



# LEAF DISEASE IDENTIFICATION

---

A Vision and Perception project

Giulia Neccia 1847033

Prof.ssa Irene Amerini

Prof. Paolo Russo



# Proposed solution

---

Web app to help farmers identify easily and rapidly the disease that affects the crop



## **Fast intervention**

Timely and correct identification of a disease is critical



## **Availability**

Always available with no added cost



## **Ease of use**

No need for specialized personnel





# Dataset used: PlantVillage

---



## 70295 images

Images of both healthy and unhealthy leaves from different crops

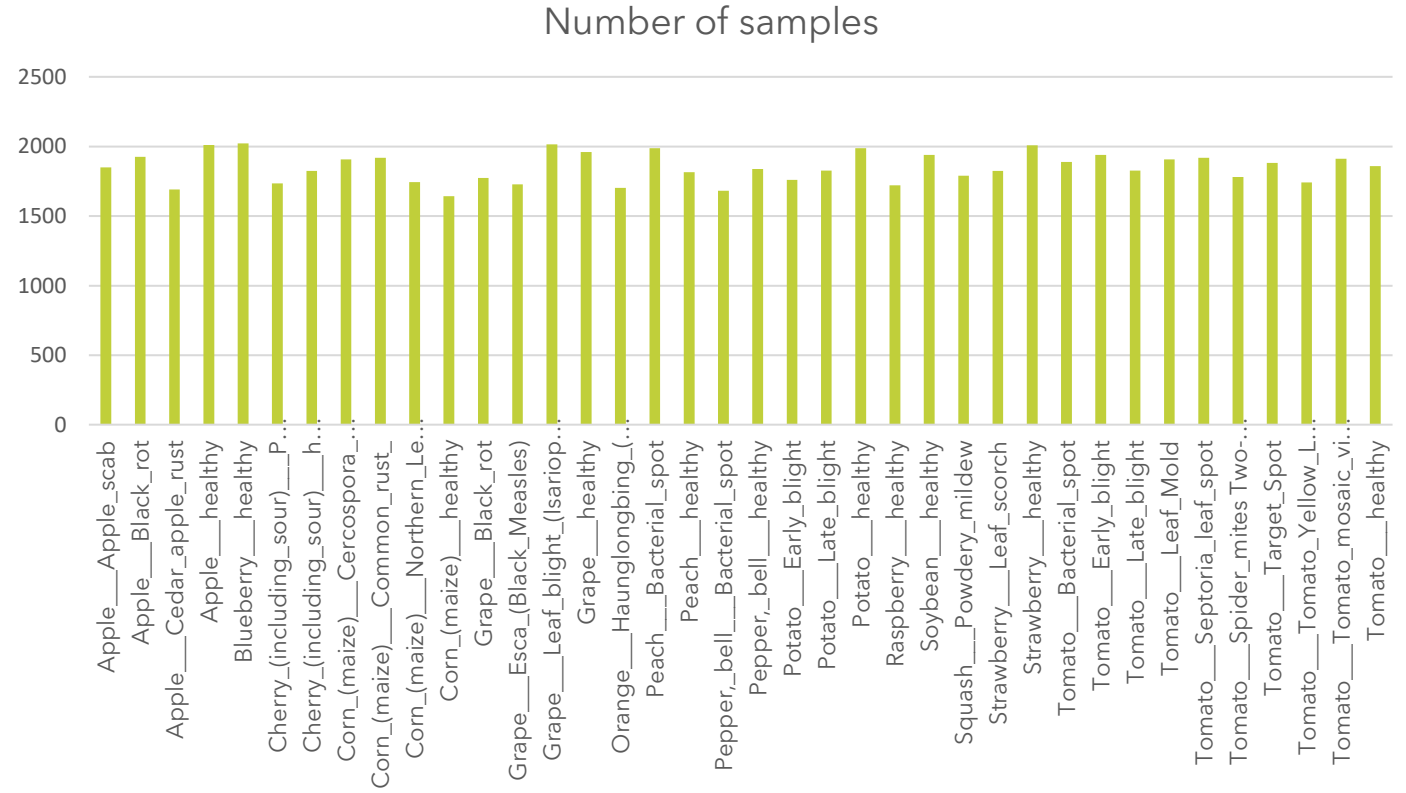
## 38 classes

The classes are specified as crop-type of disease

## 14 types of crop

['Tomato', 'Grape', 'Orange', 'Soybean', 'Squash', 'Potato', 'Corn\_(maize)', 'Strawberry', 'Peach', 'Apple', 'Blueberry', 'Cherry\_(including\_sour)', 'Pepper,\_bell', 'Raspberry']

# Dataset used



# Dataset augmentation

For this task the original images are resized to a fixed size of **224x224** pixels. Then, two new images are created from each original image by horizontally and vertically flipping it. Next, both the original and flipped images are rotated 45 degrees clockwise.





# Transfer learning

---

Transfer learning is a powerful technique in deep learning that involves using pre-trained models and adapting them to new tasks.

## MobileNetV2

**MobileNetV2** is a convolutional neural network architecture that seeks to perform well on mobile devices.

01

## CNN trained from scratch

It contains 3 convolutional layers, using dropout and batch normalization to avoid overfitting.

04

## VGG16

VGG16, is a 16-layer deep neural network, it's a relatively extensive network with a total of 138 million parameters. However, the simplicity of the VGGNet16 architecture is its main attraction.

03

## 02 InceptionV3

Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7 convolutions ...





# MobileNetV2 01

**MobileNetV2** is a convolutional neural network architecture that seeks to perform well on mobile devices.

MobileNet is a popular pre-trained model for image classification tasks that is optimized for mobile and embedded devices. In this case was used a pretrained mobileNet v2 with ImageNet

Accuracy obtained with: optimizer=Adam, num. Epoch=10

Learning rate	Accuracy	Training time
0.01	10.03%	~350 s/epoch
0.0001	81.04%	~400 s/epoch
0.000001	<b>87.09%</b>	~300 s/epoch

} overfitting



# MobileNetV2 01

**MobileNetV2** is a convolutional neural network architecture that seeks to perform well on mobile devices.

MobileNet is a popular pre-trained model for image classification tasks that is **optimized for mobile** and embedded devices. In this case was used a pretrained mobileNet v2 with ImageNet

Accuracy obtained with: optimizer=SGD, num. Epoch=10

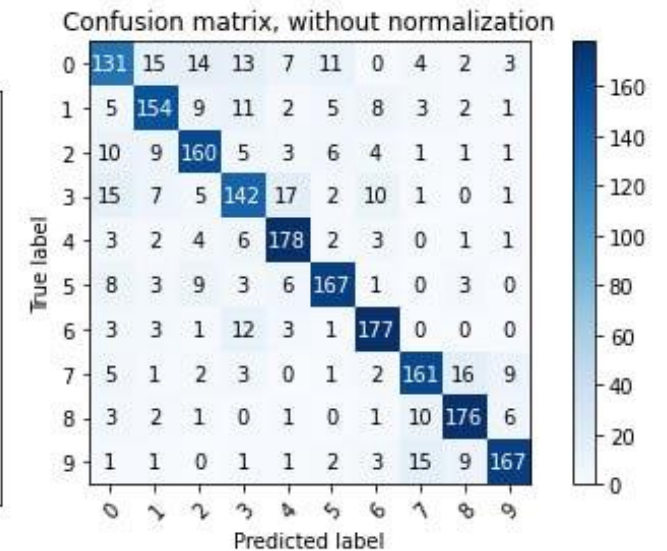
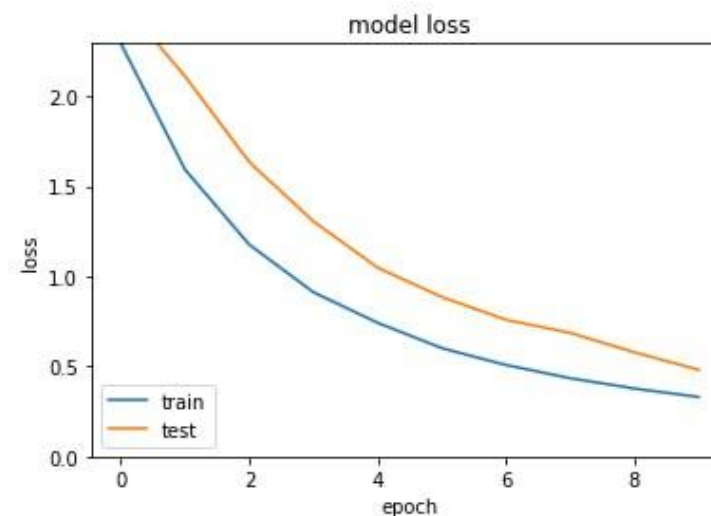
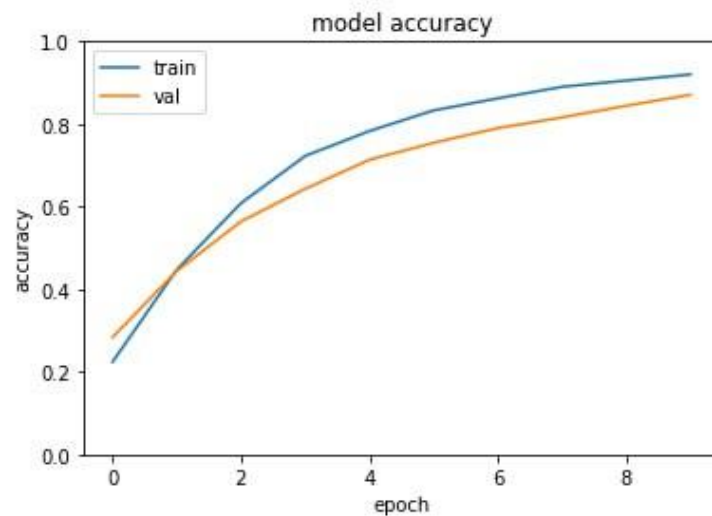
Learning rate	Accuracy	Training time
0.01	27.84%	~350 s/epoch
0.0001	<b>77.40%</b>	~400 s/epoch
0.00001	40.82%	~280 s/epoch
0.000001	27.84%	~300 s/epoch



# MobileNetV2 01

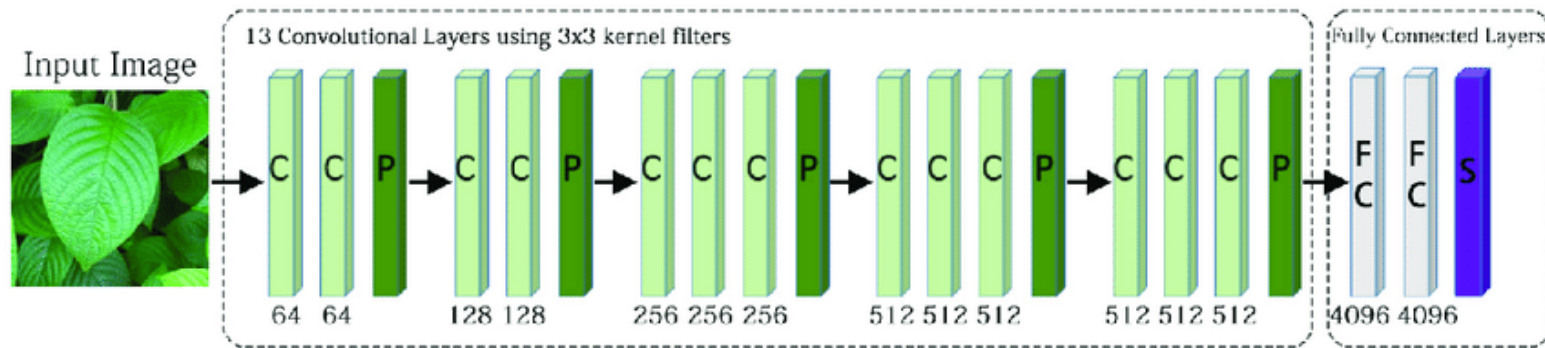
**MobileNetV2** is a convolutional neural network architecture that seeks to perform well on mobile devices.

MobileNet is a popular pre-trained model for image classification tasks that is **optimized for mobile** and embedded devices. In this case was used a pretrained mobileNet v2 with ImageNet



# VGG16 03

VGG16, is a **16-layer deep neural network**, it's a relatively extensive network with a total of **138 million parameters**. However, the simplicity of the VGGNet16 architecture is its main attraction.



## Advantages

simplicity and strong performance on image classification tasks,

Learning rate	Accuracy	Training time
0.0001	99.04%	~350 s/epoch

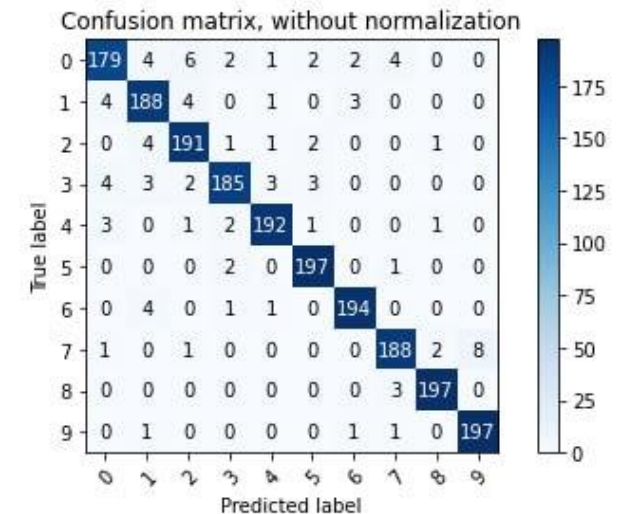
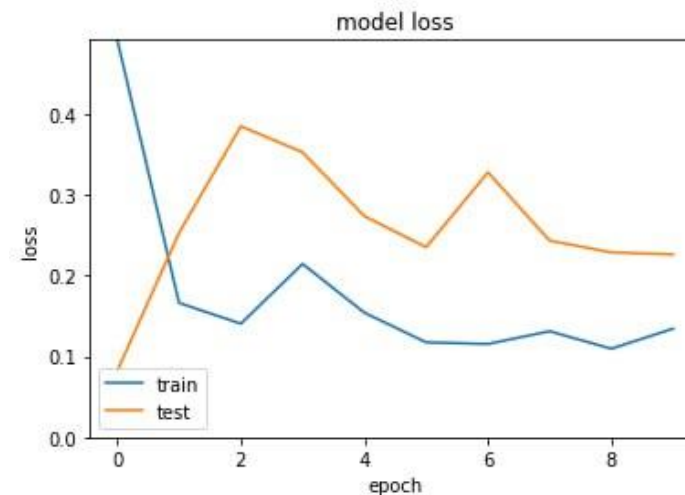
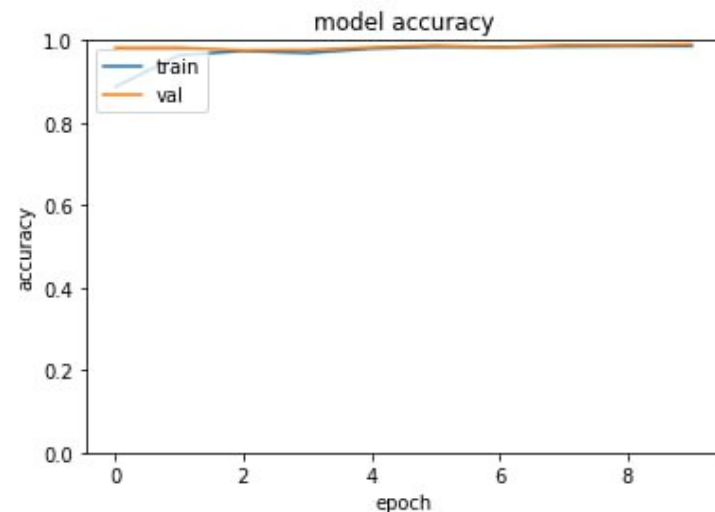
## Disdvantages

533 MB  
time consuming

# VGG16 03

VGG16, is a 16-layer deep neural network, it's a relatively extensive network with a total of 138 million parameters. However, the simplicity of the VGGNet16 architecture is its main attraction.

Learning rate	Accuracy	Training time
0.0001	99.04%	~350 s/epoch



# InceptionV3 02

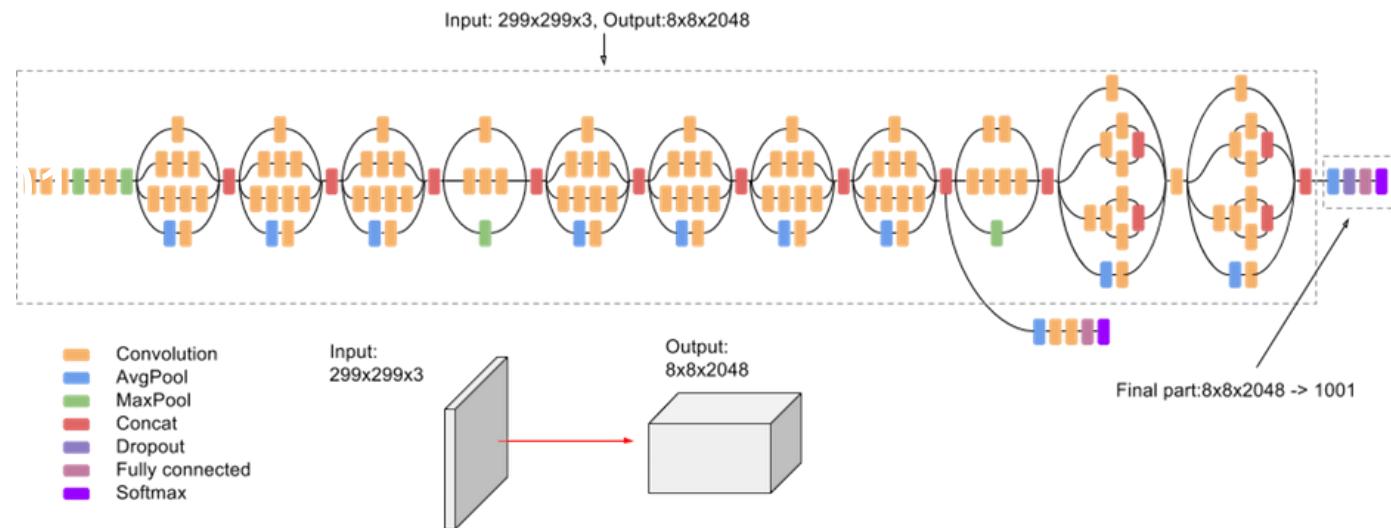
The inception v3 model was released in the year 2015, it has a total of **42 layers** and a **lower error rate** than its predecessors.

## Advantages

Efficient utilisation of computing resource with minimal increase in computation load

## Disdvantages

91 MB  
time consuming

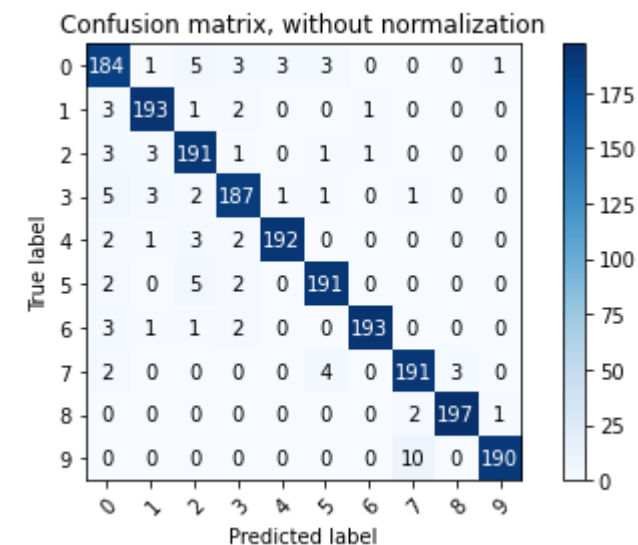
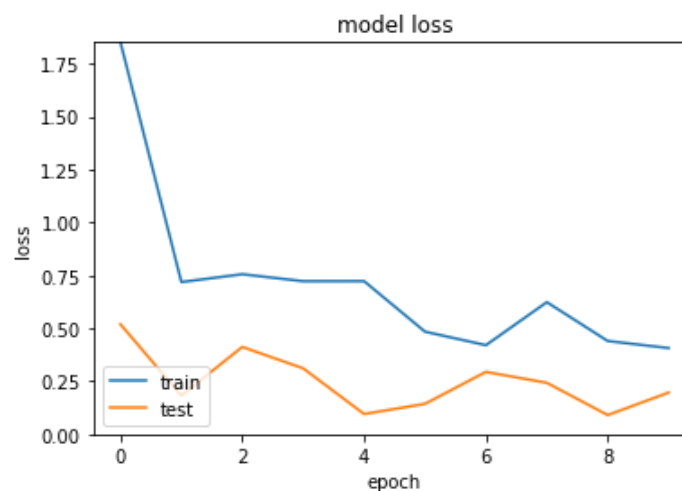
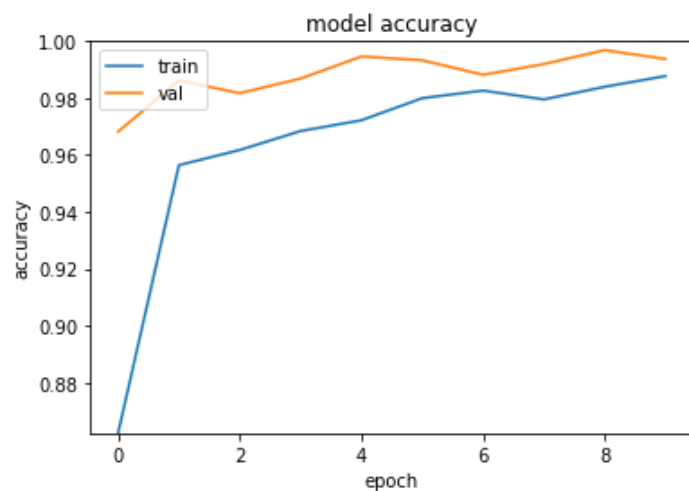


Learning rate	Accuracy	Training time
0.0001	99.38%	~350 s/epoch

# InceptionV3 02

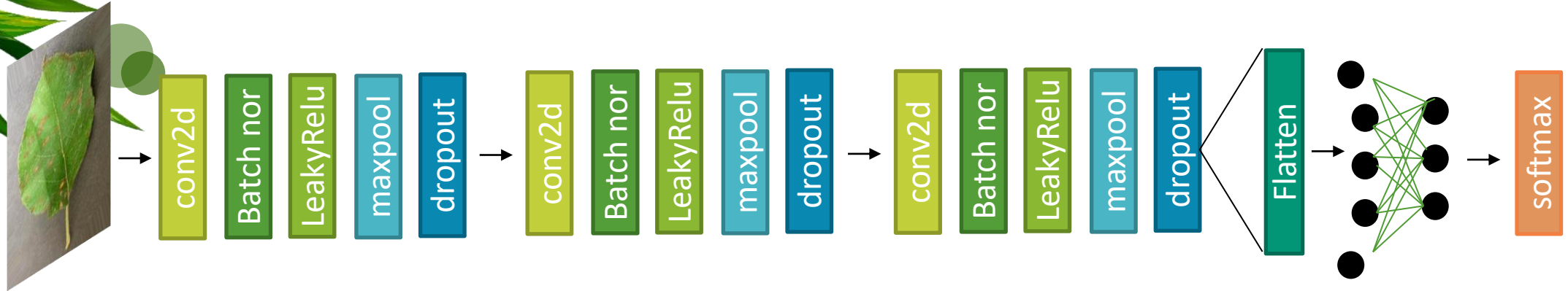
The inception v3 model was released in the year 2015, it has a total of **42 layers** and a **lower error rate** than its predecessors.

Learning rate	Accuracy	Training time
0.0001	99.38%	~350 s/epoch





# CNN built from scratch



Input shape of (224, 224, 3) representing RGB images with a size of **224x224** pixels.

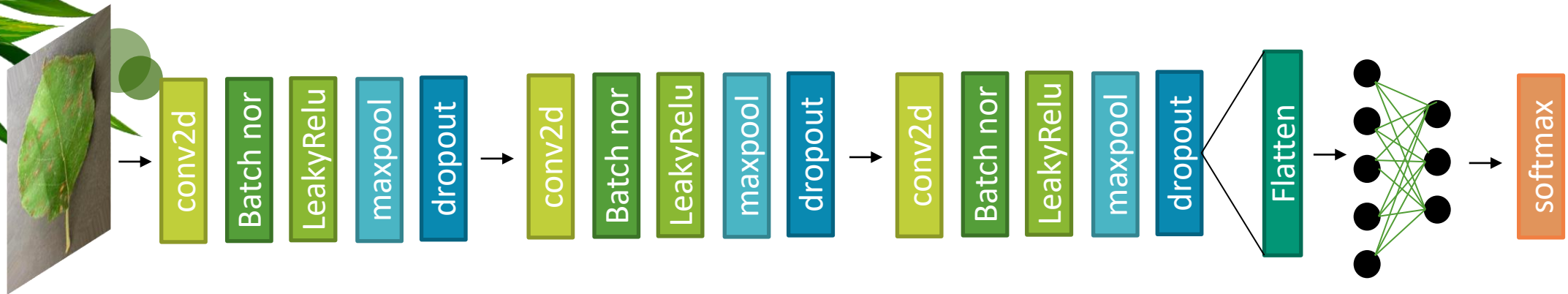
The architecture of the CNN consists of three **pairs of Convolutional, Batch Normalization, LeakyReLU, and MaxPooling** layers, followed by a **Dropout** layer for regularization.

After the third MaxPooling layer, the output is flattened into a 1D array, and passed through a fully connected Dense layer with 512 neurons, followed by Batch Normalization and LeakyReLU activation for non-linearity. Finally, the output layer has 10 neurons, representing the 10 classes, with a softmax activation function to output class probabilities.

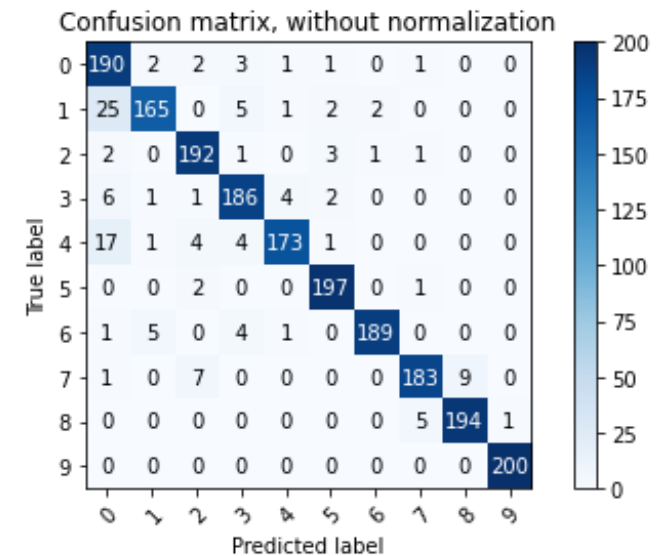
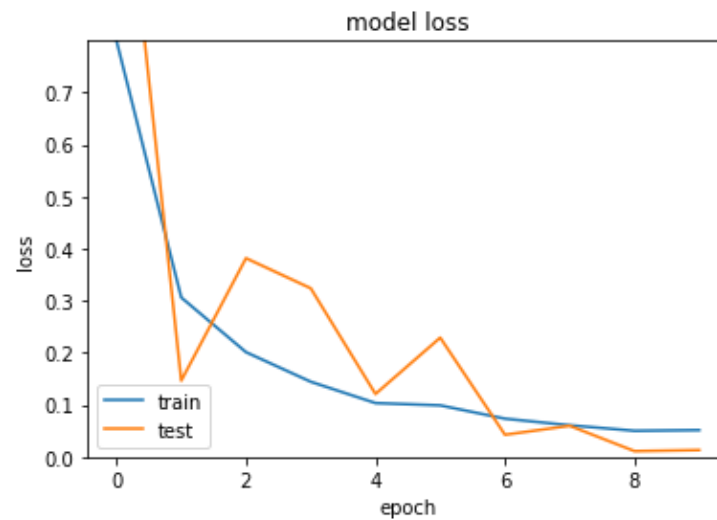
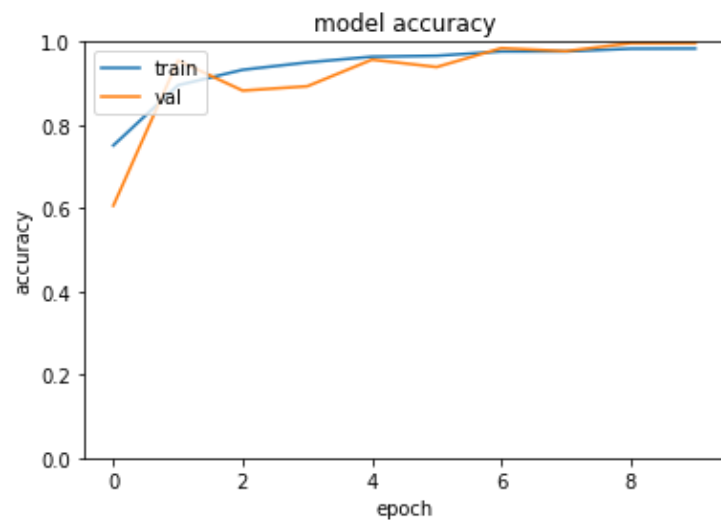
## Main features

- simple design
- no additional complex features (i.e. residual connections or inception modules)
- It perform reasonably well on image classification tasks with small to medium-sized datasets

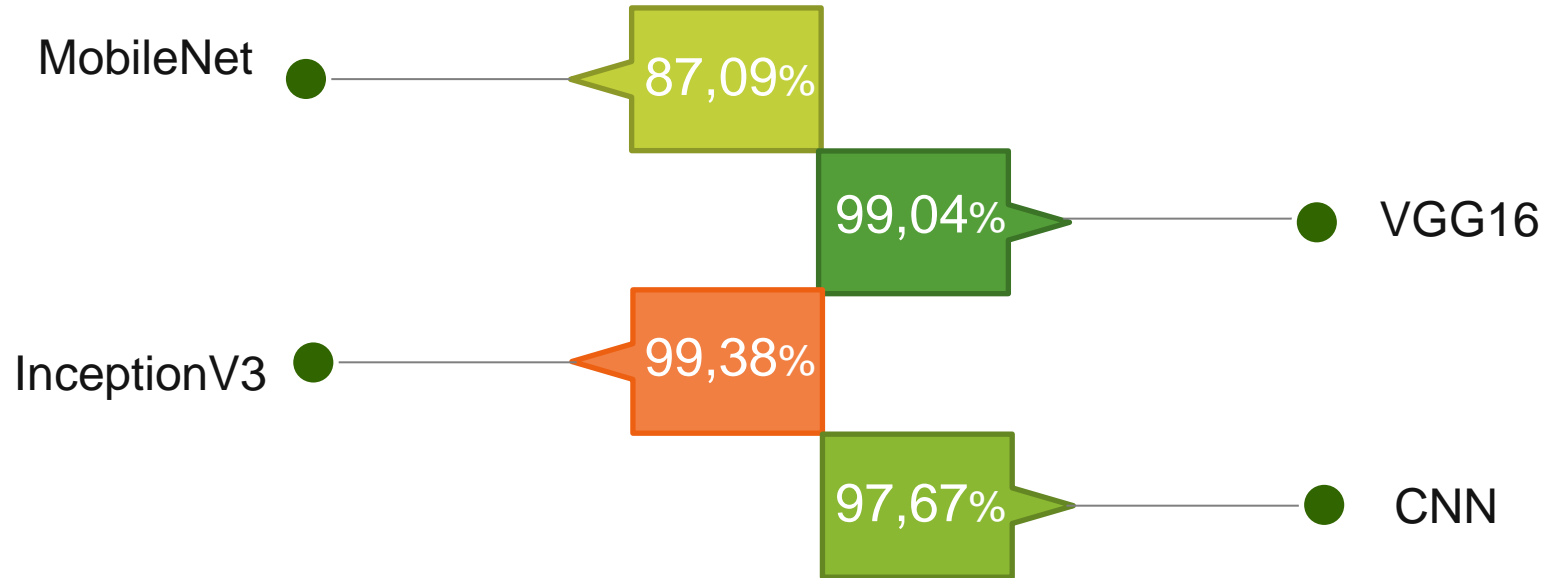




Accuracy obtained with: optimizer=**RMSprop**, num. Epoch=10, learning rate=0.001



# Final comparison: Accuracy



Even though MobileNet has the lowest accuracy it's the best suited for a **mobile app** which is the future work in mind for this project.

# WebApp: Leaf disease identification

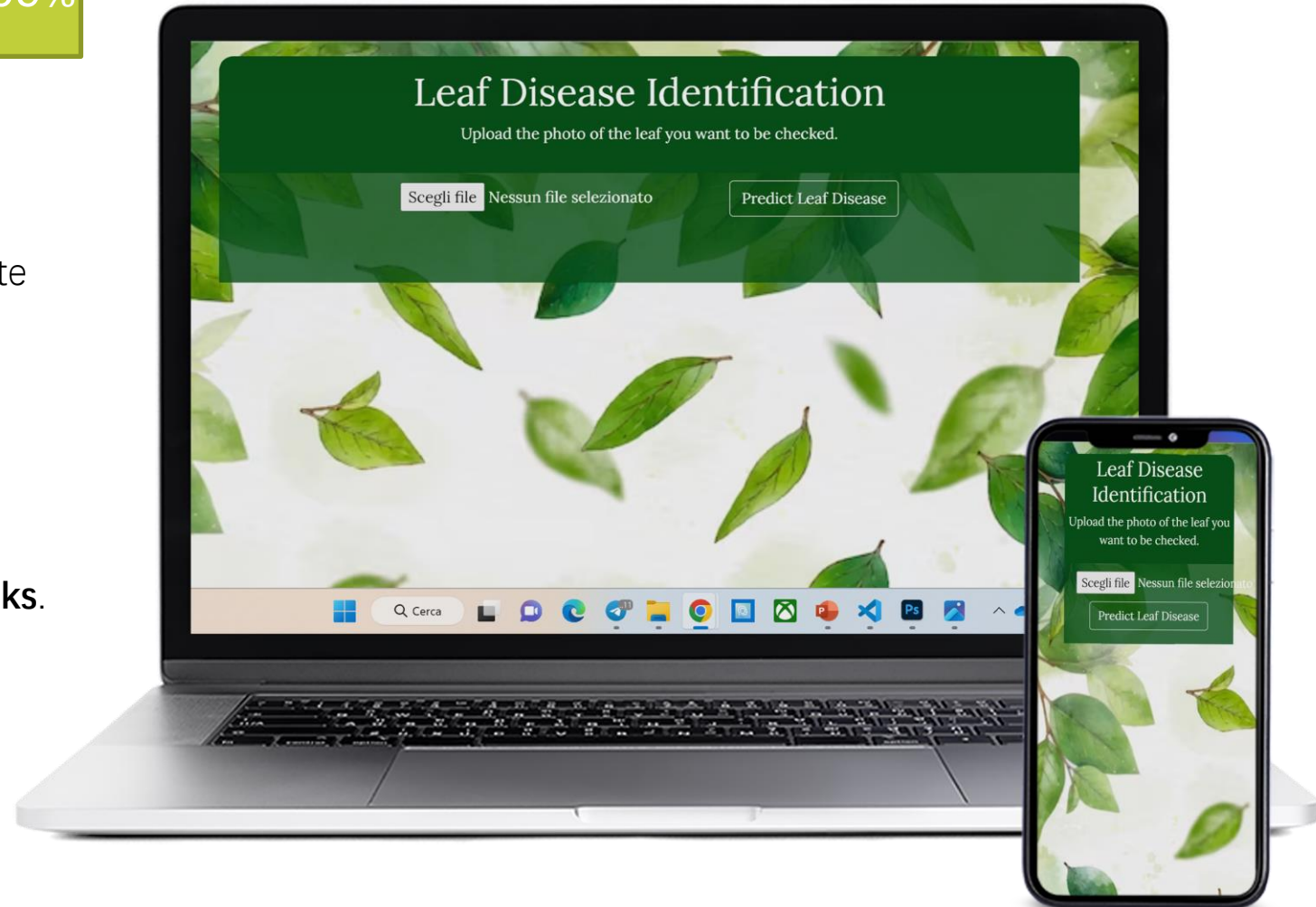
MobileNet

87,09%

**Used Tools:** Flask, pickle

1. Save model to disk as pkl
2. Obtain data from the form on the website using the method POST
3. Call the predict function on such data
4. Return the results on screen

Flask is a lightweight WSGI web application framework. It has become one of the most popular **Python web application frameworks**.





# Future works

---

**Real-time classification:** We can try deploying this model on a mobile device or a Raspberry Pi and performing real-time classification of leaf diseases.



**Explore other modalities:** such as hyperspectral or thermal imaging to detect leaf diseases. These modalities can provide more detailed information about the leaf's health and can improve the accuracy of the classification.



**Data collection:** We can try collecting more data from different sources and locations to increase the diversity of your dataset. This can also help to reduce bias and improve the generalization of the model.



**Android app and improved User Interface:** we can work on improving the user interface of the app, making it more user-friendly and easy to navigate.



THANKS

---