Image-Based Plant Disease Detection: A Comparison of Deep Learning and Classical Machine Learning Algorithms

Draško Radovanović [1], Slobodan Đukanović [1, 2], *Member, IEEE*

***Abstract* --- Rapid human population growth requires corresponding increase in food production. Easily spreadable diseases can have a strong negative impact on plant yields and even destroy whole crops. That is why early disease diagnosis and prevention are of very high importance. Traditional methods rely on lab analysis and human expertise which are usually expensive and unavailable in a large part of the undeveloped world. Since smartphones are becoming increasingly present even in the most rural areas, in recent years scientists have turned to automated image analysis as a way of identifying crop diseases. This paper presents the most recent results in this field, and a comparison of deep learning approach with the classical machine learning algorithms.**

***Keywords* — machine learning, crop diseases, deep learning.**

I. Introduction

HUMAN population steadily continues to grow, and along with it the need for food production increases. According to the UN projections [1], human population is expected to reach 9.7 billion in 2050, 2 billion more than today. Considering that most of the population growth is to occur in the least developed countries (around 80% increase in the next 30 years), where the food scarcity is the main problem, it is easy to conclude that minimizing food loss in those countries is a primary concern. It is estimated that the yield loss worldwide is between 20 and 40 percent [2], with many farms suffering a total loss.

Traditional methods for detecting diseases require manual inspection of plants by experts. This process needs to be continuous, and can be very expensive in large farms, or even completely unavailable to many small farm holders living in rural areas. This is why many attempts to automate disease detection have been made in the last few decades. One of the notable approaches is the use of hyperspectral imaging. Hyperspectral images are usually taken by satellites or airborne imaging devices and used for monitoring large areas. A downside of this approach is extremely high equipment cost, as well as high dimensionality and small number of samples which make them unsuitable for machine learning (ML) analysis.

Because of the recent breakthroughs in computer vision and the availability of cheap hardware, currently the most popular approach is the analysis of RGB images. The other motive for analysing RGB images is that with the current smartphone ubiquitousness these solutions have potential to reach even the most rural areas. RGB images can be analysed by classical ML algorithms or the deep learning (DL) approach.

Classical methods rely on image pre-processing and the extraction of features which are then fed into one of the ML algorithms. Popular algorithm choices are Support Vector Machines (SVM), k-Nearest Neighbours (*k*-NN), Fully Connected Neural Networks (FCNN), Decision Trees, Random Forests etc. In the last few years, the researchers shifted almost exclusively to the DL methods for image classification tasks. The reason is that they almost always outperform classical algorithms when given reasonably sized dataset, and can be implemented without the need for hand-engineered features. In this paper, we compare the DL approach with classical ML algorithms for the study case of plant disease classification.

II. Dataset

In this paper, the PlantVillage Dataset is used [3]. It consists of images of plant leaves taken in a controlled environment. In total, there are 54 306 images of 14 different plant species, distributed in 38 distinct classes given as species/disease pair. Species present in this dataset are: Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. There are 17 fungal diseases, 4 bacterial diseases, 2 viral diseases, 2 mould diseases, and one mite disease present in this dataset, as well as images of healthy plants from 12 different species.

Images were taken with a standard digital camera, outside, under different weather conditions, and were collected from multiple sources, making the dataset more diverse. Large number of samples and different diseases make this dataset suitable for applying ML algorithms, especially DL ones. The downside of this dataset is that individual leaves were cropped and put against a uniform background to be photographed, making conditions substantially different than those in the field. The distribution of images is not uniform, and the number of samples per class varies from 150 to 5500, which can be seen in Figure 1. Also, reports about significant number of mislabelled samples were made in [4]. The dataset offers colour images, grayscale images, and segmented

[1] Faculty of Electrical Engineering, University of Montenegro, Džordža Vašingtona bb, 81000 Podgorica, Montenegro. e-mail: {draskor, slobdj}@ac.me.

[2] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic djukaslo@fel.cvut.cz.

978-1-7281-5136-6/20/$31.00 ©2020 IEEE

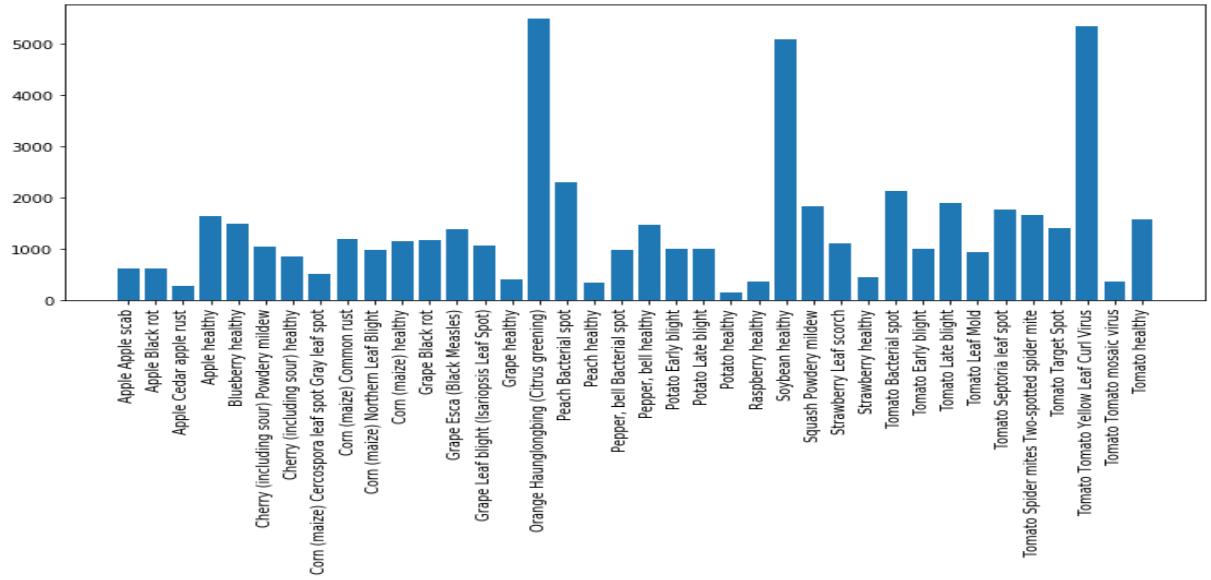

Figure 1: Number of samples per class

images where the background is masked. In this paper, segmented images are used.

III. Classical ML approach

When using classical algorithms, certain pre-processing steps need to be taken. The basics are outlined in Figure 2.

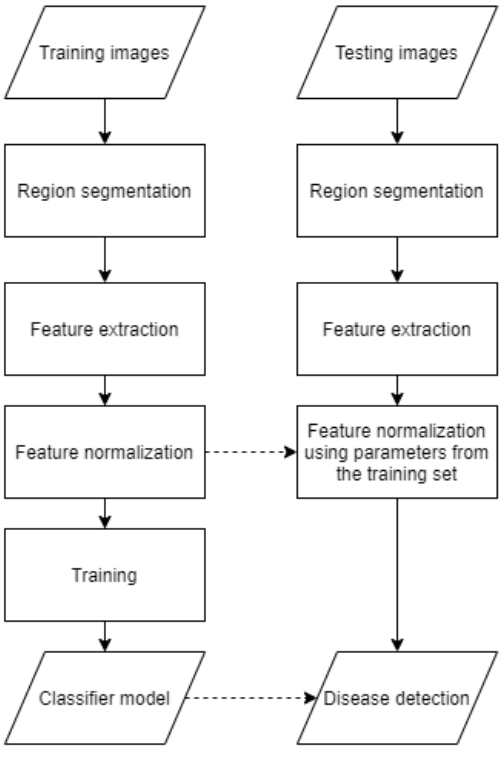

Figure 2: Flowchart of the basic training and testing steps

A. Region segmentation

When working on image classification, the usual pre-processing steps include scaling images to the same dimensions, removal of the background and artifacts. Since the PlantVillage dataset includes already segmented and scaled images, these steps were not needed in our case. We preprocessed these images by futher segmenting them in order to extract potentionally infected leaf areas, which has been done by removing all pixels whose green channel value exceeded those of red and blue channels. Examples of segmented images and images with green pixels removed can be seen in Figure 3.

B. Feature extraction

Feature selection is arguably the hardest and the most important part in the implementation of ML algorithms. Selecting appropriate features requires detailed analysis and expertise in the domain of interest. In this work, we have used texture features (on both full images and images with removed green pixels) obtained by analysing the grey level co-occurrence matrix (GLCM) [5] and general colour statistical features obtained by histogram analysis of the full image. The GLCM is used to describe a spatial relationship of neighbouring pixels, i.e. it describes a probability of two pixel values, i and j, being on a distance d and at angle θ from one another. It is defined as a matrix of dimension $N \times N$ (N is the number of distinct pixel values), where value $G(i,j)$ represents a number of times when pixel of value j occurred at a distance d and at angle θ from a pixel of value i. From the GLCM, texture features like correlation, contrast, energy, homogeneity and dissimilarity can be obtained.

Colour features are obtained by extracting statistical features from image histograms. They are used to provide a general description of colour statistics in the image.

In this paper, we have used 120 features obtained by texture analysis and 96 features obtained by colour analysis, 216 in total. We have calculated 12 GLCMs for both a full image and image with removed green pixels. We have used 4 distances (1, 3, 10 and 20 pixels), and 3 angles (0, $\pi/4$, $\pi/2$). For each GLCM we have calculated 5 features (correlation, contrast, energy, homogeneity and dissimilarity). Colour features were calculated only for full images. We used 6 features per colour channel (mean, standard deviation, kurtosis, skew, entropy and RMS), 18 features in total. We also calculated a histogram with 26

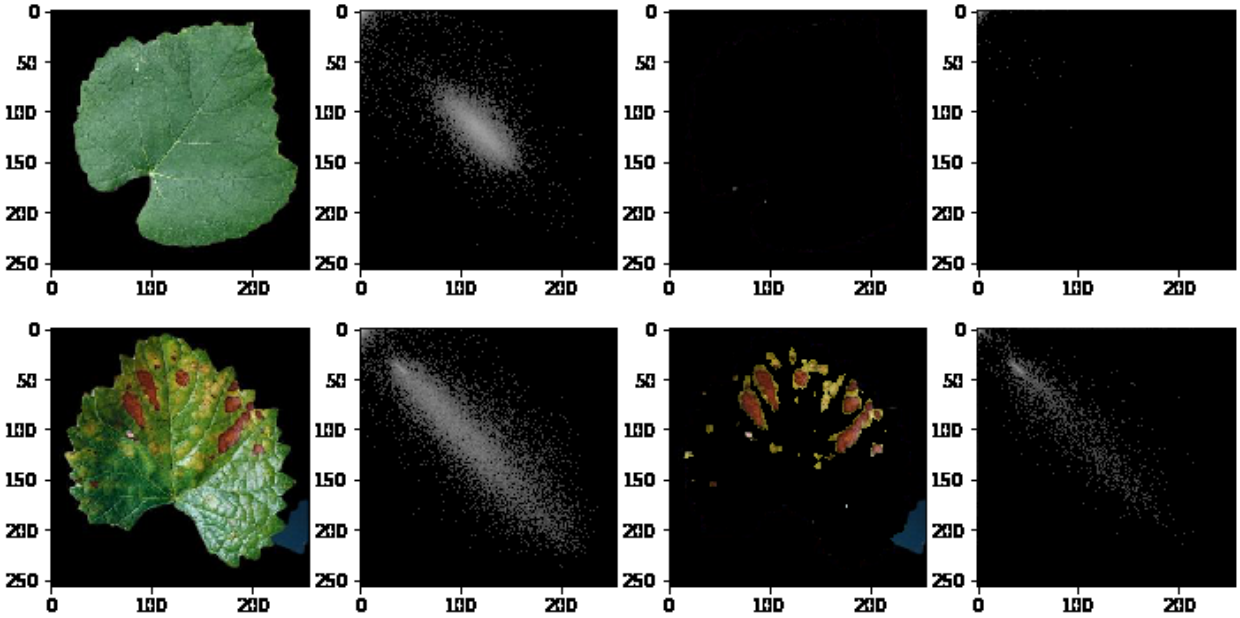

Figure 3: Example of two leaf images (top: healthy, bottom: diseased). From left to right: Full image, GLCM calculated on a full image, image with removed green pixels, GLCM calculated on image with removed green pixels.

buckets per channel and used pixel count per bucket as features, which multiplied with 3 channels gave us 78 features.

C. *Support Vector Machines*

SVM is a supervised learning algorithm used for classification or regression problems. Classification is done by defining a separating hyperplane in the feature space. In the original form, it performs linear classification on two classes. By using kernels, it can also perform non-linear classification. Kernels are used for an efficient transformation of the original feature space into high dimensional or infinite dimensional feature space, allowing for highly non-linear hyperplanes. SVM can fit highly complex datasets and at the same time exhibit good generalization properties [6].

Multiclass classification using SVM can be implemented using one-vs-all or one-vs-one strategies. One-vs-all approach trains N classifiers (N is the number of classes), where each classifier considers examples from one class as positive, and all others as negative. One-vs-one approach trains $N(N-1)/2$ binary classifiers and determines the winner by max-wins voting [10].

We have experimented with different configurations and found that the best results were obtained by using the radial basis function kernel and regularization parameter C=100. One-vs-all approach was used. The achieved accuracy on the test set is 91.74%,

D. *k-Nearest Neighbours*

k-NN [7] is a very simple algorithm often used for classification problems. It is both non-parametric (doesn't have a fixed number of parameters) and lazy learning (doesn't have a training phase). k-NN works under the assumption that most samples from the same class are close to each other in the feature space. When determining the class of the sample, k-NN will look at its k closest neighbours and decide to which class it belongs by the simple majority rule. Small values of k will allow for higher non-linearity but will be sensitive to outliers. High values of k achieve good generalization but fail to fit complex boundaries. The best value for parameter k is determined experimentally.

For this dataset, small values of k were shown to give the best results. Varying k from 1 to 9 doesn't change the accuracy much, with best result being 78.06% much lower than the SVM. We used k=5 in this work.

E. *Fully Connected Neural Network*

FCNN is the simplest type of artificial neural networks. It is a supervised learning algorithm able to model highly non-linear functions. As opposed to SVM and k-NN, it does not converge to the global optimum, but when properly configured, it usually gives good enough results. Important neural network configuration parameters are:

- number of hidden layers
- activation function
- number of neurons per layer
- regularization method
- optimization method

In this paper, we used an FCNN with four hidden layers with 300, 200, 100 and 50 neurons per layer, respectively. Activation function in hidden layers is a rectified linear unit (ReLU), with a softmax in the output layer [8]. We used L2 regularization with regularization parameter equal to 0.3. Adam optimizer with default parameters was used. This configuration gave us the accuracy of 91.46% on the test set.

IV. DEEP LEARNING

DL represents a class of ML algorithms that uses multiple layers in order to learn features in a more hierarchical manner. There are various types of DL algorithms, but they are mostly based on artificial neural networks. The state-of-the-art solutions in artificial

intelligence (AI) today are mostly based on this class of algorithms. DL models have shown to be able to learn extremely complex patterns given enough data. Besides the fact that they can fit extremely complex models, a big advantage of DL algorithms is feature learning. With DL, there is no need for feature engineering, since DL models are fed raw data and they are responsible for learning appropriate features. For image recognition problems, convolutional networks (CNN) are mostly used.

For comparison with classical models, in this paper we have used a GoogLeNet [9] model with parameter configuration presented in [3]. We used the model with weights pretrained on the ImageNet [11] dataset. The used parameters are:

- Optimizer: Stochastic Gradient Descent
- Learning rate: 0.005
- Momentum: 0.9
- Weight decay: 0.0005
- Batch size: 24
- Number of epochs: 10

This DL model converges in 10 epochs and achieves accuracy of 99.32%, outperforming classical algorithms by a huge margin.

V. Experimental results

We have compared three classical algorithms (SVM, *k*-NN, FCNN), and one DL algorithm (CNN). Detailed description of the used parameters was given in sections III and IV. The implementation was done in Python, with the help of the scikit-learn library for the classical algorithms, and Keras on top of TensorFlow for the DL model. The code was executed on the Google Colab platform which offers free CPU and GPU resources, Training was done on the CPU for the classical algorithms, and on the GPU for the DL model.

Data was divided into training and test sets in a 80-20 manner (80% of data was used for training, 20% for testing). For metrics we have used accuracy, precision, recall and F1 score. Precision, recall and F1 score are macro averaged. Results are given in Table 1:

	Accuracy	F1 score	Precision	Recall
SVM	0.917	0.894	0.903	0.89
k-NN	0.78	0.727	0.742	0.724
FCNN	0.914	0.892	0.899	0.892
CNN	0.993	0.99	0.991	0.99

Table 1: Metrics of tested algorithms

We can see that *k*-NN has a much lower score than the other choices. SVM and FCNN have comparable results, although still much lower than CNN which was shown to give the best results by far. The error rate for CNN was less than 1%, compared to 8-9% for SVM and FCNN, and more than 20% for *k*-NN.

VI. Conclusion

This paper presents the dominance of the DL method over the classical ML algorithms. Both the simplicity of the approach and the achieved accuracy confirm that the DL is the way to follow for image classification problems with relatively large datasets.

As the achieved accuracy of the DL method is already very high, trying to improve its results on the same dataset would be of little benefit. Further work with the DL model could be done by expanding the dataset with more diverse images, collected from multiple sources, in order to allow it to generalize better.

The considered ML algorithms achieved relatively high accuracy, but with error rates still an order of magnitude higher than the DL model. Further work in improving accuracy of the classical approach can be done by experimenting with other algorithms and by improving the features, as most likely they are the limiting factor of this approach.

Resources

The full code can be accessed on Google Colab by using the following link:
https://colab.research.google.com/drive/1L6s--5ZASj-pq_xoan9EhMmiMQd60L7v
or on GitHub:
https://github.com/drasko95/plantvillage-classical-vs-deeplearning

Literature

[1] United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019: Highlights (ST/ESA/SER.A/423).

[2] Savary, Serge, et al. "The global burden of pathogens and pests on major food crops." *Nature ecology & evolution* 3.3 (2019): 430.

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

[4] Fujita, E., et al. "A practical plant diagnosis system for field leaf images and feature visualization." *International Journal of Engineering & Technology* 7.4.11 (2018): 49-54.

[5] Haralick, Robert M., Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification." *IEEE Transactions on systems, man, and cybernetics* 6 (1973): 610-621.

[6] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

[7] Cunningham, Padraig, and Sarah Jane Delany. "k-Nearest neighbour classifiers." Multiple Classifier Systems 34.8 (2007): 1-17.

[8] Haykin, Simon. Neural networks: a comprehensive foundation. Prentice Hall PTR, 1994.

[9] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] Duan, Kai-Bo, and S. Sathiya Keerthi. "Which is the best multiclass SVM method? An empirical study." International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2005.

[11] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009.