



WILD PLANTS EDIBILITY PREDICTION USING IBM WATSON STUDIO

A PROJECT REPORT

Done as a part of the Externship Program in

SMARTINTERNZ

Submitted By

A.Sravya Sri

A.Ashok kumar reddy

D.Dinesh Sai Sri Ram

E.Supriya

Roll No:19481A0505

Roll No:19481A0509

Roll No:19481A0552

RollNo:19481A0556

of

School of Computing Sciences and Engineering

IN

**Seshadri Rao Gudlavalleru Engineering college,
Gudlavalleru**

TABLE OF CONTENTS

1. Introduction.....	4
a. Overview.....	4
b. Purpose.....	5
2. Literature_Survey.....	6
a. ExistingProblems.....	6
b. Proposed Solution.....	6
3. Theoretical_Analysis.....	7
a. BlockDiagram.....	7
b. Hardware_and Software_Requirements.....	7
4. Experimental_Investigations.....	11
a. Analysis_of the_Project.....	11
i. Import_Libraries and Initialize_the_Model.....	11
ii. Add_CNN_Layers.....	11
1. Adding_Convolution Layer.....	11
2. Adding_Max Pooling Layer.....	12
3. Adding_Flatten Layers.....	12
iii. Adding_DenseLayers.....	12
1. Adding_DenseLayers.....	12
2. Adding_Hidden layers.....	12
iv. Configuring_The_Learning Process.....	13
v. Train_And Save_The_Model.....	13
1. Train_the_Model.....	13
2. Save_the Model.....	13
vi. Test_The Model.....	14
vii. Application_Building.....	14

viii.	Build_Python	
	Code.....	14
	1. Importing_Libraries.....	14
	2. Routing_to thehtml Page.....	14
	3. Showcasing_prediction on UI.....	14
	4. Main_Function.....	15
ix.	Run The_App.....	15
5.	Flowchart.....	15
6.	Result.....	16
	a. findings_(Output) of_the project.....	16
7.	Application.....	21
8.	Conclusion.....	22
9.	Future_Scope.....	22
10.	Bibliography.....	23

1.INTRODUCTION

a. Overview

Agriculture plays a critical role in the global economy. Unfortunately, it is currently pressured by a growing population that requires immediate attention. When technology was first integrated into agriculture more than one century ago, with the first tractor in 1913 (Santos et al., 2020), there was a large increase in food production. Nowadays, mechanical technology has dramatically impacted agriculture, improving its efficiency (Schmitz and Moss, 2015). However, it may not be enough to sustain the increasing global population predicted to reach 9.6 billion by 2050, increasing by approximately 1.8 billion people. This massive increase will require an immense demand for new food (Gikunda and Jouandeau, 2019), making further agricultural advances essential. Several studies on a concept known as smart farming use a combination of technologies, such as the Internet of Things (IoT), Big Data, and robotics, and plays a crucial step in increasing the sustainability and reliability of food production while simultaneously reducing its environmental impact (Walter et al., 2017). This modern approach assists in tackling the key issues facing agriculture and expands future agricultural advances. It focuses on DL tools, specifically CNNs, for identifying patterns in images, which have now become fundamental in agriculture (Liakos et al., 2018), aiding in the identification of plants and diseases through image recognition. However, the applications of smart farming are limited. It solely focuses on existing farms and currently consumed crops, despite the existence of over 50,000 edible plants worldwide, with only 15 of them providing 90% of the world's food energy intake (National Geographic Headquarters, Undated). Examining other edible flora and considering them as a resource for daily consumption can prove beneficial, assisting in the limitations of future food provisions. Furthermore, with the continued decline in the number of botany students (Drea, 2011; Lauer, 2015), the world's understanding of plants may diminish, demanding the need for new methods to understand the plants around us.

This paper presents a method to efficiently identify edible plants in the wild, using CNN architectures. The purpose of this project is to expand the use of DL within horticulture, enabling inexperienced individuals the ability to swiftly recognise wild edible plants within the vast expanse of flora in the world. Opening the opportunity for scientists and individuals interested in horticulture to extract, experiment, and reproduce new types of edible vegetation. Additionally, encouraging future research that concentrates on highlighting and obtaining a deeper understanding of plants.

Classifying plants can be challenging due to the wide variety of similarities between them. What distinguishes plants from one another are their features, such as colour, shape, style, and unique properties, e.g. ground ivy has whiskers, and coneflowers have cones on top of their petals. For botany experts, the difficulty with classifying plants lies in finding their distinct features. Unfortunately, this is the same for CNNs, except they identify th

e features through numeric pixel values.

b. Purpose

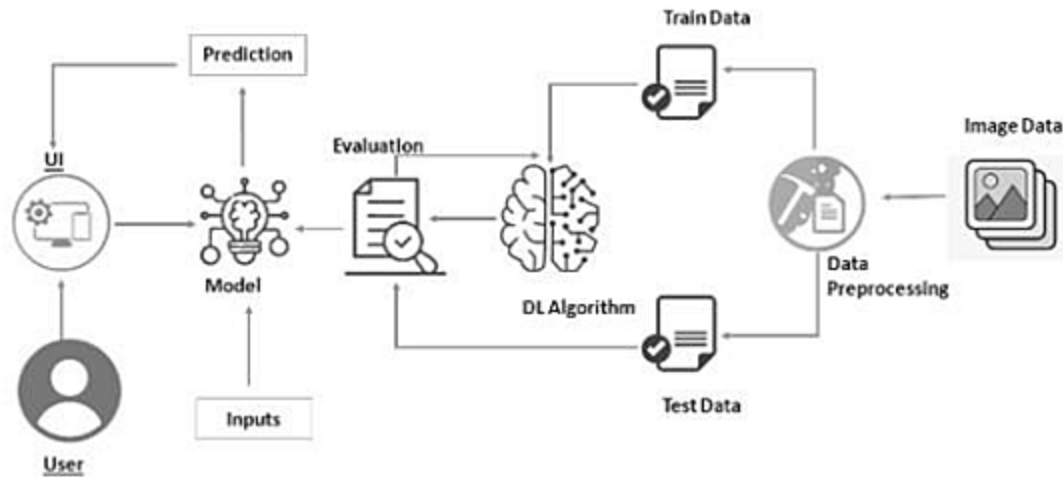
With the world population continuing to increase, additional food sources are required. This project aimed to successfully classify different classes of wild edible plants and understand each of their distinct features, allowing easy identification within a large expanse of vegetation. Three CNN architectures were implemented to investigate the unique elements of each plant. The process to achieve this can be divided into multiple objectives.

Objective 1: maximise each model's classification accuracy by creating a dataset containing 35 classes of wild edible plants, where each category has a minimum of 400 images per one. Additionally, employ data augmentation techniques onto the images, increasing the diversity of the dataset.

Objective 2: examine, select, and perform classification with three different CNN architectures and apply transfer learning to each one, bolstering the classification performance to achieve an accuracy of over 80%.

3.THEORETICAL ANALYSIS

a. BlockDiagram7



b. Hardware and Software Requirements

The artefact created utilises the Python programming language, specifically version 3.9.2, and has its functionality extended with the assistance of 8 packages (libraries). While the latest Python version is recommended, it is possible to run the artefact with a minimum of version 3.7. However, it is critical to ensure that the libraries installed utilise the same versions described in this section. The 8 libraries used include Flask, NumPy, gevent, Tensorflow, Werkzeug, Pandas, Scikitlearn, and Scikit-plot.

Another core piece of software used for creating the artefact is virtual environments. When developing software, it is common for developers to have packages and libraries already set up with the latest package versions. Unfortunately, it can be time-consuming and troublesome to switch between package versions when working on multiple projects. However, virtual environments help to mitigate this issue by preventing conflicting package versions. Anaconda

provides a platform to manage Python environments and enables accessibility through a Python kernel, integrated seamlessly with Jupyter Notebooks.

CNNs can take a long time to train and tune when only using a CPU. To increase this, NVIDIA has a parallel computing toolkit that allows the transfer of data onto GPUs, called CUDA. Fortunately, PyTorch has CUDA functionality built into its library that automatically keeps track of the available GPUs on a system and the tensors allocated to them. When experimenting with different CUDA versions, CUDA toolkit version 10.1 proved fundamental and is required to run the artefact within this project to minimise the training, tuning and evaluation computation speed.

The system hardware used to create the artefact is as follows:

1. Intel Core i7-4790k CPU 4.00GHz
2. 16GB of RAM
3. NVIDIA GeForce GTX 970 (4GB) GPU
4. Windows 11 Operating System

It is recommended, but not required, to have similar hardware specifications to run the artefact to maximise its efficiency. If a GPU is not available, it is possible to run the artefact on a typical CPU. However, the tuning time taken to complete the 36 model variants on a GPU took approximately two days. Using a CPU, the time taken will increase by a minimum of 4x as long.

4. EXPERIMENTAL INVESTIGATIONS

a. Analysis of the Project

i. Import Libraries and Initialize The Model

Keras has 2 ways to define a neural network:

1. Sequential
2. Function API

The `Sequential` class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the `Sequential` constructor to create a model, which will then have layers added to it using the `add()` method.

ii. Add CNN Layers

We will be adding three layers for CNN

1. Convolution layer
2. Pooling layer
3. Flattening layer
4. Full Connection

1. Adding Convolution Layer

In the convolution2D function we given arguments like, 64,(3,3), that means we are applying 64 filters of 3x3 matrix filter, and input_shape is the input image shape with rgb, here 128x128 is the size and 3 represent the channel, rgb color images.

And Activation function defines the output of input or set of inputs or in other terms defines node of the output of node that is given in inputs. They basically decide to deactivate neurons or activate them to get the desired output.

2. Adding Max Pooling Layer

Pooling reduces the dimensionality of images by reducing the number of pixels in the output from the previous convolutional layer. It keeps only the necessary details. Pooling is a technique in CNN which helps us to avoid over fitting of data, spatial invariance and distortion. After applying max pooling we will get another feature map called Pooled Feature Map.

3. Adding Flatten Layers

Now the pooled feature map from the pooling layer will be converted into a one single dimension matrix or map, where each pixel in one single column, nothing but flattening. Flattening layer converts multi dimension matrix to one single dimension layer.

iii. Adding Dense Layers

1. Adding Dense Layers

The name suggests that layers are fully connected (dense) by the neurons in a network layer. Each neuron in a layer receives an input from all the neurons present in the previous layer. Dense is used to add the layers.

2. Adding Hidden layers

This step is to add a dense layer (hidden layer). We flatten the feature map and convert it into a vector or single dimensional array in the Flatten layer. This vector array is fed into the neural network and applies an activation function, such as sigmoid or other, and returns the output.

Understanding the model is very important phase to properly use it for training and prediction purposes. Keras provides a simple method, summary to get the full information about the model and its layers.

iv. Configuring The Learning Process

1. The compilation is the final step in creating a model. Once the compilation is done, we can move on to training phase. Loss function is used to find error or deviation in

the learning process. Keras requires loss function during model compilation process.

2. Optimization is an important process which optimize the input weights by comparing the prediction and the loss function. Here we are using adam optimizer
3. Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in training process

V. Train And Save The Model

1. Train the Model

Now let us train our model with our image dataset. `fit_generator` functions used to train a deeplearning neural network.

- a. **steps_per_epoch:** It specifies the total number of steps taken from the generator as soon as one epoch is finished and next epoch has started. We can calculate the value of `steps_per_epoch` as the total number of samples in your trainingfolder divided by the batchsize.
- b. **Epochs:** an integer and number of epochs we want to train our model for.
- c. **Validation_data** can be either an inputs and targets list a generator an inputs, targets, and sample_weights list which can be used to evaluate. The loss and metrics for any model after any epoch has ended.
- d. **Validation_steps:** Only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

2. save the Model

The model issaved with .h5 extension as follows:

An H5 file is a data file saved in the Hierarchical Data Format (HDF).It contains multidimensionalarrays of scientific data

vi. Test The Model

The last and final step is to make use of our saved model to do predictions. For that we have a class in keras called `load_model`. `load_model` is used to load our saved model h5 file (edible_non.h5).

vii. Application Building

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the user where he has to upload an image. The uploaded image is given to the saved model and prediction is showcased on the UI.

This section has the following tasks:

1. Building HTML Pages
2. Building server-side script

Viii. Build Python Code

We will be using python for server side scripting. Let's see step by step process for writing backend code.

1. Importing Libraries

Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of current module (`__name__`) as argument. Pickle library to load the model file.

2. Routing to HTML page

Here we will be using declared constructor to route to the html page which we have created earlier.

2. Showcasing prediction on UI

When the image is uploaded, it predicts the category of uploaded image is either 'Asparagus_edible', 'Blue Vervain_edible', 'Cattail_edible', 'Chicory_edible_non edible', 'Fireweed_edible_non edible', 'green castor bean_non edible'. If the image

predictsvalue as 0, thenit is displayed as “Left Bundle Branch”. Similarly, if the predicted value is 1, it displays “Normal” as output and so on.

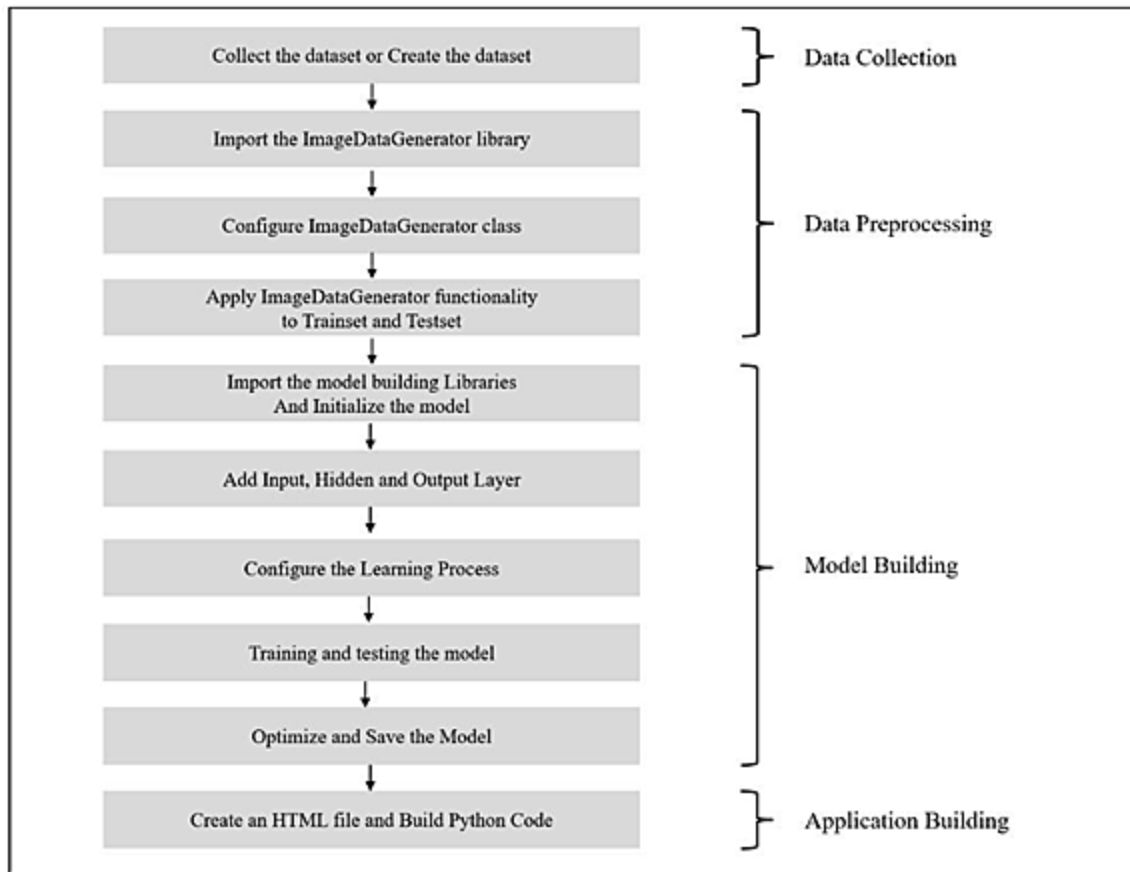
3. Main Function

This is used to runthe application in localhost.

iX.Run The App

1. Open anaconda promptfrom start menu.
2. Navigate to the folder where your app.py resides. •Now type “python app.py”command.
3. It will show thelocal host where your app is running on [http://127.0.0.1.5000/](http://127.0.0.1:5000/)
4. Copy that local host URL and open that URL in browser. It does navigateyou to the wheretheyou can view your web page.
5. Enter the values, click on predict button and see the result/prediction on web page

5. FLOWCHART



6. RESULT

Final findings(Output) of the program

Wild Plants Produce Edibility Prediction

Wild food often has superior nutritional qualities, whether eaten cooked or raw. Such foraging is a great way to avoid the drawbacks of agribusiness produce, such as hybridization, genetic engineering, commercial fertilizers, pesticides & herbicides, lack of freshness, fungicides, wax, and socially transmissible diseases. Since most of the Wild plant produce are not edible like Moonseed, Horse Nettle, Pokeberries and Wild Cherry etc....


What are Edibility Plants

Uses of Edible Wild Plants

Effects of Non Edible Plants


What are Edibility Plants

An organism of the vegetable kingdom suitable by nature for use as a food, especially by human beings. Not all parts of any given plant are edible but all parts of edible plants have been known to figure as raw or cooked food: leaves, roots, tubers, stems, seeds, buds, fruits, and flowers. The most commonly edible parts of plants are fruit, usually sweet, fleshy, and succulent. Most edible plants are commonly cultivated for their nutritional value and are referred to as vegetables..



"Edible - Non Edible Plants"

"To ensure good health: eat lightly, breath deeply"



click the browse button and upload the image of your plant to predict..

Choose image...

Choose File No file chosen

EDIBLE PLANTS

Edible plant stems are one part of plants that are eaten by humans. Most plants are made up of roots, stems, leaves, flowers, buds and produce fruits containing seeds. Humans most commonly eat the seeds (e.g. maize, wheat, coffee and various nuts), fruit (e.g. tomato and apple), leaves (e.g. lettuce, spinach, and cabbage), or roots (e.g. carrots and beets), but humans also eat the stems of many plants (e.g. asparagus). There are also a few edible petioles (leaf stalks) such as celery, as well as some edible flowers.

NON-EDIBLE PLANTS

There are a number of non-edible plants that can be found in many regions. Most of these plants are inedible because they are toxic, and a number of them can kill you. So, it's important to know about these plants when you're out foraging if you want to survive in the wilderness.

7.APPLICATION

Quantitative ethnoecological analysis of seasonal availability and implication to food security of wild edible plants (WEPs) was conducted in Boosat and Fantalle districts of semiarid east Shewa, Ethiopia from October, 2009 to September, 2010. Semistructured interview, focus group discussions, key informants discussions, seasonal record of fruits abundance were used to collect data on gathering and consumption of WEPs to cope with food shortage and adapt to climate change. Collected data was summarised into frequency tables, graph and qualitatively described under each subtopic. Thirty seven WEPs were identified for use as human food, and livestock feed and other multipurpose uses. About 24.3 % of WEPs were locally marketed, 75.7% were not marketed. All wild fruits were not included in official production system in the study area. It has indicated the underutilized existing potential of WEPs. Wild edible plants were preferred by local people of the study area not only for their food value, but also for their availability during dry seasons and shortage of food, potential for dryland agrobiodiversity and multipurpose to human wellbeing, livestock and environmental services they provide. Pairwise ranking by key informants was in agreement with direct matrices ranking for multiple uses of WEPs. The pairwise ranking, market survey and participant observations, community preference has confirmed the real potential of top seven priority WEPs species for dryland agrobiodiversity and agroforestry. Hence, these WEPs can be potential for dryland agrobiodiversity and agroforestry, to enhance people's livelihoods in semiarid areas. This result can shed light on further research and promotion work on WEPs utilization and management.

8.CONCLUSION

The research within this paper aimed to explore DL techniques to address the food limitations in 2050. Some of the challenges identified include resource scarcity and climate change that are mitigated using disease identification and plant recognition techniques. However, research shows that these techniques alone cannot fully resolve these issues, requiring consideration of alternative methods. One of these methods, explored in this paper, involves identifying wild edible plants in natural environments, providing an opportunity for scientists and researchers the ability to retrieve and examine them as a potential food source.

Using a combination of the Waterfall and Kanban methodologies, and the Jupyter Lab IDE, an artefact was created that involved transfer learning to train and tune three state-of-the-art CNN architectures, GoogLeNet, MobileNet v2, and ResNet-34, on a dataset containing 16,535 images, divided into 35 classes of wild edible plants. When tuning the architectures, seven hyperparameters, consisting of three batch sizes and four sets of hidden node sizes, were used to create 36 models evaluated on the dataset.

Further evaluation revealed that only a small number of plant classes were incorrectly classified, varying per architecture. Nevertheless, the models struggled to correctly classify two main sets of plant types, meadowsweet against cow parsley and common mallow against geranium.

Overall, determining the best model for classifying wild edible plants depends on the users' requirements.

9. FUTURE SCOPE

Wild food plants are increasingly considered a potential source of natural healthy products, it is fundamental to foster biochemical research aimed at documenting their nutritional properties and main bioactive products. The phytochemical and nutritional profiles of the species in question can therefore constitute basic knowledge for food pairing with other ingredients to improve nutritional and/or sensory quality and to find innovative cooking methods, allowing key molecules responsible for functional properties to be enhanced.

In conclusion, wild plants represent a crucial section of the human diet. It is hoped that an increasing amount of scientific research will focus on plant diversity, traditional knowledge, and agricultural studies and will foster bio-conservation strategies and sustainable food production. Biochemical knowledge is of crucial importance to evaluate the health benefits and physiological effects of WEPs in order to develop clinical investigations concerning their mechanisms of action, safety, and efficacy.

10.BIBLIOGRAPHY

- Adams, R., Hurd, B., Lenhart, S. and Leary, N. (1998) Effects of global climate change on agriculture: an interpretative review. *Climate Research*, 11(1) 19–30. Available from <https://www.intres.com/abstracts/cr/v11/n1/p19-30/> [accessed 19 January 2021].
- Affouard, A., Goeau, H., Bonnet, P., Lombardo, J-C. and Joly, A. (2017) *Pl@ntNet app in the era of deeplearning*. OpenReview. Available from <https://openreview.net/forum?id=HJVJpENFg> [accessed 22 January 2021].
- Ahmad, M.O., Markkula, J. and Oivo, M. (2013) Kanban in Software Development: A Systematic Literature Review. In: *Software Engineering and Advanced Applications (SEAA)*, Santander, Spain, 4-6 September. IEEE Computer Society, 9–16. Available from https://www.researchgate.net/publication/260739586_Kanban_in_Software_Development_A_Systematic_Literature_Review [accessed 28 January 2021].
- Al-Masri, A. (2019) *How Does Back-Propagation in Artificial Neural Networks Work?* Available from <https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7> [accessed 2 March 2021].
- Ampatzidis, Y., De, L. and Luvisi, A. (2017) iPathology: Robotic Applications and Management of Plants and Plant Diseases. *Sustainability*, 9(6) 1010. Available from <https://www.mdpi.com/2071-1050/9/6/1010> [accessed 23 January 2021].
- Anaconda (Undated) *Individual Edition - Your data science toolkit*. Anaconda. Available from <https://www.anaconda.com/products/individual> [accessed 3 February 2021].
- Aydinalp, C. and Cresser, M. (2008) The Effects of Global Climate Change on Agriculture. *American Eurasian J. Agric. Environ. Sci.*, 3. Available from https://www.researchgate.net/publication/238091112_The_Effects_of_Global_Climate_Change_on_Agriculture [accessed 19 January 2021].
- Barré, P., Stöver, B., Müller, K. and Steinhage, V. (2017) LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics*, 40 50–56. Available from <http://www.sciencedirect.com/science/article/pii/S1574954116302515> [accessed 23 January 2021].
- Bulajic, A., Sambasivam, S. and Stojic, R. (2013) An Effective Development Environment Setup for System and Application Software. *Issues in Informing Science and Information Technology*, 10 037–066. Available from <https://www.informingscience.org/Publications/1795> [accessed 2 February 2021].

