ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

Leaf Disease Detection Using Deep Learning Algorithms

AMRUTHA GOPA, VIJAYA BHASKAR MADGULA, DIGALA RAGHAVA RAJU

Assistant Professor [1,2,3]

amruthacse9f@gmail.com, vijaya.bhaskar2010@gmail.com, raghava.digala@gmail.com
Department of Computer Science and Engineering, Sri Venkateswara Institute of Technology, N.H 44, Hampapuram, Rapthadu, Anantapuramu, Andhra Pradesh 515722

ABSTRACT

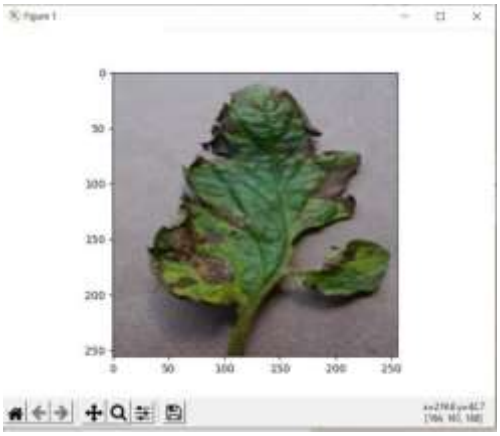
Recognising plant diseases using deep convolutional networks to analyse leaf photos is the main focus of this article. We train a deep convolutional neural network to detect crop illnesses using a large publicly available dataset made up of photos of healthy and sick plant leaves taken in controlled environments. The convolution method involves applying filters to an image in order to produce a feature map. The input picture or feature map is first passed through a linear filter with a bias added, then a nonlinear filter is applied. Following this, Max Pooling is used, which simplifies calculations for higher layers, removes the minimum value, and offers translational invariance. There are three methods in which our accuracy in determining whether tomato and potato leaves are healthy was proven: Early blight or the bacterial spot is contained inside. Our approach achieves an accuracy of 91% thanks to the given outcome.

ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

I. INTRODUCTION

For the most part, fungus, bacteria, and viruses are to blame when plants become sick. Although the word "disease" is often reserved for the killing of living organisms, the bacterial and fungal processes that cause dry rot and the spoilage of harvested crops while in transit or storage are quite similar to those that destroy plants while they are in growth. There would be more chances of infection and more damage caused by parasitic diseases if there are environmental factors that promote the development of parasites or disease transmitters and factors that hinder the growth of the plants. The infectious agent (in the case of bacteria and viruses) or its reproductive structures (in the case of fungus) may transmit parasitic illnesses. The vectors for spread might be anything from the wind and rain to insects and even people.



Assists in optimising performance criteria via the use of gained expertise. Many different kinds of practical computing challenges may be addressed with the use of supervised machine learning. When given an unknown input instance, a supervised learning model may be trained to produce a prediction. In this method of learning, the learning algorithm trains the classification model using a dataset that contains input instances and the classifiers associated with them. Following this, the trained model will provide a forecast for the test dataset or unexpected input.

I. Usage Of ML Algorithms

Computer algorithms that learn better on their own are the focus of machine learning (ML). Artificial intelligence (AI) includes it. Without being explicitly coded, machine learning algorithms may generate predictions or judgements by building a mathematical model using sample data, or "training data". Many different applications make use of machine learning methods.

In cases when traditional algorithm development would be too time-consuming or costly, such computer vision and email filtering. A. Learning Under Supervision Machines in this subfield of ML learn on labelled training data. Discovering a mapping function that converts input variables into output variables is the fundamental goal of supervised learning algorithms. The following sample diagram exemplifies supervised learning:

A. Unsupervised Learning

ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

Instead of using a labelled dataset, models in Unsupervised Machine Learning are left to discover patterns and insights on their own. using the information that was provided. When compared to the human brain, it is said to learn things more quickly. A dataset's underlying structure, similarity-based data grouping, and compressed representation are the end goals of an unsupervised machine learning technique.

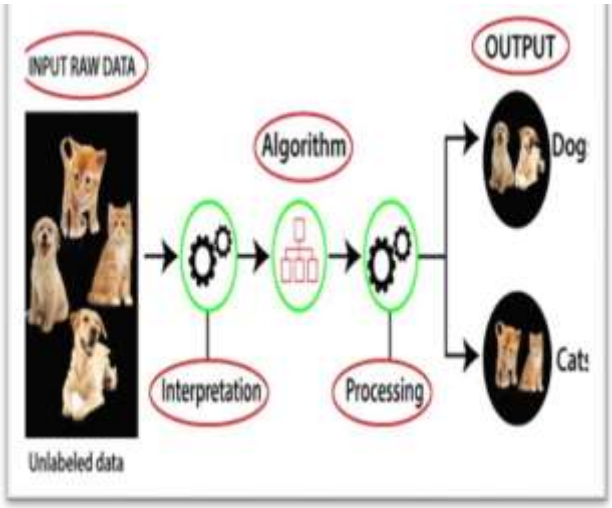


Fig3: Unsupervised Learning

In this case, we provide it with unlabeled data, and it will analyse it to uncover hidden patterns. Then, it will use appropriate techniques, including decision trees and clustering algorithms, to uncover those patterns. Due to the lack of labelled input data in unsupervised learning, it is better suited to more complicated problems than supervised learning. Generally speaking, Wireless Sensor Networks benefit better from Unsupervised Machine Learning.

B. Reinforcement Learning

It is a Machine Learning techniquein which the agent learns automatically using feedbacks without any labeled data. RL solvesa specific type of problem where decision making is sequential, andthe goal is long-term, such as game- playing, robotics e.t.c.

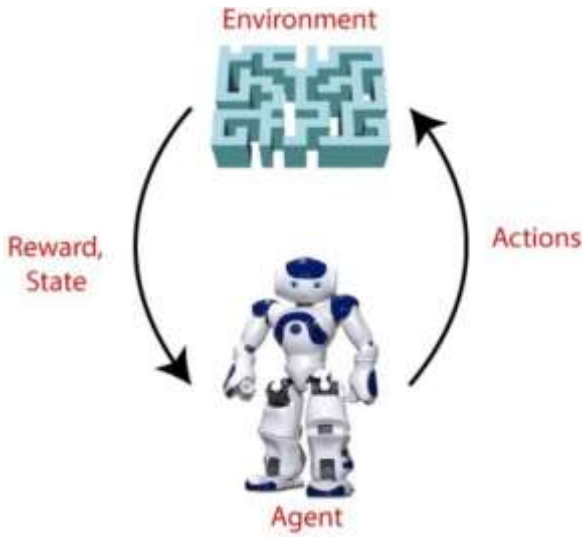


Fig3: Reinforcement Learning

ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

Agent is an entity that can perceive. Environment is a situation in which an agent is present or surrounded with. Actions are the moves taken by an agent within the environment. Reward is the feedback returned to the agent from the environment. In RL, the agent is not instructed about the environment and what actions need to be taken. It is based on thehit and trial process. The agent takesthe next action and changes statesaccording to the feedback of the previous action. The agent may get adelayed reward. The environment is stochastic, and the agent needs to explore it to reach to get the maximum positive rewards. Thus forReinforcement ML technique it takes more delay to produce the results by default as it is more of a hit and trail process.

III Related Work

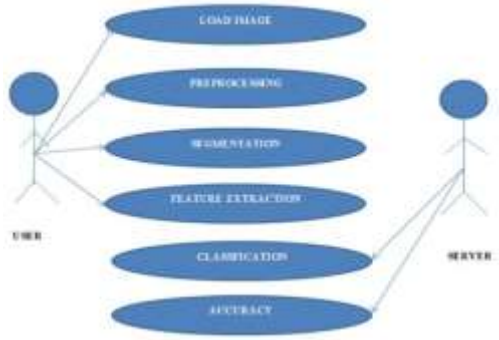
This section details the many deep learning-based algorithms for disease detection in plant leaves. 1. The most frequent plant kinds in the globe and Iraq in particular are tomatoes, peppers, and potatoes; this article by Marwan Adnan Jasin et al. finds illnesses associated to these plants. Images of plants and illnesses are included in this dataset with 20636 total. Using a convolutional neural network (CNN), the authors of the proposed system were able to classify plant leaf illnesses into fifteen distinct categories: twelve for various plant diseases (e.g., fungus, bacterium, etc.) and three for healthy leaves. With a training accuracy of (98.29%) and a testing accuracy of (98.029%) across all datasets, they achieved remarkable precision in both the training and testing phases. 2. A technique was developed by Sammy V. Militante and colleagues to identify and distinguish between many types of plants, including tomato, sugarcane, apple, maize, grapes and potatoes. Multiple plant diseases may also be detected by the technology. Scientists trained deep learning models to identify plant illnesses and their absence using a dataset consisting of 35,000 photos of diseased and healthy leaves. With a 96.5% success rate, the trained model has completed the

Vese (CV) algorithm. Finally,the segmented leaves are input into the transfer learning model and trained by the dataset of diseased leaves under simple background.Furthermore, the model is examined with black rot, bacterial plaque, and rust diseases. The results show that the accuracy of the method is 83.57%,

IV. Proposed Solution AndAnaysis Of Result

Image Acquisition-The images of theplant leaf are captured through the camera.This image is in RGB (Red, Green, and Blue) for colour transformation structurefor the RGB leaf image is created, andthen, a device-independent colour space transformation for the colour transformation structure is applied

B. Image Pre-processing To remove noise in the image or other object removals, different pre-processing techniques isconsidered. RGB to Gray Converter- Weighted method or luminosity method-You has seen the problem that occurs inthe average method. The weighted method has a solution to that problem.



Libraries Used : The Python-based deep learning API known as Keras sits atop the TensorFlow machine learning framework.
When starting up a neural network, sequential is the way to go.The convolutional network that processes the

ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

photos is constructed using Convolution2D. To include the pooling layers, a MaxPooling2D layer is used. To transfer the pooled feature map to the fully linked layer as a single column, the flatten function is used. The fully connected layer is included into the neural network using Dense.

Test And Train Model Used :

You may evaluate your model's precision using the Train/Test approach. You divide the dataset into two halves, a training set and a testing set, which is why it's called Train/Test. The ratio of 80% to 20% is for training and testing. The training set is used to train the model. The testing set is used to test the model. If you want to build a model, you have to train it. If you want to be sure the model is accurate, you need test it. The number of samples processed before updating the model is called the batch size. How many epochs are there? the total amount of times the training dataset has been traversed. Batch sizes can't be less than one but more than or equal to the training dataset's sample size.

Process Followed:

1. Convolution :
2. Calling the add method with the classifier object and passing in parameters allows us to add the convolution layer. The nb_filter parameter is the first. nbfilter is the desired output in terms of feature detectors. The feature detector matrix dimensions are the second and third parameters.Typically, convolutional neural networks (CNNs) are trained using 32 feature detectors. Following this, we have input_shape, which specifies the input image's shape. While the photos are being preprocessed, they will be transformed into this shape. A 2D array will be created from a black-and-white picture, and a 3D array will be created from a coloured one. Step two, maxpooling, involves shrinking the feature map. For maximum pooling, we often choose a 2x2 pool size. Because of this, we may shrink the feature map without sacrificing any of the image's semantic content. Step three, flattening, involves merging all of the pooled feature maps into one vector. All the feature maps are collapsed into one column using the Flatten method.
3. **Full Connection :**The next step isto use the vector we obtained above as the input for the neural network by using the Dense function inKeras. The second parameter is the activation function. We usually use the ReLu activation function in the hidden layer. The next layer we haveto add is the output layer. In thiscase, we'll use the sigmoidactivation function since we expect abinary outcome. If we expectedmore than two outcomes we would use the softmax function.
4. Compiling CNN :

model.compile(loss='binary_cross entropy',optimizer='Adam',metrics=['accuracy'])

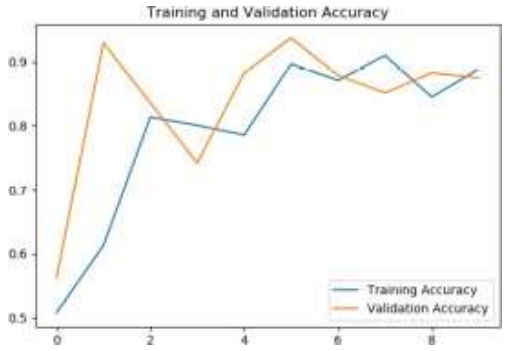
We then compile the CNN using the compile function. This function expects three parameters: the optimizer, the loss function, and the metrics of performance.Theoptimizer is the gradient descent algorithm we are going to use. We use the binary_crossentropy loss function since we are doing a binary classification.

5. **Fitting the CNN :** This function works by flipping, rescaling, zooming, and shearing the images. The first argument rescale ensures the images are rescaled to have pixel values between zero and one. horizontal_flip=True means that the images will be flipped horizontally. All these actions are part of the image augmentation.

We then use the ImageDataGenerator function toMake the test set's pixels range from zero to one by rescaling them.Making the training set is the next step. The function train_datagen, which was just built earlier, and the flow_from_directory are used to do this. With the help of the flow_from_directory function, we can get the training set photos from the directory where we are now working. The route to the training data set is the first parameter. The second argument specifies the desired picture size for

A ES
ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020
https://doi.org/10.5281/zenodo.12707096

the convolutional neural network (CNN). The number of pictures that will be processed by the network prior to updating the weights is represented by batch size. If the classification is binary or not, it is indicated by the class_mode argument.



Finally, we need to fit the model to the training dataset and test its performance with the test set. We achieve this by calling the fit_generator function on the classifier object. The first argumentit takes is the training set. The second argument is the no of arguments in our training set.Epochs is the no of epochs we want to use to train the CNN.

Validation_data is the test data set.nb_val_samples is the number of images in the test set.

6. Predicting The Output :

We can use the predict method to make predictions using new images. In order to do this we need to preprocess our images before we pass them to the predict method. To achieve this we'll usesome functions from numpy. We also need to import the image module from Keras to allow us to load in the new images.

The next step is to load the image that we would like to predict. To accomplish this we use the load_img function from the image module. The first argumentthis function takes is the path to the location of the image and the secondargument is the size of the image. The size of the image should be the same as the size used during the training process.

```
test_image= image.load_img('brain_image1.jpg', target_size=(256, 256))
```

We use the expand_dims method from numpy to add this new dimension. It takes the first parameter as the test image we are expanding, and the second parameter is the position of the dimension we are adding. The predict method expects this new dimension in the first position, which corresponds to axis 0.

```
test_image= np.expand_dims(test_image,axis=0)
```

Now we use the predict method to predict which class the image belongs to.

```
prediction = classifier.predict(test_image)
```

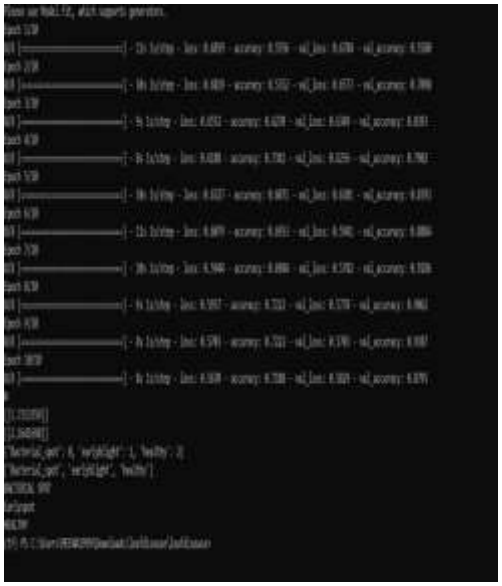
After running this function we'll getthe result:

ISSN: 2096-3246
Volume 52, Issue 01, Feb, 2020

https://doi.org/10.5281/zenodo.12707096

Bacterial spot – 0(0-1).Early bright – 1 (1-2).

Healthy – 2 (>2).



References:

"Detection and Classification of Plant Leaf Diseases Using Image Processing and Deep Learning Techniques" (Marwan, this year). The work of Adnan Jasim and Jamal Mustafa AL-Tuwaijari will be published in the IEEE 2020 journal.
The Hindawi 2020 publication "Plant disease identification based on deep learning algorithm in smart farming" was penned by Yan Guo, Jin Zhang, Chengxin Yin, Xiaonan Hu, Yu Zhou, Zhipeng Xue, and Wei Wang.
The 2019 article "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks" was published in the IEEE Magazine and was written by PENG JIANG, YEUHAN CHEN, BIN LIU, DONGJIAN HE, + Chunquan Liang.
"Deep Learning for Plant Leaf Detection and Disease Recognition," was given by Sammy V. Militante, Bobby D. Gerardo, and Nanette V. Dionisio at the 2019 IEEE conference.
[5] A 2018 publication by IEEE including the work of XIHAI Zhang, Yue QiAO, Fan Feng, Chengguo Fan, and Ming Mingming Zhang titled "Improving Deep Convolutional Neural Networks for Disease Detection in Maize Leaves" appears.