Assignment No: - DL-Group1-3

TITLE	Plant Disease analysis and detection using Convolutional neural network (CNN)
PROBLEM STATEMENT /DEFINITION	**Convolutional neural network (CNN)** (Any One from the following) • Use any dataset of plant disease and design a plant disease detection system using CNN. • Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories (Analyze and detect the plant diseases using CNN Model by training the data set.)
OBJECTIVE	To build a CNN model to detect Plant diseases .
OUTCOME	To understand the exploratory data analysis,split the training and testing data, Model Evaluation and Prediction by the CNN on the Plant disease detection data set
S/W PACKAGES AND HARDWARE/ APPARATUS USED	Jupyter notebook IDE, python3 PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
REFERENCES	https://www.kaggle.com/code/deveshkaushik/plant-disease-detection-using-cnn https://www.kaggle.com/code/emmarex/plant-disease-detection-using-keras
STEPS	1. Installing Jupyter notebook with python3 2. import the required libraries-numpy,matplotlib.pyplot,pandas,seaborn 3. Importing Data (kaggle and scikit-learn library) and Checking out 4. Exploratory Data Analysis for stock price Prediction and time series analysis 5. Get Data Ready For Training RNN model **6.** Split Data into Train, Test
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition

	4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

Prerequisites: Programming language, machine learning

Concepts related Theory:

Deep Convolutional Neural Network is utilized in this study to identify infected and healthy leaves, as well as to detect illness in afflicted plants. The CNN model is designed to suit both healthy and sick leaves; photos are used to train the model, and the output is determined by the input leaf.

Algorithm:

Importing Libraries: Importing the required libraries which will be used to train the model.such as numpy, tensor flow.

```
import numpy as np
import pickle
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from keras.losses import categorical_crossentropy
from sklearn.metrics import confusion_matrix,classification_report
from keras.utils.vis_utils import plot_model
```

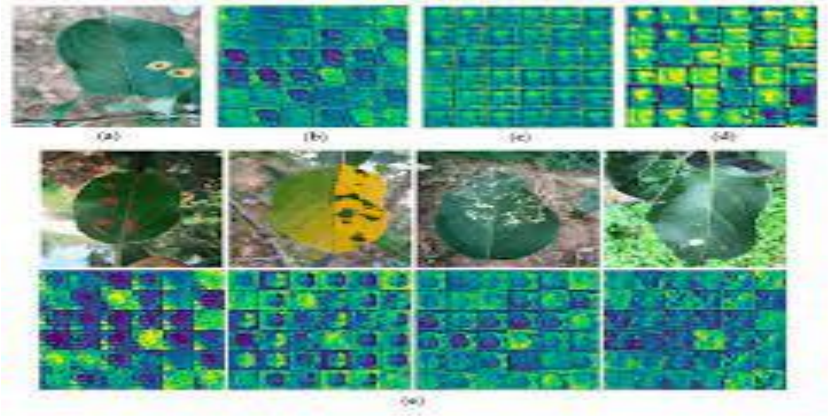
Importing Data and Checking out: As data is in the CSV file, we will read the CSV using pandas read_csv function and check the first 5 rows of the data frame using head().

```
HouseDF.info()
```

```
OUTPUT

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
Avg. Area Income                5000 non-null float64
Avg. Area House Age             5000 non-null float64
Avg. Area Number of Rooms       5000 non-null float64
Avg. Area Number of Bedrooms    5000 non-null float64
Area Population                 5000 non-null float64
Price                           5000 non-null float64
Address                         5000 non-null object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

Exploratory Data Analysis forPlant Disease :



Get Data Ready For Training using CNN Model:

```
print("[INFO] Spliting data to train, test")
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_size=0.2, random_state = 42)
```

```
aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2,horizontal_flip=True,
    fill_mode="nearest")
```

Split Data into Train, Test: Now split our dataset into a training set and testing set using sklearn train_test_split(). The training set will be used for training the model and testing set for testing the model. We are creating a split of 40% training data and 60% of the training set.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

X_train and y_train contain data for the training model. X_test and y_test contain data for the testing model. X and y are features and target variable names.

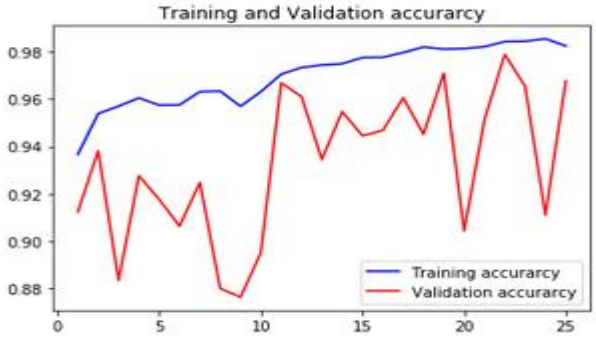
Creating and Training the CNN Model : import and create sklearn linear_model LinearRegression object and fit the training dataset in it.

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
#Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```

CNN Model Evaluation: Now let's evaluate the model by checking out its coefficients and how we can interpret them.

10.



In the above scatter plot, we see data is in a line form, which means our model has done good

predictions.

Conclusion-We have analyzed a CNN Model which will help to predict plant disease.

Review Questions:

1. What is the use of the convolution layer in CNN?
2. What are the advantages of using CNN over DNN?
3. Why is CNN preferred over ANN for image data?
4. How would you visualise features of CNN in an image classification task?
5. What do you understand about shared weights in CNN?
6. Explain the role of a fully connected (FC) layer in CNN.
7. What is the importance of parameter sharing
8. Explain the different types of Pooling.