Plant Seedling Classification

1. INTRODUCTION:

The agricultural sector has recognized that for crop management to thrive, acquiring relevant information on plants is needed. However, studies have shown that agricultural problems remain difficult in many parts of the world due to the lack of the necessary infrastructures. Using a dataset of plant images that consist of descriptions under a controlled condition concerning camera radiance and stabilization. This paper uses a convolutional neural network for training and does data augmentation to identify 3 plant species. The trained model achieved an accuracy categorization of 98.07%. In future works, we plan to utilize the model by training it to other types of plants like herbalmedicinal plants and other crops in other countries. Moreover, the proposed method can be integrated into a mobile application with the goal to proviide farmers efficient farming practices.

1.1 Overview:

An essential aspect of life on earth is the presence of plants. In fact, it gives us sustenance, medicines, protect, and provide us with a sound breathable climate. However, with a bloating population, observed climate change and environmental changes, there is a growing risk to several ecosystems. Identification of plant seedling is essential as it can help in recognizing different plant species. In most cases, identification is performed manually, either visually or by microscopy. Recently, manual identification of plant species made it difficult for human experts, and more so for amateurs as it may take several minutes, hours or even days to solve expert tasks. A good understanding of shapes, petal and leaf types, as well as the entire plants, are equally essential to help measure plant condition and even model climate change using machine learning algorithms. Consequently, it is vital to categorize new or unusual plant classes. Furthermore, the interests in image classification are growing. Given the availability of digital images from online databases, the need to create a fast method to classify plant species is urgent.

Recently, classical techniques have been extensively used for identification and classification of plant images. However, these conventional techniques have been replaced by Deep Learning algorithms. In DL, handcrafted feature extractors are unneeded. Previously, traditional methods of sorting images involved the use of hand-crafted approach. Significant works have been done on speech, natural language and image processing. Opposing to outmoded machine learning procedures where features are manually selected and dig out through initiated processes, a class of deep learning technique called Convolutional Neural Networks (CNN) can quickly determine gradually sophisticated features from the available data. To this end, CNN becomes widespread and capable of realizing unique solutions to various concerns, particularly plant classification within agricultural fields. Currently, there have been a lot of researchers develop automated analysis to plant images. A review of the different techniques used has also been presented recently.

As a result, they turned out to become feasible for more multifaceted image recognition issues. Currently, CNN's are utilized in various existing modernistic image sorting tasks which can tackle exceedingly intricate image recognition with several entity classes to a remarkable gauge. In this article, we intend to put forward deep learning-based convolutional neural systems for plant seedling classification of images. The plant classification model is tested to plant seedling images then evaluated using accuracy, specificity, and sensitivity. The results of this study will be an excellent contribution to the continuous development of crops management and towards providing a new technique to assist and ease farming practices.

1.2 Purpose:

Our aim from the project is to make use of tensorflow, scikit, & flask libraries from python to extract the libraries for machine learning for the plant seedling classification. Then, the inputs are pre-processed images of 3 plant species and the output are its classification or the types of plant normal we have used techniques of convolutional neural network algorithms and withdrawing the conclusions.

2. LITERATURE SURVEY:

CNNs are one of the recent advance techniques which are able to perform better feature extraction in an efficient way but it is mostly suitable for the sequential data. Recently, many data scientists have proved that using CNNs in Deep Learning will improve the performance of the algorithms and theses scientists have used energy physics for the particle collision analysis in energy physics which has shown great results. Therefore, CNNs have proved very efficient in classification tasks used in Deep Learning.

3.THEORETICAL ANALYSIS

When it comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used in various classification tasks like image, audio, words. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use **Convolution Neural Network**.

To achieve our goal, we have used one of the famous machine learning algorithms out there which is used for Image Classification i.e. Convolutional Neural Network(or CNN).As we know its a machine learning algorithm for machines to understand the features of the image with foresight and remember the features to guess whether the name of the new image fed to the machine. At first we created our very own dataset which includes the plants images. Now after getting the data set, we pre-process the data a bit and provide labels to each of the images provided.

Libraries used:

- DataGenerator- The ImageDataGenerator is an easy way to load and augment images in batches for image classification tasks.
- tensorflow – To add layers as well as compare the loss and adam curve our result data or obtained log.

Layers used to build ConvNets

A convnets is a sequence of layers, and every layer transforms one volume to another through differentiable functions.

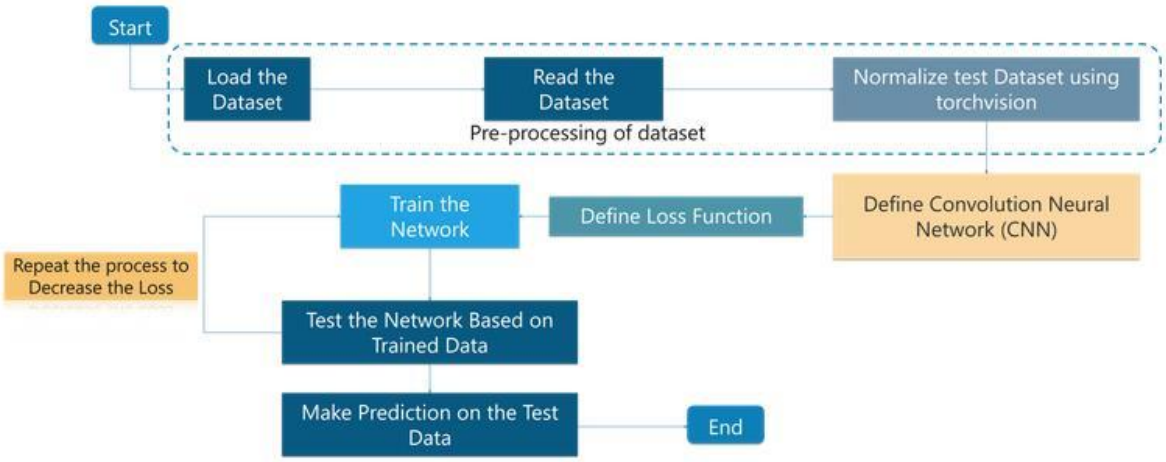
Types of layers:

1. **Input Layer**: This layer holds the raw input of the image.
2. **Convolution Layer:** This layer computes the output volume by computing dot product between all filters and image patches.
3. **Activation Function Layer**: This layer will apply element wise activation function to the output of the convolution layer. Some common activation functions are RELU, Sigmoid, Tanh, Leaky RELU, etc. We have used RELU as well as softmax for our model as this is a multiclassification problem.
4. **Pool Layer:** This layer is periodically inserted in the convNets and its main function is to reduce the size of volume which makes the computation fast, reduces memory and also prevents from overfitting. Two common types of pooling layers are max pooling and average pooling.
5. **Dense layer** - It is the regular deeply connected neural network **layer**

   Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set.

   We have used Non-Linearity (ReLU) activation function. ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0, x)$. ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. We have used the softmax function as this is a multi class classification problem.

3.1 Flow Chart:



3.2 Software Designing

● Jupyter Notebook Environment

● Spyder Ide

●Deep Learning Algorithm (CNN)

● Python

● HTML

● Flask

We developed the Plant Seedling Classification by using the Python language which is an interpreted and highlevel programming language and using the Deep Learning algorithms. For coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For pneumonia prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a web page is HTML by creating the templates to use in the functions of the Flask and HTML.

4. Screenshot:

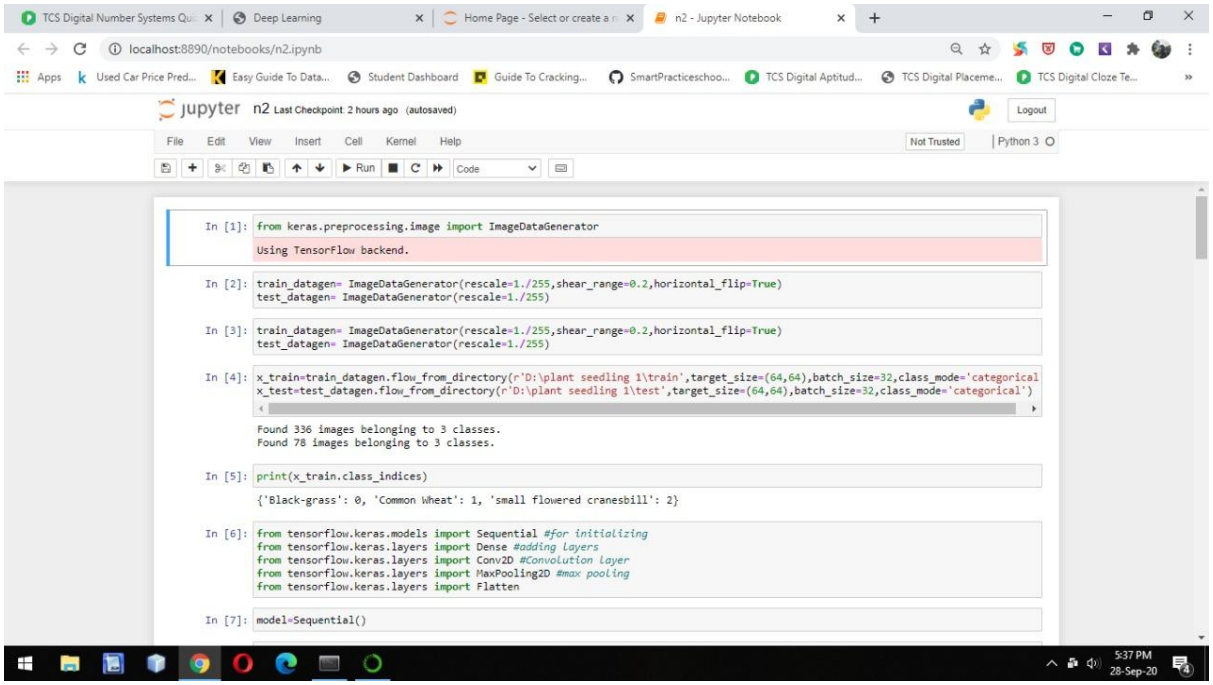


Fig1: Importing the Libraries and Dataset

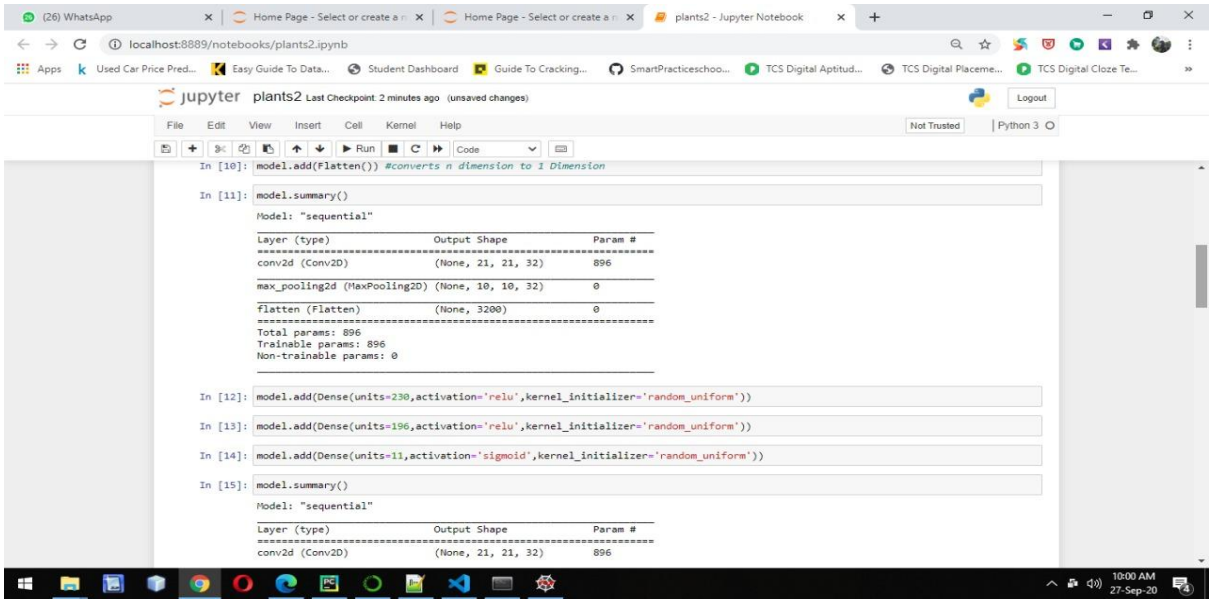


Fig 2: Represents the model.summary()

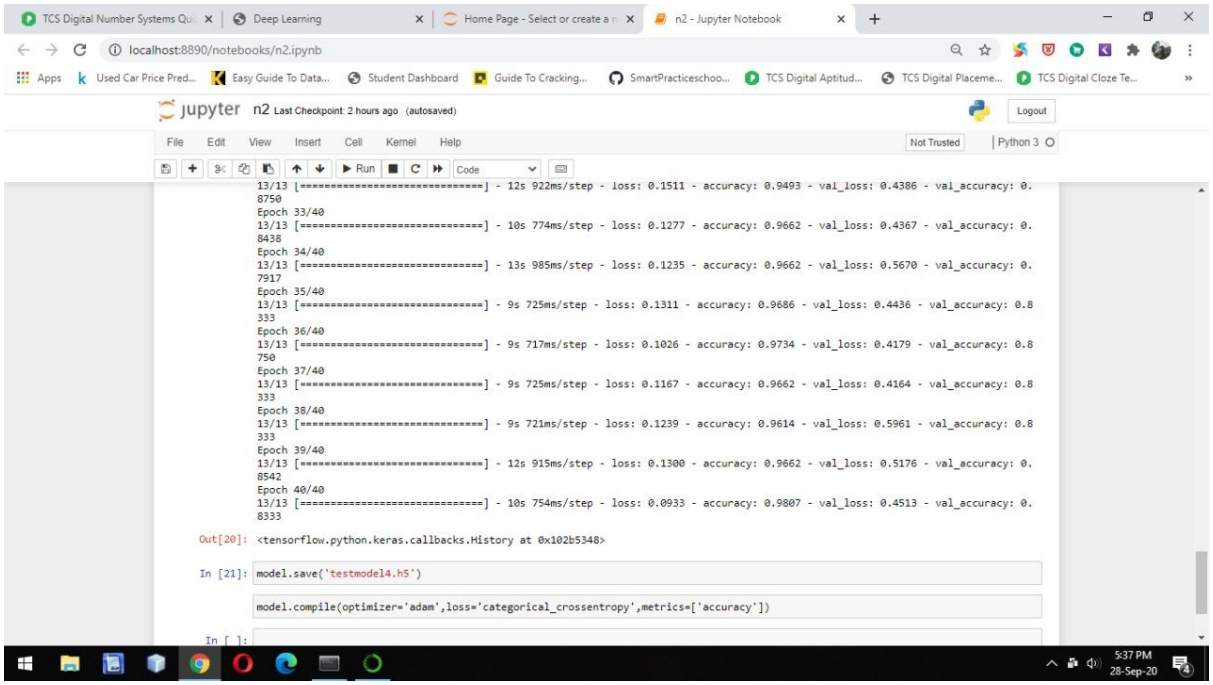


Fig3: Represents model fitting

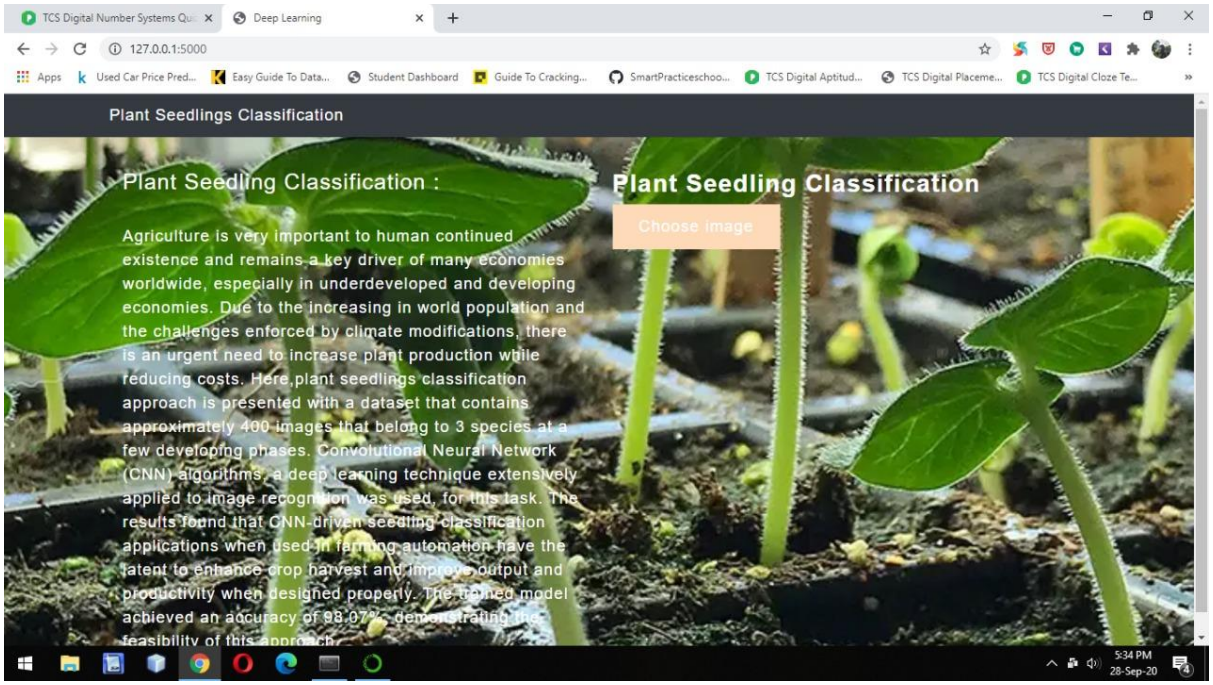


Fig4: Represents Output Screen

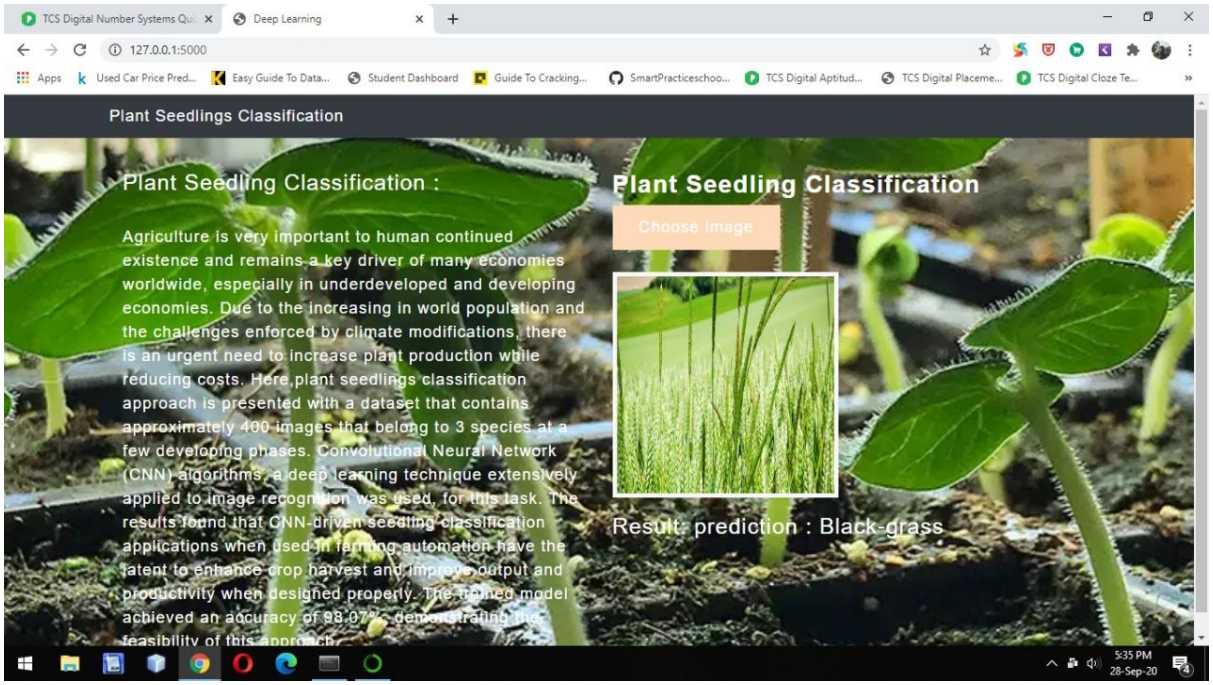


Fig5: Represents the final web page done with flask application

7. APPLICATIONS:

- CNNs are now-a-days widely used in the computer vision and automation fields. This helps in developing such artificial systems which has capability of performing complex tasks with efficiency.
- CNNs are also being used in the domain of natural language processing for language analysis, language modeling, language designing. CNN models helps in determining the various semantics of any sentence for knowing the better about the client's requirements.
- CNNs are being used for object detection purpose for identifying the objects in the way. Segmentation of images is also being done using the CNNs.
- Image Classification is one of the very important task which is done using the CNNs in the present scenario by various data augmentation techniques and feature extraction techniques.
- One of the most important applications is the speech recognition in which the speech is being recognized using some automated devices. For example, Google's speech recorder.
- CNNs are also widely used for the data which are computationally very limited in resources. There are several techniques which are still being working on small datasets with improved accuracy of classification.

- CNNs are also being used for the images which are having low resolution. Many researchers have given different techniques to work on the images having low resolution using CNN.

8. ADVANTAGES AND DISADVANTAGES OF CNN MODEL

Advantages:

- Except from the improvements in precision observed in the classification/prediction problems at the surveyed works, there are some other important advantages of using CNN in image processing. Previously, traditional approaches for image classification tasks were based on hand-engineered features, whose performance and accuracy greatly affected the overall results. Feature engineering (FE) is a complex, time-consuming process which needs to be altered whenever the problem or the data set changes. Thus, FE constitutes an expensive effort that depends on experts' knowledge and does not generalize well.

- Convolutional neural networks seem to generalize well and they are quite robust even under challenging conditions such as illumination, complex background, size and orientation of the images, and different resolution. It helps farmers to identify which animals are causing harm to their crops.so They can take preventive measures.

Disadvantages:

- The main disadvantage is that CNN can sometimes take much longer to train. However, after training, their testing time efficiency is much faster than other methods

- Other disadvantages include problems that might occur when using pre-trained models on similar and smaller data sets, optimization issues because of the models' complexity, as well as hardware restrictions.

9. CONCLUSION:

In this paper, a new method for categorizing plant species at early growth stages was explored using deep learning-based convolutional neural networks. The developed model demonstrated a validation accuracy of 98.07% for testing, outperforming conventional approaches. This result is an excellent contribution to the continuous development in the agricultural research area and to the widespread objective of augmenting global agricultural yield. However, some limitations should be noted. The first is that the performance of deep learning algorithm is dependent on large-scale datasets, containing millions of training samples. This means the smaller the training samples, the higher the possibility of overfitting throughout the training stage. To lessen the likelihood of overfitting, there's a need to increase the training data into hundreds of thousands or even millions of training data through augmentation. Other techniques such as the implementation of dropout can also help prevent overfitting. The second is that CNN requires high hardware demands and requires a massive amount of memory for training the model. Furthermore, the model is focused on considering ten plant species. We can extend this model to other types of plants such as herbal-medicinal plants or other crops in other countries as well as considering other datasets from online databases. Finally, the proposed technique can be integrated into a mobile application with the goal to provide farmers efficient farming practices.

10. Bibliography:

- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Aurelien Geron.
- Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get M for free. arXiv preprint arXiv:1704.00109.
- Guillaume Lemaitre, Fernando Nogueira & Christos K. Aridas (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research.
- J. Chaki, R. Parekh, and S. Bhattacharya, “Plant leaf classification using multiple descriptors: A hierarchical approach,” J. King Saud Univ. - Comput. Inf. Sci., pp. 1–15, 2018.
- R. M. Prakash, “Detection of Leaf Diseases and Classification using Digital Image Processing,” 2017.
- A. Kamilaris and F. X. Prenafeta-boldú, “Deep learning in agriculture : A survey,” vol. 147, no. July 2017, pp. 70–90, 2018.
- M. Mehdipour, B. Yanikoglu, and E. Aptoula, “Neurocomputing Plant identi fi cation using deep neural networks via optimization of transfer learning parameters,” Neurocomputing, no. April 2016, pp. 0–1, 2017.

Appendix:

HTML:

Base HTML:

Index HTML:

App.py:

```
from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import numpy as np
from keras.preprocessing import image


from keras.applications.imagenet_utils import preprocess_input, decode_predictions

from keras.models import load_model
from keras import backend
from tensorflow.keras import backend

import tensorflow as tf

global graph
graph=tf.get_default_graph()

#global graph
#graph = tf.get_default_graph()
```

```
from skimage.transform import resize

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
MODEL_PATH = 'models/testmodel4.h5'

# Load your trained model
model = load_model(MODEL_PATH)
      # Necessary
# print('Model loaded. Start serving...')

# You can also use pretrained model from Keras
# Check https://keras.io/applications/
#from keras.applications.resnet50 import ResNet50
#model = ResNet50(weights='imagenet')
#model.save('')
print('Model loaded. Check http://127.0.0.1:5000/')




@app.route('/', methods=['GET'])
```

```
def index():
    # Main page
    return render_template('index.html')



@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']


        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(64, 64))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)


        with graph.as_default():
            preds = model.predict_classes(x)
        index = ['Black-grass','Charlock','Cleavers','Common Chickweed','Common
wheat','Fat Hen','Loose Silky-bent','Maize','Scentless Mayweed','Shepherds
Purse','Small-flowered Cranesbill','Sugar beet']
        text = "prediction : "+index[preds[0]]


            # ImageNet Decode
```

```
        return text


if __name__ == '__main__':
    app.run(debug=False,threaded = False)
```