WILD PLANTS EDIBILITY PREDICTION USING

IBM WATSON STUDIO

1. INTRODUCTION

1.1 Overview

The rural communities of developing countries depend on wild edible plants to meet their food requirements during periods of food shortage. Wild edible plants are mostly serving as supplementary foods in different parts of the world because they are nutritionally rich and can supplement especially vitamins and micronutrients. The main objective of this project is to build Convolutional neural networks are a deep model to detect and classify the edibility of the wild plant. The model also suggests the effects of non-edible wild plant produce.

We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the predicted class will be displayed on the webpage.

1.2 Purpose

Many of the communities in developing countries are facing shortage of food, so they depend on wild edible plants to meet their food requirements during periods of food shortage. The main purpose of this project is to build Convolutional neural networks are a deep model to detect and classify the edibility of the wild plant.

2. LITERATURE SURVEY

2.1 Existing problem

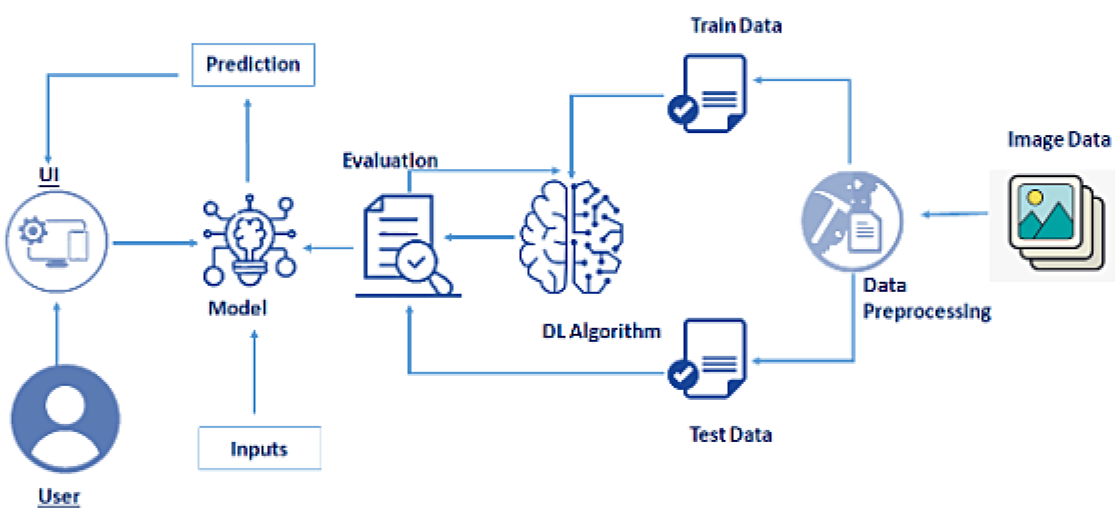
Wild food often has superior nutritional qualities, whether eaten cooked or raw. Such foraging is a great way to avoid the drawbacks of agribusiness produce, such as hybridization, genetic engineering, commercial fertilizers, pesticides & herbicides, lack of freshness, fungicides, wax, and socially transmissible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

2.2 Proposed solution

The proposed system is the effective wild edible prediction system. The rural communities of developing countries depend on wild edible plants to meet their food requirements during periods of food shortage. Wild edible plants are mostly serving as supplementary foods in different parts of the world because they are nutritionally rich and can supplement especially vitamins and micronutrients. The main objective of this project is to build Convolutional neural networks are a deep model to detect and classify the edibility of the wild plant. The model also suggests the effects of non-edible wild plant produce.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

Software Requirements

- Anaconda Navigator
- Tensorflow
- Keras
- Flask

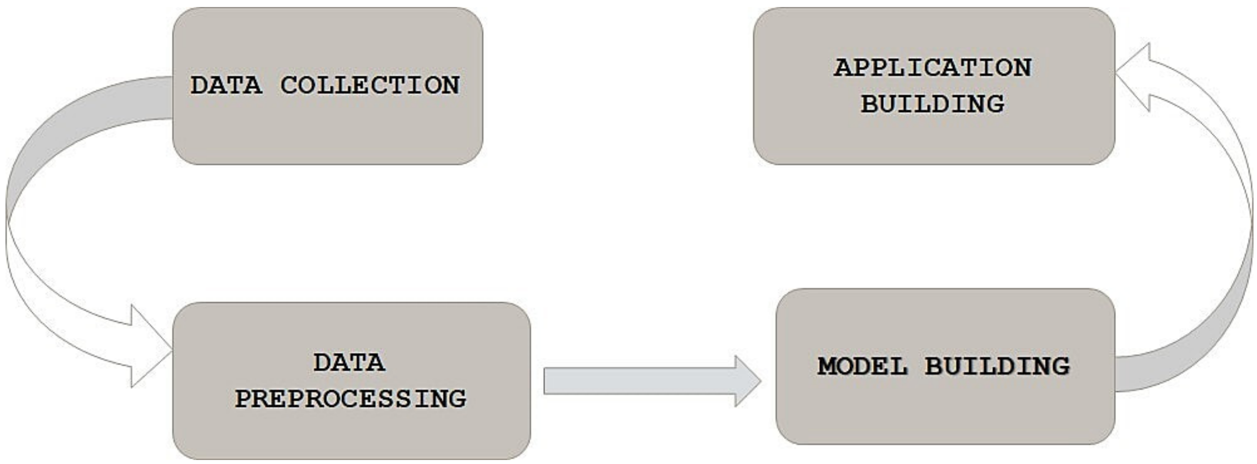
Hardware Requirements

- Processor : Intel Core i3
- Hard Disk Space : Min 100 GB
- Ram : 8 GB

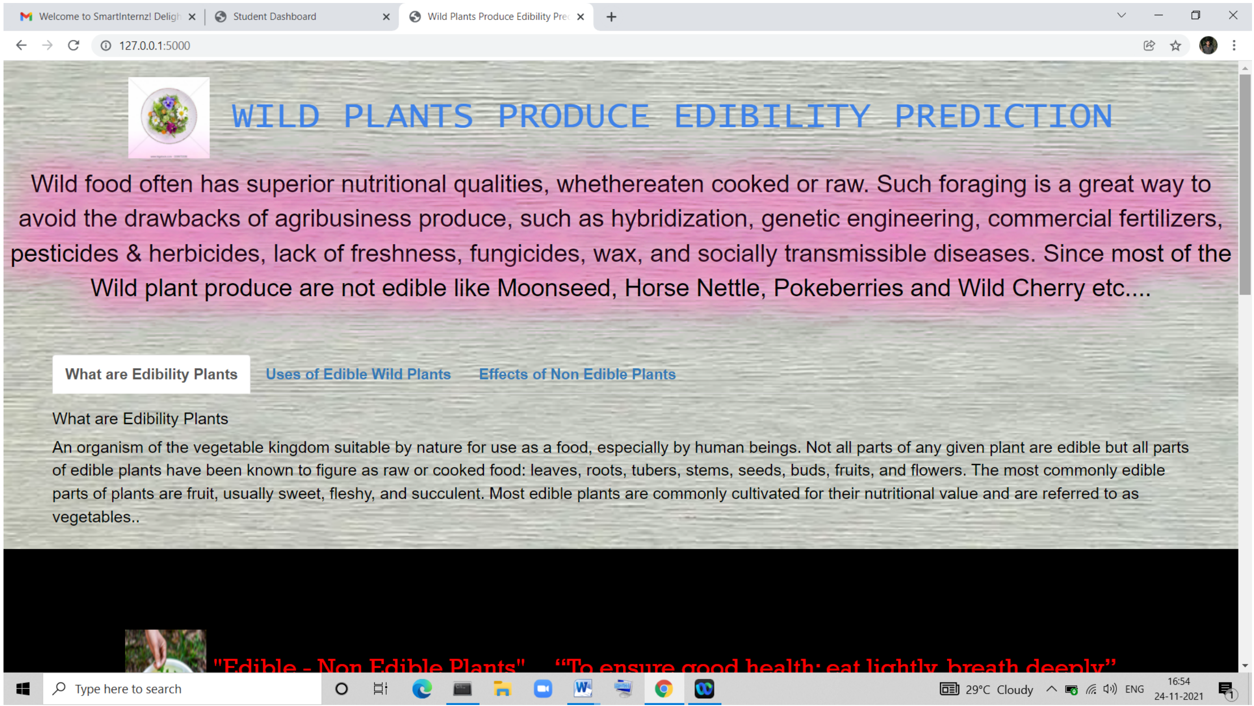
4. EXPERIMENTAL INVESTIGATIONS

Wild edible plants are mostly serving as supplementary foods in different parts of the world because they are nutritionally rich and can supplement especially vitamins and micronutrient. 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.

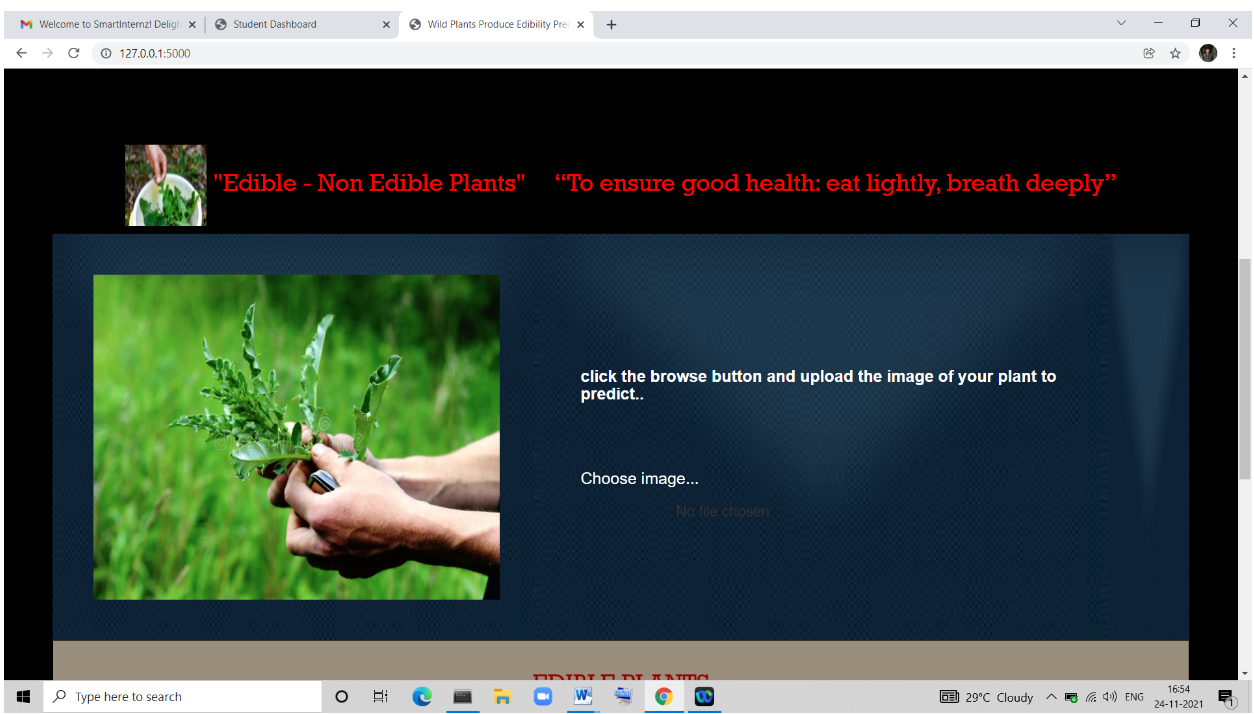
5. FLOWCHART


DATA COLLECTION
APPLICATION BUILDING
DATA PREPROCESSING
MODEL BUILDING

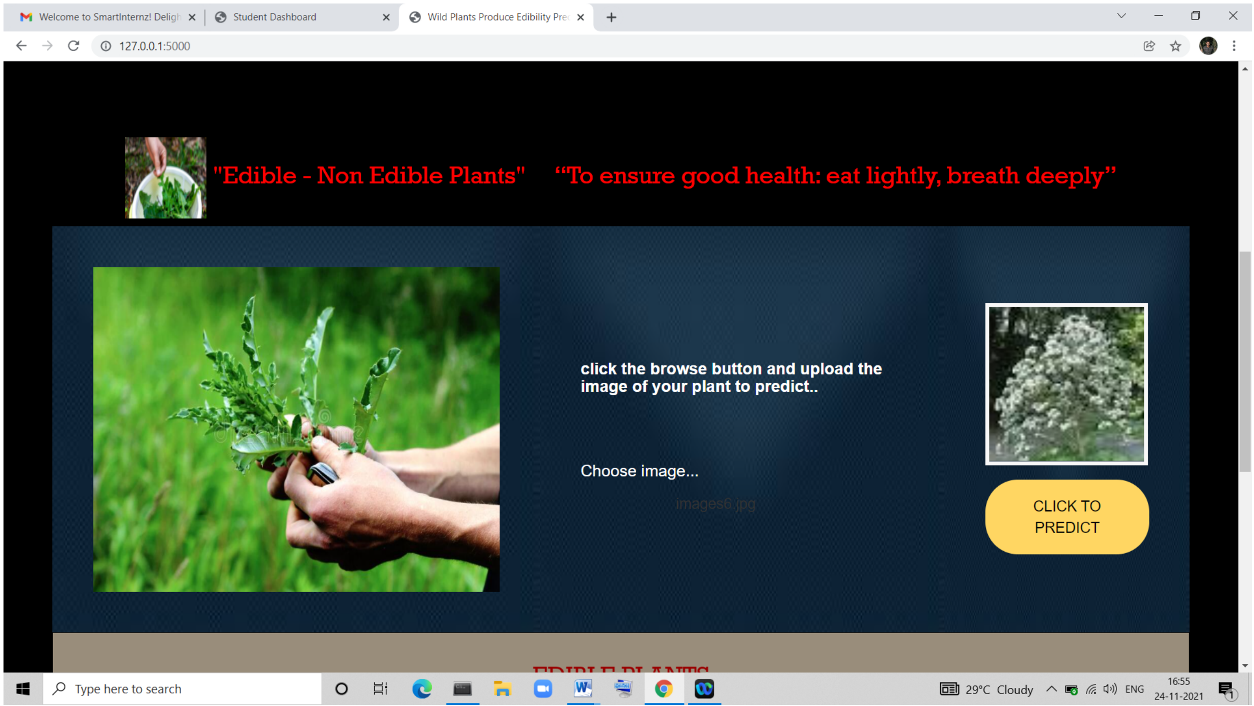
6. RESULT

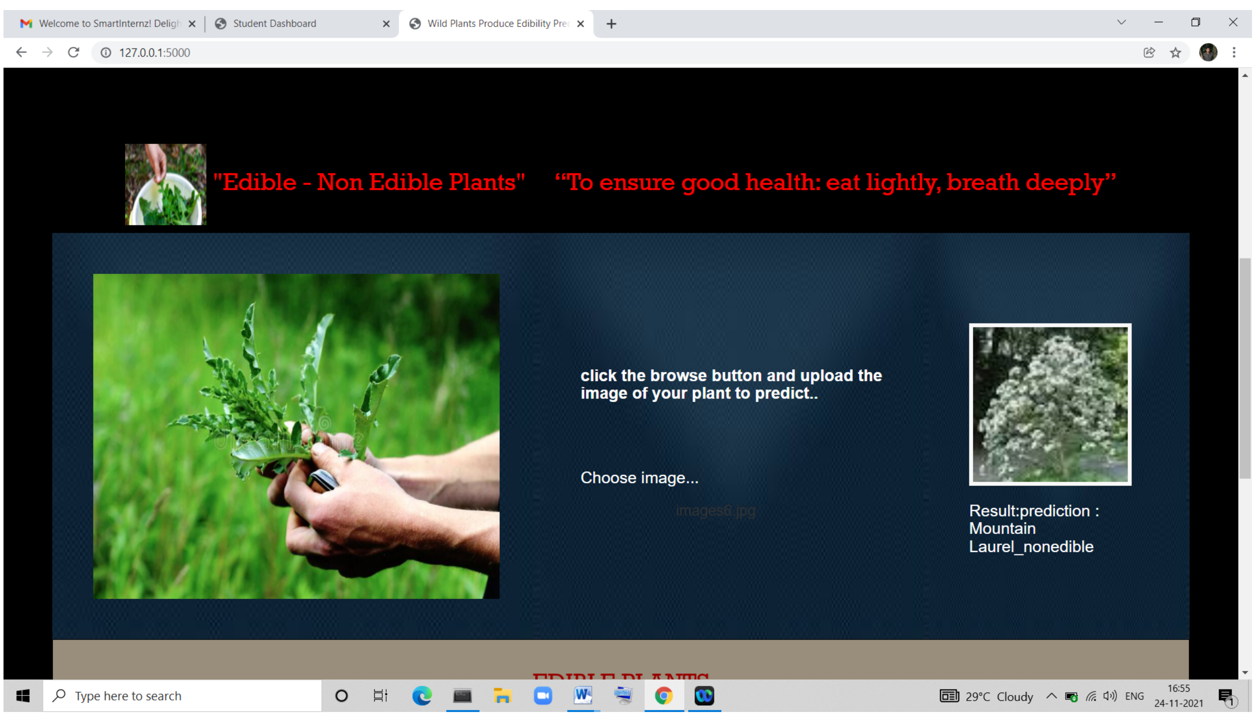


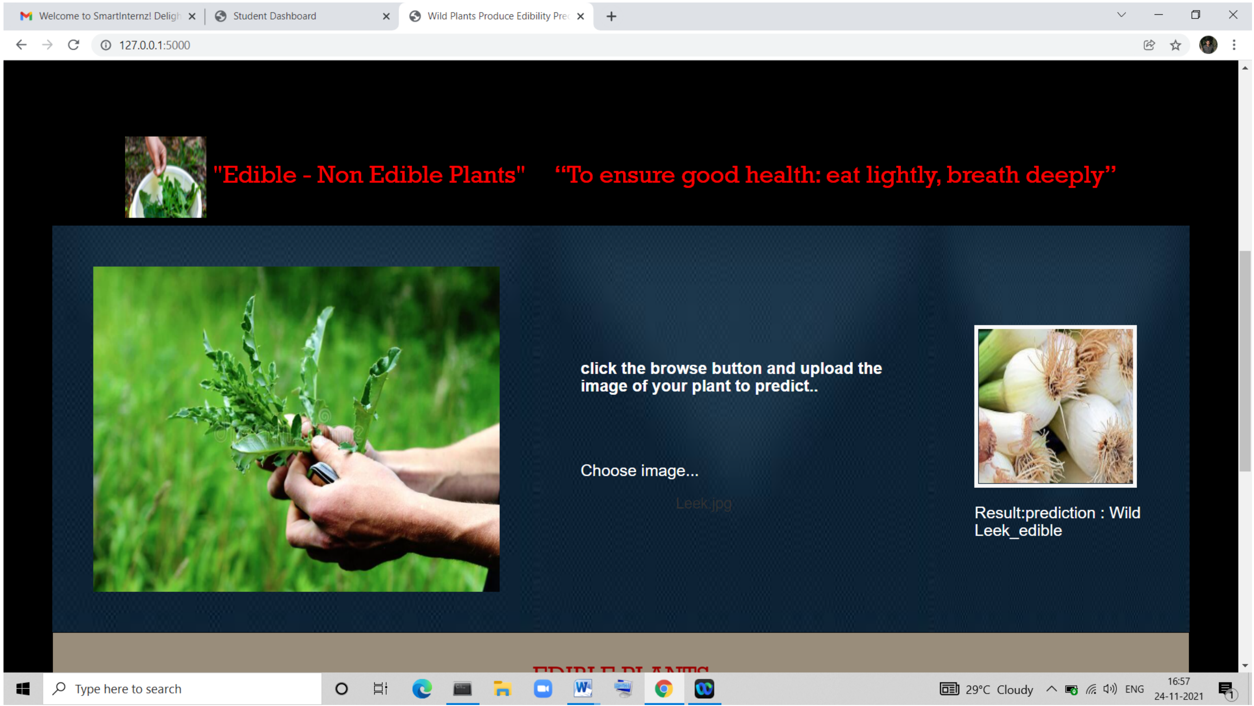
Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
WILD PLANTS PRODUCE EDIBILITY PREDICTION
Wild food often has superior nutritional qualities, whethereaten cooked or raw. Such foraging is a great way to avoid the drawbacks of agribusiness produce, such 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..
Type here to search
29°C Cloudy
ENG
16:54
24-11-2021

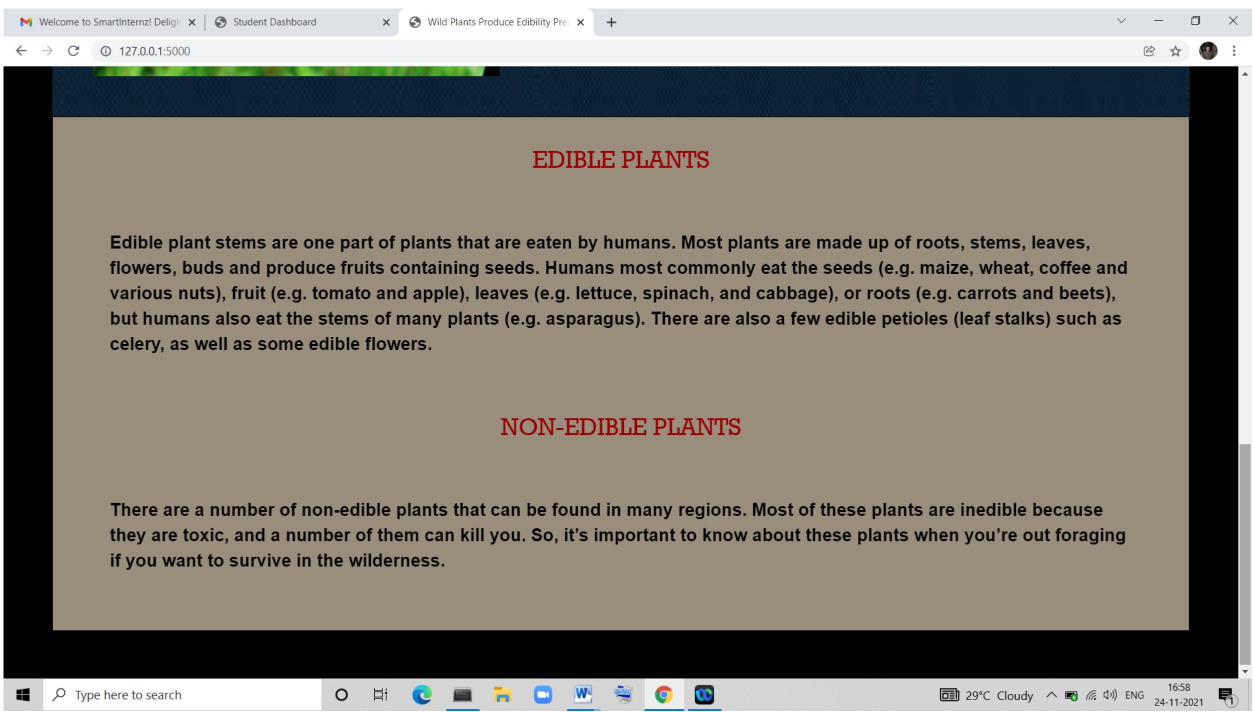


Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
"Edible - Non Edible Plants"
"To ensure good health: eat lightly, breath deeply"
click the browse button and upload the image of your plant to predict..
Choose image...
No file chosen
Type here to search
29°C Cloudy
ENG
16:54
24-11-2021


Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
"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...
images6.jpg
CLICK TO PREDICT
Type here to search
29°C Cloudy
ENG
16:55
24-11-2021


Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
"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...
images6.jpg
Result:prediction : Mountain Laurel_nonedible
Type here to search
29°C Cloudy
ENG
16:55
24-11-2021


Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
"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...
Leek.jpg
Result:prediction : Wild Leek_edible
Type here to search
29°C Cloudy
ENG
16:57
24-11-2021


Welcome to SmartInternz! Deligh
Student Dashboard
Wild Plants Produce Edibility Pre
127.0.0.1:5000
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.
Type here to search
29°C Cloudy
ENG
16:58
24-11-2021

7. ADVANTAGES & DISADVANTAGES

Advantages

Wild plants are nutritionally rich and can supplement especially vitamins and micronutrients. These show that wild edible plants are essential components of many African diets, especially in a period of seasonal food shortage.

Disadvantages

Effects of non edible plants that contain substance that may exert toxic effect on skin, effect to lung, cardiovascular system, liver, kidney, bladder, blood, nervous system, bone and the endocrine and the reproductive systems.

8. APPLICATIONS

- Better Output wild edible plants are important in efficiently using it for eating.

- web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the predicted class will be displayed on the webpage.

- Efficient Predicting features of edible and non edible images of plants can greatly helps to consume it

9. CONCLUSION

In this project, to build Convolutional neural networks are a deep model to detect and classify the edibility of the wild plant. The model also suggests the effects of non-edible wild plant produce. The rural communities of developing countries depend on wild edible plants to meet their food requirements during periods of food shortage. Wild edible plants are mostly serving as supplementary foods in different parts of the world because they are nutritionally rich and can supplement especially vitamins and micronutrients.

10. FUTURE SCOPE

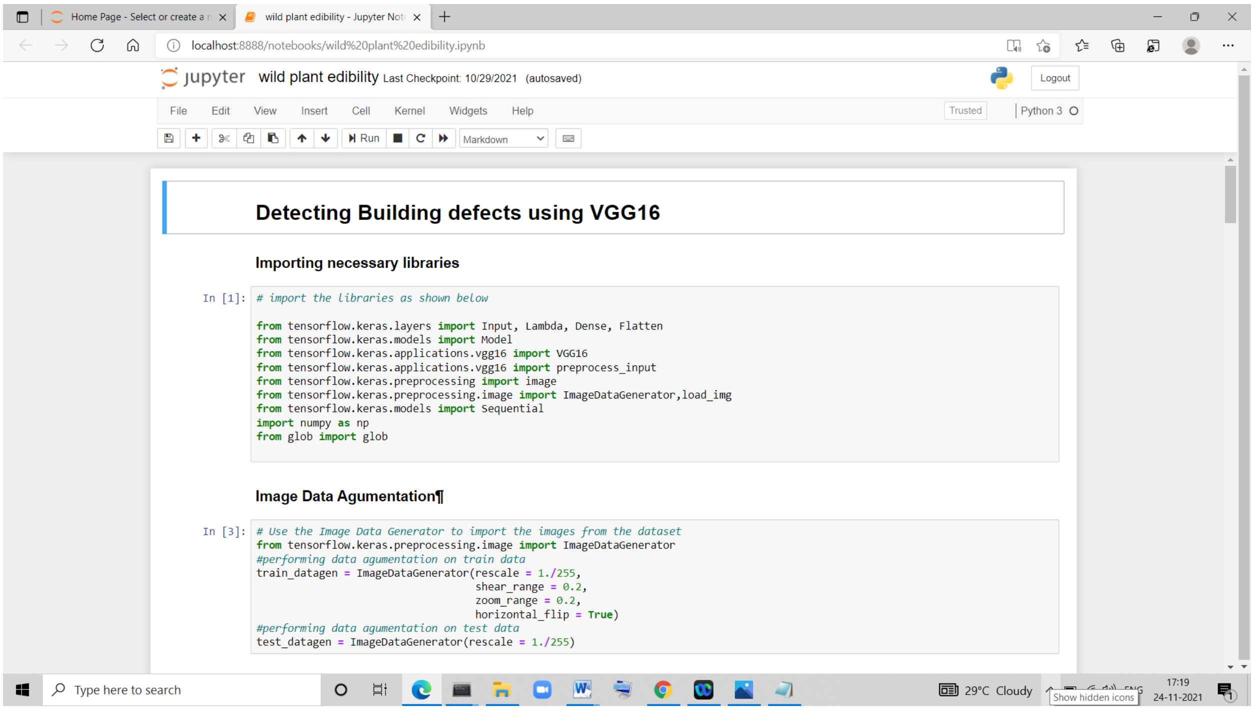
In future, our attempt would be to further improve the predictions using the system with more accuracy. Imparting more features like location, availability to our training set will enhance the predictions and will open up a new perspective on every wild edible predictions

11. BIBILOGRAPHY

- https://www.kaggle.com/wild_edible
- https://keras.io/api/preprocessing/image
- https://victorzhou.com/blog/intro-to-cnns-part-1
- https://youtu.be/kE5QZ8G_78c

APPENDIX

Source Code



Detecting Building defects using VGG16

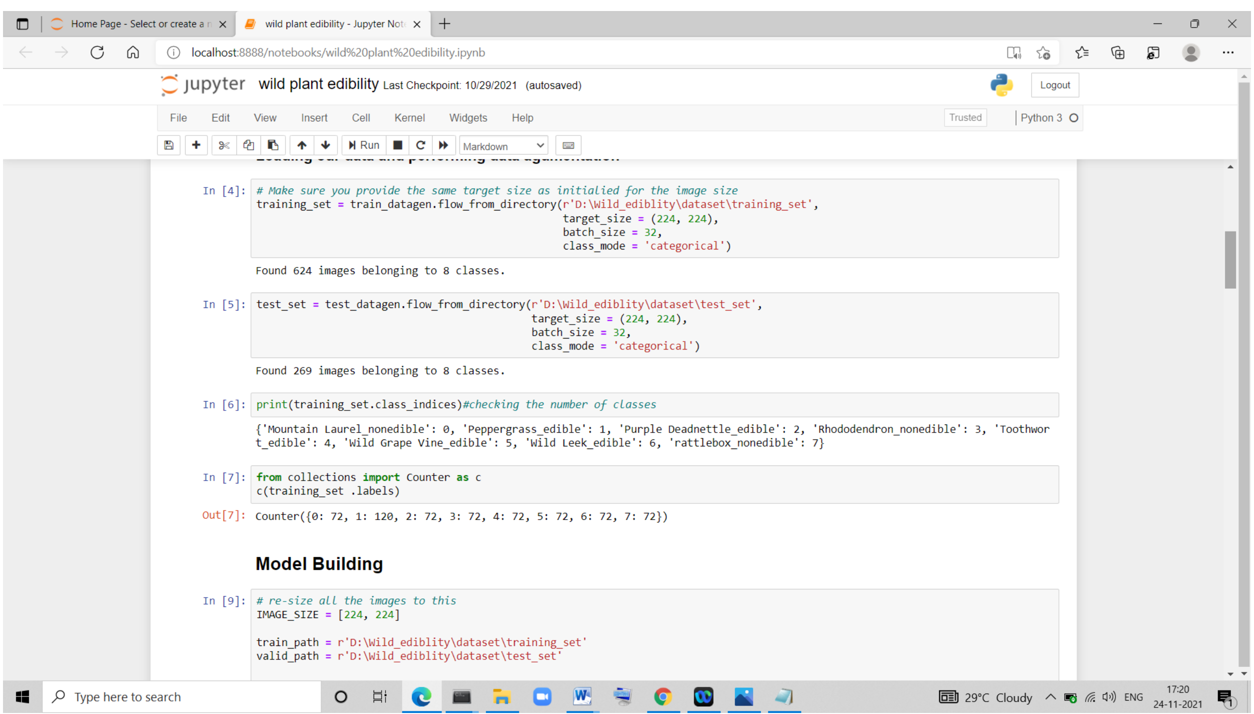
Importing necessary libraries

In [1]:
```
# import the libraries as shown below

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
```

Image Data Agumentation¶

In [3]:
```
# Use the Image Data Generator to import the images from the dataset
from tensorflow.keras.preprocessing.image import ImageDataGenerator
#performing data agumentation on train data
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
#performing data agumentation on test data
test_datagen = ImageDataGenerator(rescale = 1./255)
```



In [4]:
```
# Make sure you provide the same target size as initialied for the image size
training_set = train_datagen.flow_from_directory(r'D:\Wild_ediblity\dataset\training_set',
                                                 target_size = (224, 224),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')
```

Found 624 images belonging to 8 classes.

In [5]:
```
test_set = test_datagen.flow_from_directory(r'D:\Wild_ediblity\dataset\test_set',
                                            target_size = (224, 224),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

Found 269 images belonging to 8 classes.

In [6]:
```
print(training_set.class_indices)#checking the number of classes
```

{'Mountain Laurel_nonedible': 0, 'Peppergrass_edible': 1, 'Purple Deadnettle_edible': 2, 'Rhododendron_nonedible': 3, 'Toothwort_edible': 4, 'Wild Grape Vine_edible': 5, 'Wild Leek_edible': 6, 'rattlebox_nonedible': 7}

In [7]:
```
from collections import Counter as c
c(training_set .labels)
```

Out[7]: Counter({0: 72, 1: 120, 2: 72, 3: 72, 4: 72, 5: 72, 6: 72, 7: 72})

Model Building

In [9]:
```
# re-size all the images to this
IMAGE_SIZE = [224, 224]

train_path = r'D:\Wild_ediblity\dataset\training_set'
valid_path = r'D:\Wild_ediblity\dataset\test_set'
```

jupyter wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

In [10]:
```
# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG
# Here we will be using imagenet weights

vgg16 = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 117s 2us/step
```

In [11]:
```
# don't train existing weights
for layer in vgg16.layers:
    layer.trainable = False
```

In [12]:
```
  # useful for getting number of output classes
folders = glob(r'D:\Wild_ediblity\dataset\training_set\*')
```

In [13]:
```
folders
```

Out[13]:
```
['D:\\Wild_ediblity\\dataset\\training_set\\Mountain Laurel_nonedible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Peppergrass_edible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Purple Deadnettle_edible',
 'D:\\Wild_ediblity\\dataset\\training_set\\rattlebox_nonedible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Rhododendron_nonedible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Toothwort_edible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Wild Grape Vine_edible',
 'D:\\Wild_ediblity\\dataset\\training_set\\Wild Leek_edible']
```

In [14]:
```
# our layers - you can add more if you want
x = Flatten()(vgg16.output)
```

In [15]:
```
prediction = Dense(len(folders), activation='softmax')(x)

# create a model object
model = Model(inputs=vgg16.input, outputs=prediction)
```

jupyter wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

In [16]:
```
# view the structure of the model
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
```

jupyter wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

Compiling the model

In [17]:
```
# tell the model what cost and optimization method to use
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

Fit the model

In [18]:
```
# fit the model
# Run the cell. It will take some time to execute
r = model.fit_generator(
  training_set,
  validation_data=test_set,
  epochs=10,
  steps_per_epoch=len(training_set),
  validation_steps=len(test_set)
)
```

```
C:\Users\HABIN\Anaconda3\lib\site-packages\tensorflow\python\keras\engine\training.py:1940: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

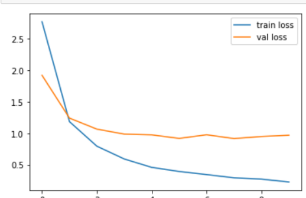
```
Epoch 1/10
20/20 [==============================] - 157s 8s/step - loss: 2.7713 - accuracy: 0.2564 - val_loss: 1.9207 - val_accuracy: 0.35
69
Epoch 2/10
20/20 [==============================] - 161s 8s/step - loss: 1.1845 - accuracy: 0.6266 - val_loss: 1.2406 - val_accuracy: 0.57
25
Epoch 3/10
20/20 [==============================] - 155s 8s/step - loss: 0.7938 - accuracy: 0.7468 - val_loss: 1.0654 - val_accuracy: 0.62
```

jupyter wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

```
Epoch 9/10
20/20 [==============================] - 155s 8s/step - loss: 0.2707 - accuracy: 0.9343 - val_loss: 0.9484 - val_accuracy: 0.66
54
Epoch 10/10
20/20 [==============================] - 155s 8s/step - loss: 0.2250 - accuracy: 0.9535 - val_loss: 0.9686 - val_accuracy: 0.63
57
```

In [30]:
```
# plot the loss
import matplotlib.pyplot as plt
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')

# plot the accuracy
plt.plot(r.history['acc'], label='train acc')
plt.plot(r.history['val_acc'], label='val acc')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')
```



wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

Saving our model

In [22]:
```
# save it as a h5 file

from tensorflow.keras.models import load_model

model.save('WildPlantEdibility.h5')
```

Predicitng our results

In [36]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("WildPlantEdibility.h5") #loading the model for testing
```

In [37]:
```
img=image.load_img(r'D:\Wild_edibliity\dataset\test_set\rattlebox_nonedible\images116.jpg',target_size=(224,224))
x=image.img_to_array(img)
#x=x/255C:\Users\HP\Desktop\Wild_edibliity\dataset\training_set
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
#model.predict(img_data)
a=np.argmax(model.predict(img_data), axis=1)
```

In [38]:
```
a
```

Out[38]:
```
array([7], dtype=int64)
```

In [39]:
```
index=['Mountain Laurel_nonedible', 'Peppergrass_edible', 'Purple Deadnettle_edible', 'Rhododendron_nonedible', 'Toothwort_edibl
result=str(index[a[0]])
result
```

Out[39]:
```
'rattlebox_nonedible'
```

wild plant edibility Last Checkpoint: 10/29/2021 (autosaved)

In [36]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("WildPlantEdibility.h5") #loading the model for testing
```

In [37]:
```
img=image.load_img(r'D:\Wild_edibliity\dataset\test_set\rattlebox_nonedible\images116.jpg',target_size=(224,224))
x=image.img_to_array(img)
#x=x/255C:\Users\HP\Desktop\Wild_edibliity\dataset\training_set
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
#model.predict(img_data)
a=np.argmax(model.predict(img_data), axis=1)
```

In [38]:
```
a
```

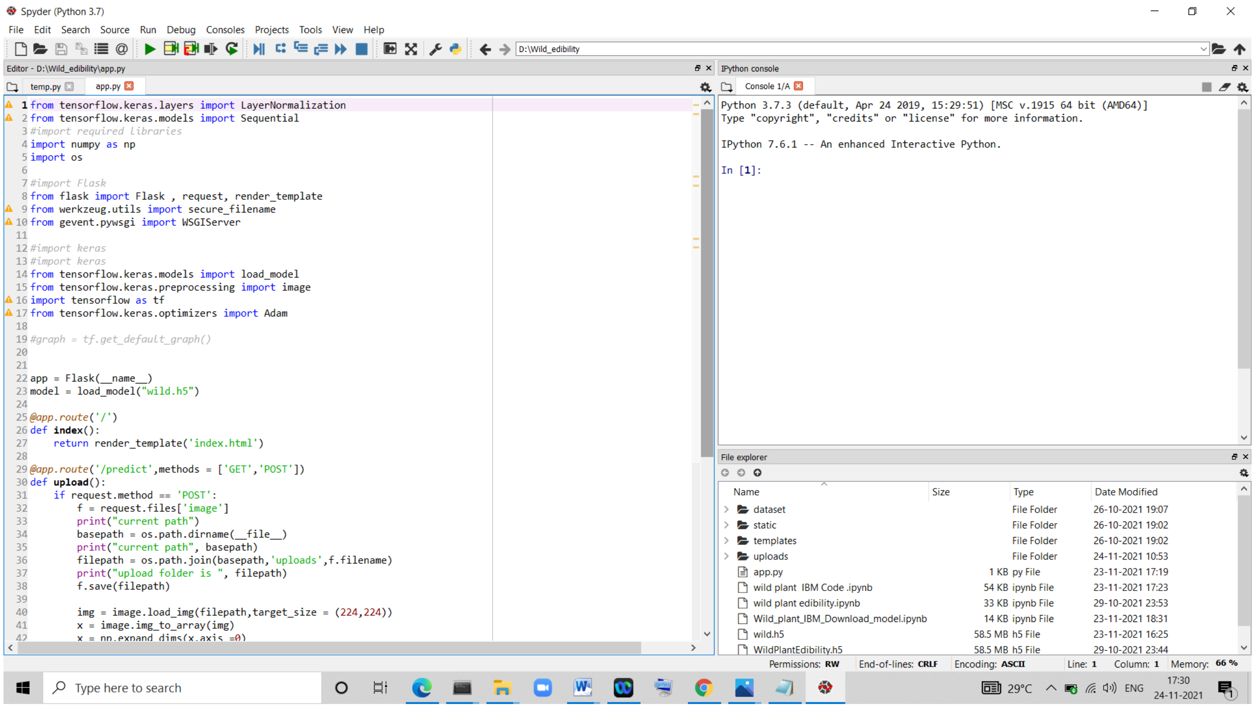
Out[38]:
```
array([7], dtype=int64)
```

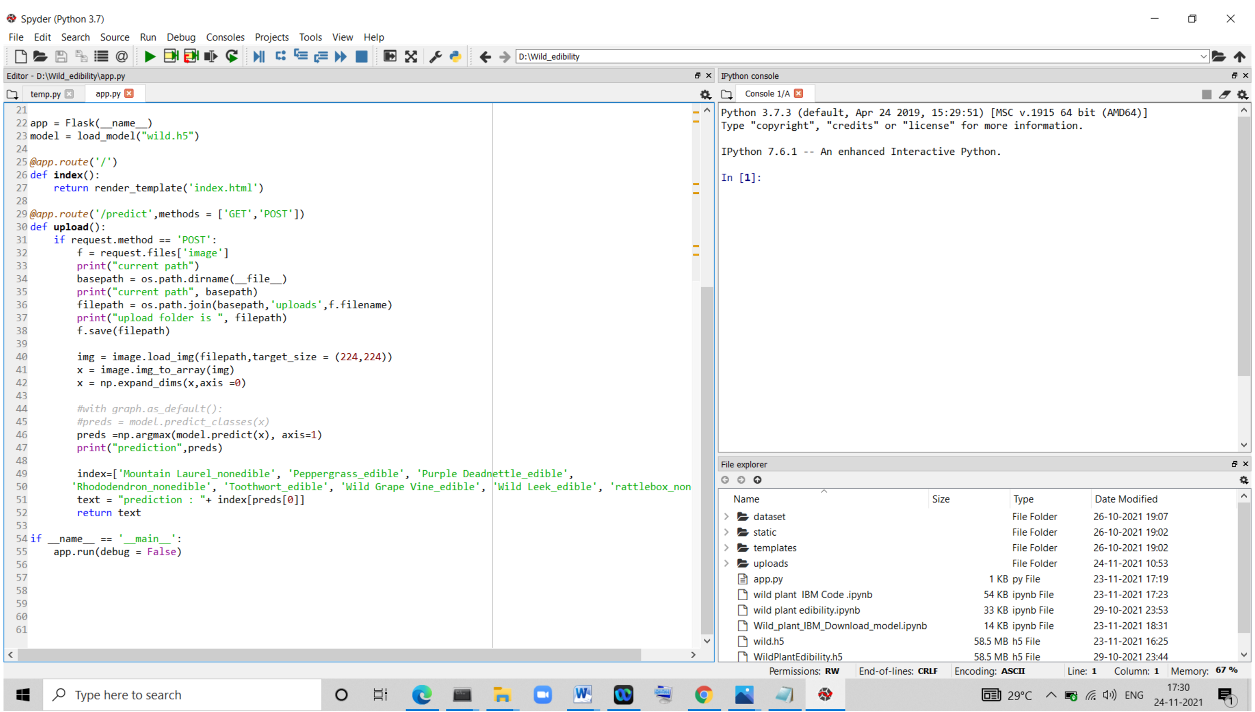
In [39]:
```
index=['Mountain Laurel_nonedible', 'Peppergrass_edible', 'Purple Deadnettle_edible', 'Rhododendron_nonedible', 'Toothwort_edibl
result=str(index[a[0]])
result
```

Out[39]:
```
'rattlebox_nonedible'
```


In [ ]:

In [ ]:


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Wild_edibility
Editor - D:\Wild_edibility\app.py
temp.py
app.py
from tensorflow.keras.layers import LayerNormalization
from tensorflow.keras.models import Sequential
#import required libraries
import numpy as np
import os
#import Flask
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
#import keras
#import keras
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import tensorflow as tf
from tensorflow.keras.optimizers import Adam
#graph = tf.get_default_graph()
app = Flask(__name__)
model = load_model("wild.h5")
@app.route('/')
def index():
return render_template('index.html')
@app.route('/predict',methods = ['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['image']
print("current path")
basepath = os.path.dirname(__file__)
print("current path", basepath)
filepath = os.path.join(basepath,'uploads',f.filename)
print("upload folder is ", filepath)
f.save(filepath)
img = image.load_img(filepath,target_size = (224,224))
x = image.img_to_array(img)
IPython console
Console 1/A
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.6.1 -- An enhanced Interactive Python.
In [1]:
File explorer
Name Size Type Date Modified
dataset File Folder 26-10-2021 19:07
static File Folder 26-10-2021 19:02
templates File Folder 26-10-2021 19:02
uploads File Folder 24-11-2021 10:53
app.py 1 KB py File 23-11-2021 17:19
wild plant IBM Code .ipynb 54 KB ipynb File 23-11-2021 17:23
wild plant edibility.ipynb 33 KB ipynb File 29-10-2021 23:53
Wild_plant_IBM_Download_model.ipynb 14 KB ipynb File 23-11-2021 18:31
wild.h5 58.5 MB h5 File 23-11-2021 16:25
WildPlantEdibility.h5 58.5 MB h5 File 29-10-2021 23:44
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 1 Column: 1 Memory: 66 %
Type here to search
29°C ENG 17:30 24-11-2021


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Wild_edibility
Editor - D:\Wild_edibility\app.py
temp.py
app.py
app = Flask(__name__)
model = load_model("wild.h5")
@app.route('/')
def index():
return render_template('index.html')
@app.route('/predict',methods = ['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['image']
print("current path")
basepath = os.path.dirname(__file__)
print("current path", basepath)
filepath = os.path.join(basepath,'uploads',f.filename)
print("upload folder is ", filepath)
f.save(filepath)
img = image.load_img(filepath,target_size = (224,224))
x = image.img_to_array(img)
x = np.expand_dims(x,axis =0)
#with graph.as_default():
#preds = model.predict_classes(x)
preds =np.argmax(model.predict(x), axis=1)
print("prediction",preds)
index=['Mountain Laurel_nonedible', 'Peppergrass_edible', 'Purple Deadnettle_edible',
'Rhododendron_nonedible', 'Toothwort_edible', 'Wild Grape Vine_edible', 'Wild Leek_edible', 'rattlebox_non
text = "prediction : "+ index[preds[0]]
return text
if __name__ == '__main__':
app.run(debug = False)
IPython console
Console 1/A
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.6.1 -- An enhanced Interactive Python.
In [1]:
File explorer
Name Size Type Date Modified
dataset File Folder 26-10-2021 19:07
static File Folder 26-10-2021 19:02
templates File Folder 26-10-2021 19:02
uploads File Folder 24-11-2021 10:53
app.py 1 KB py File 23-11-2021 17:19
wild plant IBM Code .ipynb 54 KB ipynb File 23-11-2021 17:23
wild plant edibility.ipynb 33 KB ipynb File 29-10-2021 23:53
Wild_plant_IBM_Download_model.ipynb 14 KB ipynb File 23-11-2021 18:31
wild.h5 58.5 MB h5 File 23-11-2021 16:25
WildPlantEdibility.h5 58.5 MB h5 File 29-10-2021 23:44
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 1 Column: 1 Memory: 67 %
Type here to search
29°C ENG 17:30 24-11-2021