

Table 13 – Comparative analysis between Related Works and the Proposed Approach

Related Work	Objective	Dataset	Strategy	Variations	Results
(MOHANTY; SALATHé, 2016)	Identify crop and disease through leaf image	PlantVillage, 54306 images, Lab	CNN	Creation of new dataset, Architectures, dataset versions, Dataset split ratio, learning type	99% accuracy over PlantVillage dataset, 31.4% accuracy over field images
(XIE et al., 2020)	Real-time detection for grape leaf diseases	Custom, 4449 images, Lab and Field	Faster DR-IACNN	New architecture, INSE-ResNet, double-RPN	81.1% mAP
(BARBEDO, 2018)	Discover factors influencing plant disease recognition	Digipathos, 50000 images, Lab and Field	CNN	Extended dataset, 3 datset versions, Intrinsic and extrinsic factors	87% accuracy over subdivided dataset
(BARBEDO, 2019)	Plant disease identification from individual lesions and spots	Digipathos, 1567 images, Lab and Field	CNN (GoogLeNet)	Extended dataset, 3 datset versions, Lesions and spots separation	82% accuracy over original dataset, 94% accuracy over extended dataset
(JI et al., 2020)	Crop leaves disease recognition and severity estimation	AIChallenger, 12691 images, Lab	CNNMulti-label	Architectures, multi-label multi-binary technique	94.71% precision 94.70% recall 94.70% F1-Score
(PEREIRA; MORAIS; REIS, 2019)	Identify grapes variety through leaf images	Custom, 224 images, Field	CNN (AlexNet)	Image processing methods, Data augmentation, Transfer Learning, Fine-tuning	77.3% avg. accuracy, 89.1% max accuracy for particular variety
(BARROS, 2021)	Identify disease symptoms through grape leaf images	Custom, 3289 images, Field	CNN (ResNet), Single-label, Multi-label	Custom dataset, Segments division, Quality Filter, Single and Multi-label, Parameter Tuning	96.3% mean recall, 96.2% mean accuracy, 93.4% mean precision (quality classification)

Source: the author (2021)

7.1 FUTURE WORKS

The work developed in this dissertation raises several improvement possibilities:

- Expand the local dataset, including images displaying several disease agents from relevant crops for the region's economy.
- Implement a classifier that is able to identify the disease-causing agents over grape leaves images. This step will enable more complete assistance to smallholder farmers, including treatment measures that target the specific agent.
- Build detectors of disease symptoms for other crops.
- Develop a learning strategy, continuously improving the classifier performance based on the information gathered by the system.
- To embed the symptoms identification module in modules of the general system, as the digital platform or the mobile application, for example.
- Find a quality classification annotation criterion that improves this classifier's performance.
- Compare single-label and multi-label techniques for novel classifiers.
- Combine leaf pictures and other data (like soil characteristics and information provided by the producers) to improve system performance.
- Alert farmers and experts of possible crop disease outbreaks in Pernambuco state.

REFERENCES

A., T. et al. Grapes leaf disease detection using convolutional neural network. *International Journal of Computer Applications*, v. 178, p. 7–11, 06 2019.

ABRAHAM, T.; TODD, A.; ORRINGER, D. A.; LEVENSON, R. Chapter 7 - applications of artificial intelligence for image enhancement in pathology. In: COHEN, S. (Ed.). *Artificial Intelligence and Deep Learning in Pathology*. Elsevier, 2021. p. 119–148. ISBN 978-0-323-67538-3. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780323675383000075>.

AGARWAL, R. *Object Detection: An End to End Theoretical Perspective*. 2019. [Online; accessed 23-August-2019]. Disponível em: <https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a>.

ALBANESE, A.; NARDELLO, M.; BRUNELLI, D. Automated pest detection with dnn on the edge for precision agriculture. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, v. 11, n. 3, p. 458–467, 2021.

BARBEDO, J. G. A. Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, v. 172, p. 84–91, 2018. ISSN 1537-5110. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511018303027>.

BARBEDO, J. G. A. Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, v. 180, p. 96–107, 2019. ISSN 1537-5110. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511018307797>.

BAUMüLLER, H. The little we know: An exploratory literature review on the utility of mobile phone-enabled services for smallholder farmers. *Journal of International Development*, v. 30, p. 134–154, 01 2018.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. New York, NY, USA: Association for Computing Machinery, 1992. (COLT '92), p. 144–152. ISBN 089791497X. Disponível em: <https://doi.org/10.1145/130385.130401>.

CUI, N. Applying gradient descent in convolutional neural networks. *Journal of Physics: Conference Series*, IOP Publishing, v. 1004, p. 012027, apr 2018. Disponível em: <https://doi.org/10.1088/1742-6596/1004/1/012027>.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009.

EMBRAPA. *Código Florestal: Módulos Fiscais*. [S.l.], 2012 (accessed December 1st, 2021). <https://www.embrapa.br/codigo-florestal/area-de-reserva-legal-arl/modulo-fiscal>.

EMBRAPA. *Digipathos Dataset*. [S.l.], 2014 (accessed August 31, 2021). <https://www.digipathos-rep.cnptia.embrapa.br/>.

ERHAN, D.; BENGIO, Y.; COURVILLE, A.; MANZAGOL, P.-A.; VINCENT, P.; BENGIO, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, v. 11, n. 19, p. 625–660, 2010. Disponível em: <http://jmlr.org/papers/v11/erhan10a.html>.

FIX, E.; HODGES, J. L. Discriminatory analysis - nonparametric discrimination: Consistency properties. *International Statistical Review*, v. 57, p. 238, 1989.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine Learning*, v. 29, n. 2, p. 131–163, Nov 1997. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1007465528199>.

FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, v. 1, n. 2, p. 119 – 130, 1988. ISSN 0893-6080. Disponível em: <http://www.sciencedirect.com/science/article/pii/0893608088900147>.

GASSON, R.; CROW, G.; ERRINGTON, A.; HUTSON, J.; MARSDEN, T.; WINTER, D. The farm as a family business: A review. *Journal of Agricultural Economics*, v. 39, p. 1 – 41, 11 2008.

HASAN, R. I.; YUSUF, S. M.; ALZUBAIDI, L. Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants*, v. 9, n. 10, 2020. ISSN 2223-7747. Disponível em: <https://www.mdpi.com/2223-7747/9/10/1302>.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun 2016. Disponível em: <http://dx.doi.org/10.1109/cvpr.2016.90>.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Identity mappings in deep residual networks. In: LEIBE, B.; MATAS, J.; SEBE, N.; WELLING, M. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 630–645. ISBN 978-3-319-46493-0.

HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708.

HUGHES, D.; SALATHE, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. 11 2015.

IBGE. *Agricultural Census 2017*. [S.l.], 2017 (accessed March 1, 2021). <https://censoagro2017.ibge.gov.br/templates/censo_agro/resultadosagro/index.html>.

JI, M.; ZHANG, K.; WU, Q.; DENG, Z. Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. *Soft Comput.*, v. 24, p. 15327–15340, 2020.

KOIDL, K. Loss functions in classification tasks. *School of Computer Science and Statistic Trinity College, Dublin*, 2013.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, p. 1097–1105, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, May 2015. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/nature14539>.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, p. 2278 – 2324, 12 1998.

LI, M.; ZHANG, T.; CHEN, Y.; SMOLA, A. J. Efficient mini-batch training for stochastic optimization. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2014. (KDD '14), p. 661–670. ISBN 9781450329569. Disponível em: <https://doi.org/10.1145/2623330.2623612>.

LI, Q.; PENG, X.; QIAO, Y.; PENG, Q. Learning label correlations for multi-label image recognition with graph networks. *Pattern Recognition Letters*, v. 138, p. 378–384, 2020. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167865520302968>.

LIAO, W.; CHEN, X.; LU, X.; HUANG, Y.; TIAN, Y. Deep transfer learning and time-frequency characteristics-based identification method for structural seismic response. *Frontiers in Built Environment*, v. 7, p. 10, 2021. ISSN 2297-3362. Disponível em: <https://www.frontiersin.org/article/10.3389/fbuil.2021.627058>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <https://doi.org/10.1007/BF02478259>.

MOHANTY, D. P. H. S. P.; SALATHé, M. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, v. 7, p. 1419, 2016. ISSN 1664-462X. Disponível em: <https://www.frontiersin.org/article/10.3389/fpls.2016.01419>.

MOHANTY, S. P. *PlantVillage Dataset*. [S.l.], 2016 (accessed March 1, 2021). <https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw>.

NARKHEDE, M. V.; BARTAKKE, P. P.; SUTAONE, M. S. A review on weight initialization strategies for neural networks. *Artificial Intelligence Review*, Jun 2021. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-021-10033-z>.

OERKE, E.-C.; DEHNE, H.-W. Safeguarding production—losses in major crops and the role of crop protection. *Crop Protection*, v. 23, n. 4, p. 275–285, 2004. ISSN 0261-2194. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0261219403002540>.

OLIVEIRA, C.; AUAD, A.; MENDES, S.; FRIZZAS, M. Crop losses and the economic impact of insect pests on brazilian agriculture. *Crop Protection*, v. 56, p. 50–54, 2014. ISSN 0261-2194. Disponível em: <https://www.sciencedirect.com/science/article/pii/S026121941300269X>.

PARDEDE, H. F.; SURYAWATI, E.; KRISNANDI, D.; YUWANA, R. S.; ZILVAN, V. Machine learning based plant diseases detection: A review. In: *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*. [S.l.: s.n.], 2020. p. 212–217.

PATHAK, A. R.; PANDEY, M.; RAUTARAY, S. Application of deep learning for object detection. *Procedia Computer Science*, v. 132, p. 1706 – 1717, 2018. ISSN 1877-0509. International Conference on Computational Intelligence and Data Science. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050918308767>.

PATRO, S. G.; SAHU, D.-K. K. Normalization: A preprocessing stage. *IARJSET*, 03 2015.

PEREIRA, C.; MORAIS, R.; REIS, M. Deep learning techniques for grape plant species identification in natural images. *Sensors*, v. 19, p. 4850, 11 2019.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, v. 29, n. 9, p. 2352–2449, 2017.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, American Psychological Association, US, v. 65, n. 6, p. 386–408, 1958. Disponível em: <https://doi.org/10.1037/h0042519>.

RUBY, U.; YENDAPALLI, V. Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, v. 9, 10 2020.

SAMPAIO, Y. d. S. B.; VITAL, T. W. Agricultura familiar em pernambuco: O que diz o censo agropecuário de 2017. *Revista Econômica do Nordeste*, v. 51, p. 155–171, 2020.

SARANGI, S.; UMADIKAR, J.; KAR, S. Automation of agriculture support systems using wisekar: Case study of a crop-disease advisory service. *Computers and Electronics in Agriculture*, v. 122, p. 200–210, 2016. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169916000144>.

SAZLı, M. H. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, Ankara University, Ankara University Faculty of Sciences Besevler Ankara 06100 Turkey, v. 50, p. 0 – 0, 2006. ISSN 1303-6009.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, v. 61, p. 85–117, 2015. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1409.1556>.

SMITH, P. Bilinear interpolation of digital images. *Ultramicroscopy*, v. 6, n. 1, p. 201–204, 1981. ISSN 0304-3991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0304399181801994>.

SONG, L.; LIU, J.; QIAN, B.; SUN, M.; YANG, K.; SUN, M.; ABBAS, S. A deep multi-modal cnn for multi-instance multi-label image classification. *IEEE Transactions on Image Processing*, v. 27, n. 12, p. 6025–6038, 2018.

SOOFI, A.; AWAN, A. Classification techniques in machine learning: Applications and issues. *Journal of Basic & Applied Sciences*, v. 13, p. 459–465, 08 2017.

SVOZIL, D.; KVASNICKA, V.; POSPICHAL, J. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, v. 39, n. 1, p. 43–62, 1997. ISSN 0169-7439. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169743997000610>.

SZE, V.; CHEN, Y.; YANG, T.; EMER, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, v. 105, n. 12, p. 2295–2329, Dec 2017.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826.

TAWIAH, C. A.; SHENG, V. S. A study on multi-label classification. In: PERNER, P. (Ed.). *Advances in Data Mining. Applications and Theoretical Aspects*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 137–150. ISBN 978-3-642-39736-3.

THEODORIDIS, S. *Machine Learning: A Bayesian and Optimization Perspective*. 1st. ed. USA: Academic Press, Inc., 2015. ISBN 0128015225.

VOULODIMOS, A.; DOULAMIS, N.; DOULAMIS, A.; PROTOPAPADAKIS, E. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, v. 2018, p. 1–13, 02 2018.

WANG, J.; YANG, Y.; MAO, J.; HUANG, Z.; HUANG, C.; XU, W. Cnn-rnn: A unified framework for multi-label image classification. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 2285–2294.

XIE, X.; MA, Y.; LIU, B.; HE, J.; LI, S.; WANG, H. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Frontiers in Plant Science*, v. 11, p. 751, 2020. ISSN 1664-462X. Disponível em: <https://www.frontiersin.org/article/10.3389/fpls.2020.00751>.

YAMASHITA, R.; NISHIO, M.; DO, R.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, v. 9, 06 2018.

YAN, Z.; LIU, W.; WEN, S.; YANG, Y. Multi-label image classification by feature attention network. *IEEE Access*, v. 7, p. 98005–98013, 2019.

ZHANG, M.-L.; ZHOU, Z.-H. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, v. 26, n. 8, p. 1819–1837, 2014.

ZHANG, Y.-D.; DONG, Z.; CHEN, X.; JIA, W.; DU, S.; MUHAMMAD, K.; WANG, S.-H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, v. 78, n. 3, p. 3613–3632, Feb 2019. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-017-5243-3>.

ZOPH, B.; VASUDEVAN, V.; SHLENS, J.; LE, Q. V. Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 8697–8710.