

Bitter Melon Crop Yield Prediction using Machine Learning Algorithm

Marizel B. Villanueva

College of Informatics and Computing Sciences
Batangas State University
Batangas, Philippines

Ma. Louella M. Salenga

College of Information and Communication Technology
Holy Angel University
Pampanga, Philippines

Abstract—This research paper aimed to determine the crop bearing capability of bitter melon or bitter gourd more commonly called “Ampalaya” in the Filipino language. Images of bitter melon leaves were gathered from Ampalaya farms and these were used as main data of the research. The leaves were classified as good and bad through their description. The research used Machine Learning Algorithm through Convolutional Neural Network. Training of data was through the capabilities of Keras, Tensor Flow and Python worked together. In conclusion, increasing number of images could enable a machine to learn the difference between a good and a bad Ampalaya plant when presented an image for prediction.

Keywords—Agriculture; Artificial Intelligence; Keras; machine learning algorithm; machine learning; neural network; convolutional neural network; prediction; Python; tensor flow

I. INTRODUCTION

Machines have been very useful to humans from time immemorial. It has been proven that in the pursuit of man to make life easier he has found means to create new things which has helped him achieved progress. Machines vary from each other upon their inception. Their use to man is determined by the need that arises. In this paper, the authors will be able to explain its use specifically the computer in determining crop bearing capabilities of bitter melon or bitter gourd. The researchers will utilize the capabilities of computers for machine learning. Since machines are mostly used for mechanical work and to effectively use them, these are data that need to be trained, to give artificial intelligence.

Artificial Intelligence the ability of a digital computer or robot to perform tasks usually associated with human intelligence. It is frequently applied to the development of systems equipped with the intellectual processes and characteristics of humans, such as the ability to reason out, discover meaning, and generalize, or learn from experience [1]. The application of this in agriculture would be helpful a lot of researchers in this area making it easier to determine decisions for crop production. The researchers thought of creating this project since the Philippines is an agricultural country, with a land area of 30 million hectares, where 47 percent of it is agricultural land. Growing crops is one of the major bread and butter of the rural areas focusing on the development of agricultural studies to help crop management as a step to rural livelihood development [2].

“Ampalaya” the common name as it is called in the Philippines is better known as bitter melon. Production of bitter melon is profitable when grown in small or large scale either in lowland or upland rice-based areas [3]. Essentially, growing bitter melon does not need to have a big area of land, which makes it more attractive to Filipinos who do not have a large lot for growing crops. Aside from this, according to the Philippine Herbal Medicine Organization, Bitter Melon has been endorsed by Department of Health through its Traditional Health Program. It is most known as a treatment of diabetes (diabetes mellitus), for the non-insulin dependent patients [4].

Excerpts of reference researches below show how the sciences of Machine Learning, specifically Neural Networks have been utilized in the field of Agriculture.

A. Rice Blast Prediction based on Gray Ant Colony and RBF Neural Network Combination Model

This agricultural study made was in China was Rice Blast Prediction (RBF) where gray ant colony RBF learning algorithm was used. The RBF is movable and can determine its position by self-organization learning. Self-organizing learning part allocates the network resource in a sense; the purpose is to enable the RBF center to locate in the important region of the input space. The conclusion, the prediction model during 2002-2011 years, has accuracy of up to 96.77 per cent, compared with two the other single prediction models, this scheme has better feasibility and can be applied in rice blast prediction [5].

B. Statistics and Neural Networks for Approaching Nonlinear Relations between Wheat Plantation and Production in Queensland of Australia

This research dealt with treating the historical wheat data in Queensland over 130 years as non-temporal collection of mappings between wheat plantation area and production. Correlation analysis and neural network techniques to reveal whether significant nonlinear relations exist between the two factors were used. The simulation of the Neural Network process involved two phases: 1) training the network with known datasets; and 2) testing the trained network for model generalization. If the testing is conducted using some data selected from the training datasets, it is called the in-sample testing. If different known datasets are used for testing, it is then called the out-sample testing. The study made use of the Levenberg-Marquardt algorithm, chosen to train the selected Multilayer Perceptron (MLP) because this algorithm has been

reported to be the fastest method for training moderate-sized feedforward neural networks. Their MLP models are built using the NN tools in MATLAB. As concluded, their analysis demonstrated that a power correlation, a third-order polynomial correlation, and a three-layer multilayer perceptron model are all of significance, but it is the multilayer perceptron model that can produce accurate prediction [6].

II. METHOD

Focused on the objective of being able to predict crop yield of bitter melon using images, there are two fields of Artificial Intelligence that could be used for this purpose.

Traditionally, when images are a focal point to achieve a goal, Computer Vision is an obvious choice. To define, Computer Vision is a field of Artificial Intelligence and Computer Science that aims to give a computer the similar, if not better capability of humans to use their eyes and their brains to see and visually sense the world around them [7].

Computer vision algorithms their general functions extracts feature vectors from images and utilize these feature vectors to classify images. To be able to make a machine predict, and aside from classifying and grouping together images to be able to identify them, it must learn what they are when the image is presented to it. Based on the Fields of Artificial Intelligence shown in the figure below, the field of machine learning is not directly connected to vision, though they can work together to make a more intelligent machine, wherein machine learning is the focus [8].

Deep Learning is a field of Artificial Intelligence that makes use of Deep Neural Networks It is a subfield of Machine Learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks [9]. Then again, what is a Neural Network? Why is it needed in this research?

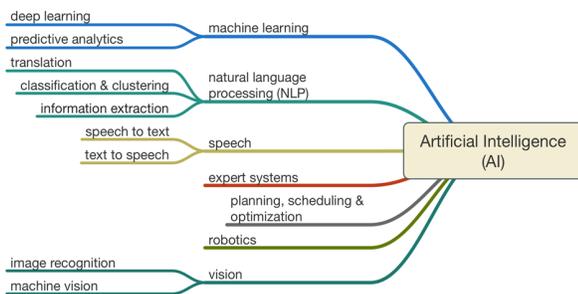


Fig. 1. Fields of artificial intelligence [10].

As seen in Fig. 1, in a white paper by Mills in 2016, all algorithms explained before in this section fall under the classification of Artificial Intelligence but to make a model and the main algorithm must be identified. This is where the Artificial Neural Network has been chosen, that describes the function [10], [11].

A Neural Network, or ‘artificial neural network’ known as ANN is a computing system made up of several simple, highly interconnected processing elements, which process information with dynamic state response to external in-puts. The research aimed to teach the machine to recognize a number of images

and to understand what type of plant they will be, good or bad – fruit bearing or not. These images when trained will create the so- called neurons. This process can increase the accuracy of predicting image samples being fed to a computer [12].

According to the research of Andy Lee from Swarthmore College, he found out that Deep Neural Networks had greater accuracy in identification of images. His study compared Traditional Computer Vision Algorithms with that of Convolutional Neural Networks, made a parallel implementation of both and concluded that with the investment in time and training, Convolutional Neural Networks were more accurate and performed better when applied whereas Traditional Computer Vision worked better in crude on-the-fly mobile robotic applications [13]. Based on another research from University of Toronto, the study used Convolutionary Neural Network to train data which is numerous in size. It enabled a network to increase its layers, learning more, increasing in accuracy and decreasing in error [14].

To accurately describe how a Neural Network has a better efficiency rating, the computational process is described here as a process of finding the best set of weights for the neural network which is referred to as training or learning. The approach used by most software to estimate the weights is backpropagation. Each time the network cycles through the training data, it produces a predicted value for the target variable, which is compared to the actual target value, and an error is computed for each observation. The errors are fed back through the network and new weights are computed to reduce the overall error. Despite the neural network terminology, the training process is actually a statistical optimization procedure. Typically, the procedure minimizes the sum of the squared residuals. Fig. 2 shows the interval activity of the neuron can be computed using the formula [15].

$$v_k = \sum_{j=1}^p w_{kj} x_j$$

Fig. 2. Internal neuron activity formula [15].

An activation function is then applied to the resulting value of v_k to produce the output value y_k . In this study, the activation function used is the sigmoid function. An example of the sigmoid function is using the hyperbolic tangent function is shown in Fig. 3.

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)}$$

Fig. 3. Sample hyperbolic tangent function [15].

This sigmoid function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range.

For an artificial neuron, the weight is a number, and represents the synapse. A negative weight reflects an inhibitory connection, while positive values designate excitatory connections. The following components of the model represent the actual activity of the neuron cell. All inputs are summed altogether and modified by the weights. This activity is referred as a linear combination. Finally, an activation function controls

the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1 [15].

Mathematically, this method is described in Fig. 4 below.

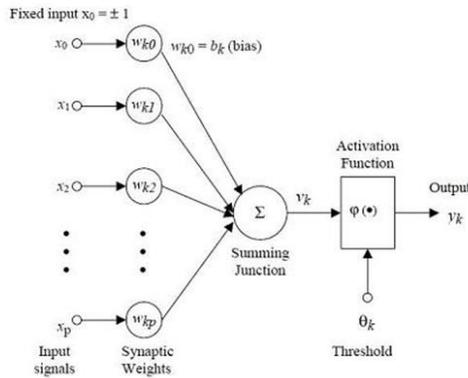


Fig. 4. Artificial neural network [15].

Since the input images are to return only two values for output, which is a good or bad Ampalaya plant, the process of the mathematical solution discussed earlier helps in the determination that the activation function will be effective in making a machine decide between an acceptable range of output which may be between 0 and 1, representing good and bad.

Having established the background of research and understanding the method that will be used, the researchers finalized their procedure to use Convolutionary Neural Networks for Deep Learning to train images of bitter melon or Ampalaya leaves from local farms within the Philippines.

III. DATA ACQUISITION

The need to identify which type of bitter melon plant the research should focus on led to the use of judgmental sampling technique to narrow down the choices of samples that will be gathered.

According to the Production Guide on Ampalaya and by the EntrePinoys and the compilation paper written by Kishore Hemlani (2012), there are the different varieties of the Bitter Melon or Bitter Gourd; these are shown in Table I [16], [17].

TABLE I. AMPALAYA VARIETY TABLE

Variety	Maturity (days)
Sta. Rita	70-75
Makiling	65-70
Sta. Isabel	70-75
Jade Star (A)	70-75
Mayon	65-70
Million Green	65-75
Galaxy (Galactica)	70-75

Among the varieties identified in Table I, the study concentrated on the bitter melon called Galaxy or locally known as “Galactica”, according to the Gazette magazine, this type was one of the most high-yielding varieties until recent studies [17]. With the restriction in accessibility, this was also chosen due to the demographic component whereby sampling needed to be done. The researchers have more accessibility to the plantation areas where this variety was being planted, to be specific, Region IV– A, Batangas City.

A. Data Acquisition Process

1) Interview with the Local Ampalaya farmer

In gathering data for the research, interviews were made with Local Ampalaya farmers in Batangas City. The respondents were Felix Morcilla and Virgilio Villanueva. Mr. Felix Morcilla has been farming bitter melon for 15 years while Mr. Virgilio Villanueva has been farming for 5 years. It was asked from these experts on the field of Ampalaya farming how their crops were determined to have good crops. Both noted that leaves alone could already identify the ability of an Ampalaya plant to bear good or bad crops.

2) Sample Collection Procedure

Since the main object of the research deals with the images of the bitter melon plant, the authors went Ampalaya Farm Lands in Batangas City, Philippines to gather these samples from the actual bitter melon plants. The authors, together with the Farm Land owners went around analyzing the plants. Each vine of the bitter melon would be analyzed for its fruit bearing capacity, and a string of leaves of this plant was taken, and there were only one to two samples per string or vine. This is to ensure the spread and variety of the samples. In the analysis of the plants, only the leaves were the focus. Soil type, irrigation, season of planting, age of the plant, temperature and the use of fertilizer was not covered in the research.

3) Sample Image Capture Procedure

After each plant was identified as fruit bearing or not, from the string of leaves taken, one leaf is placed on top of a white piece of paper then taken a picture of. Each leaf was placed in at least three angles to make sure that the leaf can be identified properly.

It might be a wonder why use Ampalaya leaves? Why not the full vine? The full vine is made up of a string of leaves, and that alone is enough to say that they all belong to the same plant. To identify the fruit bearing capacity using a vine is too big a scope, its analysis of this will also takes time, since the machine will have to deal with multiple leaves and the vine itself just to say, if it will bear fruit or not.

According to the guidelines in growing this plant, most Agricultural authors of the government pamphlets [18], [19] mentioned the analysis of its leaves as a determination of the plant’s health and ability to bear fruit. Together with the interview with the Ampalaya farm owners strengthened the claim that leaves of Ampalaya alone could determine the ability of the plant to bear fruit or not bear fruit at all. Essentially, the colors and description of the Ampalaya leaves, can be classified into the following Table II:

TABLE. II. AMPALAYA LEAF DESCRIPTION [19], [20]

Leaf Description	Fruit Equivalent
Small Dark Green	Bad Class, No Fruits
Normal or DeformedDark Green	Bad Class, No Fruits
Normal Sized Green	Good Class, Good Fruits
Normal Sized Light Green	Good Class, Good Fruits
Irregular-sized Yellow Green	Bad Class, Small or Diseased Fruits
Deformed Brown	Bad Class, Small or Diseased Fruits

Throughout the multiple days of image gathering, there were a total of 293 leaves taken account for. A total of 131 leaves classified as Bad and 162 classified as Good. These images were used to test the model for determining an Ampalaya leaf whether the actual vine will give a good or bad fruit.

The following images show examples of good and bad Ampalaya leaves:



Fig. 5. Good Ampalaya Leaf.



Fig. 6. Bad Ampalaya Leaf.

Fig. 5 and 6 were actual images that were taken by the researchers. These are only two of the 293 images trained to be recognized by the Convolutionary Neural Network.

IV. RESULTS AND DISCUSSION

The research used Experimental Method in doing the research. Multiple samples were tested, trained and processed to be able to meet the objectives. To test the samples of images, the Machine Learning Algorithm used in creating the model was Neural Networks through the media discussed in the subsections below. The aim is to prepare the machine for Sequential, Convolutional Neural Networks.

A. Testing Resources

1) MATLAB

It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include: Data analysis, exploration, and visualization [20].

Initial Testing of the Data was tested in MATLAB. Since the actual machine requirement for MATLAB cannot be supported by the researcher's machines, it was not used as the implementing model for the project. The model created here was used as comparison of the accuracy that will be generated using Tensor Flow and Keras.

After running the model on a single GPU processing machine, the following are the results were attained:

TABLE. III. TRAINING RESULTS

Epoch	Iteration	Time Elapsed	Loss	Accuracy	Rate
1	1	0.62	0.9297	56.25	0.001
17	50	74.24	0.0003	100	0.001
34	100	151.11	0.0023	100	0.001
50	150	225.24	0.0002	100	0.001
67	200	298.99	0.0001	100	0.001
84	250	372.68	0.0001	100	0.001
100	300	447.02	0.0000	100	0.001

Table III yielded a 56.25 per cent on the first run and 100 on the second run of the model. This was based on the 293 leaves taken account for. With A total of 131 leaves classified at Bad and 162 classified as Good.

2) TensorFlow and Keras

Tensor Flow is an open-source software library for machine learning across a range of tasks and developed by Google to meet their needs for systems capable of building and training neural networks to detect and decipher patterns and correlations, analogous to the learning and reasoning which humans were used. It is currently used for both research and production at Google, while Keras is a high-level Python neural networks library that runs on top of either TensorFlow or Theano [21].

These were used by the researchers for the identification of single item input to test if the image is a good or bad Ampalaya. Training of the data was separated into 60-40 sets. 60% of the images that were used to train the machine's neural network and 40% was used to test it. The authors used supervised training to make sure that images are classified and identified with higher accuracy.

Each image was treated with the following algorithm for image uniformity, so when new data is presented even if taken from different devices, they can still be learned since they must be pre-processed before training.

```
if K.image_data_format() == 'channels_first':  
    input_shape = (3, img_width, img_height)  
else:  
    input_shape = (img_width, img_height, 3)
```

In detail, the algorithm above means:

data_format: A string, one of channels_last (default) or channels_first. The ordering of the dimensions in the inputs. channels_last corresponds to inputs with shape (batch, height, width, channels) while channels_first corresponds to inputs with shape (batch, channels, height, width).

In Keras, the image in Fig. 7 describes how the 2D convolutional layers were created.

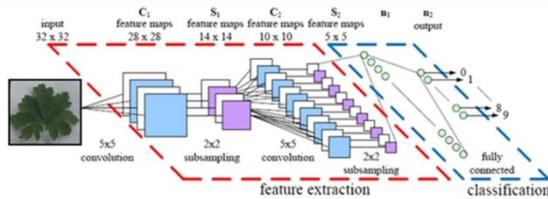


Fig. 7. Convolutional 2d Process [22].

In a research by Y LeCun, et al, it indicated that a network can be fed with the pixels in raw images. These images create patterns, whereby disregarding the image itself but concentrate local nearby pixels, forming layers upon layers of highly correlated ones [22].

In Convolutional Network Architectures, layer patterns are created. In the architecture, stacks of layers are called CONV-RELU, which are followed with POOL layers and repeats this pattern until the image has been merged spatially to a small size. At some point, it is common to transition to fully-connected layers. The last fully-connected layer holds the output, such as the class scores. In other words, the most common ConvNet architecture follows the pattern:

INPUT -> [[CONV ->RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC

where the * indicates repetition, and the POOL? indicates an optional pooling layer. Moreover, N >= 0 (and usually N <= 3), M >= 0, K >= 0 (and usually K < 3). For example, here are some common ConvNet architectures you may see that follow this pattern:

- INPUT -> FC, implements a linear classifier. Here N=M=K = 0.
- INPUT -> CONV -> RELU -> FC
- INPUT -> [CONV -> RELU ->POOL]*2 -> FC -> RELU -> FC. Here we see that there is a single CONV layer between every POOL layer.
- INPUT -> [CONV -> RELU -> CONV -> RELU ->POOL]*3 -> [FC -> RELU]*2 -> FC. Here we see two CONV layers stacked before every POOL layer. This is generally a good idea for larger and deeper networks, because multiple stacked CONV layers can develop more complex features of the input volume before the destructive pooling operation [23].

The algorithm that shows how this is done in Python code is shown in Fig. 8, which is the actual code used by the researchers in the study.

```

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense

model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(3, 150, 150)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# the model so far outputs 3D feature maps (height, width, features)
    
```

Fig. 8. Conv2d Python code.

To train the data, the actual code used by the researchers is shown in Fig. 9.

```

batch_size = 16

# this is the augmentation configuration we will use for training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

# this is the augmentation configuration we will use for testing:
# only rescaling
test_datagen = ImageDataGenerator(rescale=1./255)

# this is a generator that will read pictures found in
# subfolders of 'data/train', and indefinitely generate
# batches of augmented image data
train_generator = train_datagen.flow_from_directory(
    'data/train', # this is the target directory
    target_size=(150, 150), # all images will be resized to 150x150
    batch_size=batch_size,
    class_mode='binary') # since we use binary_crossentropy loss, we need binary labels

# this is a similar generator, for validation data
validation_generator = test_datagen.flow_from_directory(
    'data/validation',
    target_size=(150, 150),
    batch_size=batch_size,
    class_mode='binary')
    
```

Fig. 9. Model training Python code.

After identifying the code to treat the data, training was made, the images shown in Fig. 10 to 12 below are the actual outputs of testing. They were single input images from the Python Shell run time window.

Fig. 10. Model creation in Python using 125 epochs.

```

Using TensorFlow backend.
1/1 [=====] - 0s
[[1]]
Good
    
```

Fig. 11. Good Ampalaya predicted output.

