M.A.M. College of Engineering,

Tiruchirappalli is approved by the AICTE, New Delhi and affiliated to Anna University, Chennai.

Department of Computer Science and Engineering

REPORT ON

HX 8001 PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

(Nalaiya Thiran Program)

PROJECT TITLE

Fertilizer recommendation system for disease prediction

TEAM ID: PNT2022TMID45468

MENTOR:

T M LATHA

TEAM MEMBERS:

1. S. Thulasimani (Team Leader)
2. M. Ramesh Kanna
3. U. Kirubagaran
4. K. Prasanth

Fertilizer Recomentation System for Disease Prediction

PROBLEM STATEMENT:

Mr.Narasimma Rao is a 65 years old man. He had a own farming land and do Agriculture for past 30 Years , In this 30 Years he Faced a problem in Choosing Fertilizers and Controlling of Plant Disease.

- Narasimma Rao wants to know the better recommendation for fertilizers for plants with the disease.
- He has faced huge losses for a long time.
- This problem is usually faced by most farmers.
- Mr. Narasimma Rao needs to know the result immediately.

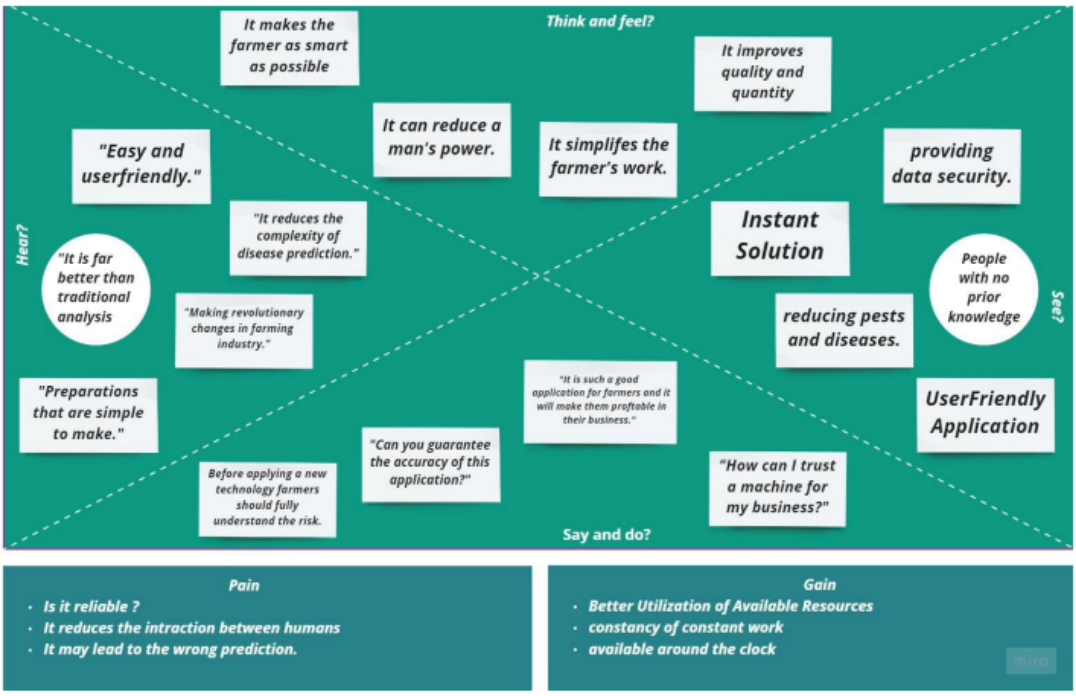
Who does the problem affect?	Persons who do Agriculture
What are the boundaries of the problem?	People who Grow Crops and facing Issues of Plant Disease
What is the issue?	In agricultural aspects, if the plant is affected by leaf disease, then it reduces the growth and productiveness. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants.
When does the issue occur?	During the development of the crops as they will be affected by various diseases.

Where does the issue occur?	The issue occurs in agriculture practicing areas, particularly in rural regions.
Why is it important that we fix the problem?	It is required for the growth of better quality food products. It is important to maximise the crop yield.
What solution to solve this issue?	An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.
What methodology used to solve the issue?	Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

Empathy Map for Fertilizer Recommendation System:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community

Empathy Map:



Pain

- Is it reliable ?
- It reduces the intraction between humans
- It may lead to the wrong prediction.

Gain

- Better Utilization of Available Resources
- constancy of constant work
- available around the clock

Fertilizer Recommendation System For Disease Prediction

Introduction

Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest. Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases. Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Literature Review

[1] Dos Santos Ferreiraet al.[2017] proposed a method to identify unwanted weeds in the soybean field. Unwanted weed includes unwanted grasses and broadleaf. Convolution neural network technique is applied in the process of identifying the weeds in the soybean field. For the purpose of capturing the image, drones were used in it. The database used for analysing purpose includes fifteen thousand pictures weeds, soil, soybean, grass weed, and broadleaf. SafeNet architecture is used for training the neural network. The cafe software includes Alex Net in it. Pynovisao algorithm is used to build a robust image database. The results are compared with Support Vector Machine, Ada Boost, and Random Forest. The accuracy of 99 % is achieved using the convolution neural network. Super pixel algorithm (Simple Linear Iterative Clustering (SLIC) Super pixel) mainly focus on object localization and segmentation of the image.

[2] Carranza-Rojaset al.[2017] proposed a technique for herbarium species identification using deep learning technique. It mainly focuses on how convolution neural networks help in automatic identification of plant species. Image-Net classification performs very well in convolution neural network process. TL is also used for domain related training. Results show a greater accuracy when it is trained and tested for a different set of species. It has been shown in it that by using herbarium dataset Transfer learning is possible to another region even when the species don't match. Handwritten tags and noise can be removed by the pre-processing technique. The transfer learning from herbarium to non-dried plants are clearly listed in the table.

[3] Luet al.[2017b] proposed a technique for identifying the pathogen in the vegetable. Deep convolution neural network technique is used for the identification of the rice disease. Training and testing the model consist of 500 images of rice leaves and stem with 10 types of rice disease in it. Ten fold cross validation method is used for identification of rice disease. The proposed novel model provides an accuracy of 95.48 %. The structure of 10 cross field deep convolution network consist of input (3@512 * 512), convolution (362@244 * 224), stochastic pooling (32@112 * 1112), convolution (16@56 * 56), stochastic pooling(16@28 * 28), convolution (16@28 * 28), stochastic pooling (16@14 *14), and two fully connected one. In the pre-processing stage scale normalization and mean normalization is done for colour image and grey image and then PCA and whitening method is applied. Finally trained and tested feature map is plotted. Recognition accuracy for mean, max and stochastic pooling is as follows 92.11, 93.24,95.48 and the recognition accuracy for different filter (5*5, 9*9, 16*16, 32*32) are 93.15, 92.56, 93.29, 92.48. The proposed method is compared with BP, SVM, and PSO.

[4] Barbedo [2019] proposed a technique based on deep learning for the purpose of image classification. Data augmentation technique helps in the lack of a database for plant image. This paper mainly focuses on identifying the individual lesion and spot instead of considering the whole leaf for identification. While using only lesion and spot the accuracy is 12 % higher than using the entire leaf. The complete details about the recent architecture used for identifying the plant disease and where the data are collected for identifying the plant disease and its accuracy after identification is also mentioned clearly. The list of disorder found in the plant specimen is also listed out clearly in it. Google Net CNN was used in the experimental setup. In the experiment, three different types of images were used and they are 1. Image with-out any modification 2. Image with background removed 3. Expanded dataset. Accuracy for both original and expanded images are calculated.

[5] Barbedo [2018] the problems faced in the machine learning technique has been overcoming by the deep learning concepts such as Convolution Neural Networks (CNN). Large data sets are needed for processing this technique. This paper mainly focuses on how the size of data and its variety affects the performance of the deep learning concepts. 12 plant species with different samples, different disease, and different character are taken into consideration. This analysis describes the different CNN network used for disease classification along with where this large amount of data are collected for classification. Accuracy is also calculated for each deep learning concepts. The number of correctly classified sample divided by the total number of samples provides the accuracy value. List of different plant species and its disease are listed in it. Removing background from image improves the prediction accuracy. This analysis was performed mainly using dataset obtained from different sources.

[6] Tavakoli and Gebbers [2019] presented an analysis of winter wheat nitrogen and assessment of water in the field by using a camera. This experiment was conducted during a period of three years (2012,2013, and 2014). Nitrogen fertilization and different level of water are applied in the field for the purpose of the experiment. Two machine learning algorithm was developed for the purpose of analysis namely Random Forest (RF) and Partial Least Square Regression (PLSR). Specter radiometer was used for radial measurement. Separately Vegetation Index (VI) is also calculated. For analysing the nitrogen content R2(RMSE) model is used and it is calculated separately for both data type. Random forest algorithm performs better in combined-date data. Nitrogen estimation calculation performs better while using the digital camera. It can also be integrated with the smartphone. It has a limitation of accessing only these spectral bands so that the analysis of plant status is also limited.

[7] Grinblatet al.[2016] proposed a method used for the identification of plant using leaf vein pattern. The classification of white bean, red bean and soybean are also done in this. Referred pipeline accuracy is also improved in it. The vein pattern is obtained by analysing the visualization technique with the obtained results. The image processing is done in four different stages namely vein segmentation, central patch extraction, vein measure, and classification. Random forest, support vector machine, and penalized discriminant analysis algorithm are used for classification purpose. Central patch extraction and vein measure are replaced by the convolution neural network technique where it learns from the data set and solve this problem. In the proposed system of CNN, the depth of the model is increased from 2 layers to6 layers. While analysing the results it shows that the accuracy gets improved when we go deep into the layer at the 5th layer an accuracy of 92.6 is achieved.

[8] Ferentinos [2018b] proposed a technique on convolution neural network. For the purpose of training, the model 87,848 images of healthy and diseased plant leaves are taken which includes 25 plant variety. These plants are tested under two different condition namely laboratory and field condition. Alex Net, AlexNetOWTBn, Google Net, over feat and VGG architecture are used for identification of plant disease from the leaves. Its implementation is done using Torch7. It is a machine learning framework. Its training portion is implemented in the Linux environment. 80 % of training data and 20 % testing data for CNN.99.49 % success rate is achieved when using AlexNetOWTBn and 99.53 % of success rate is achieved using VGG model. The success rate for both the original image and the pre-processed image is analysed as well for all the five models. The success rate is more when the model is first trained for field condition and then laboratory condition. The success rate is low when it is tested under laboratory condition and thenfield condition. It can be integrated with the mobile device due to low computational power.

References:

[1] Dhapitha Nesarajan, Lokini Kunalan, Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques", *IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS)-Genova*, Italy, pp. 20404299,2020. https://ieeexplore.ieee.org/abstract/document/9334934

[2] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gon ̧calves da Silva, Hemerson Pistori, andMarcelo Theophilo Folhes. Weed detection in soybean crops using convnets.Computers and Electronicsin Agriculture, 143:314–324, 2017.

[3] V Suma, R Amog Shetty, CNN based Leaf Disease Identification and Remedy Recommendation System, *3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 18959720, 2019. https://ieeexplore.ieee.org/abstract/document/8821872

[4] S. Huddar, S. Gowri, K. Keerthana, S. Vasanthi and S. Rupanagudi, "Novel algorithm for segmentation and automatic identification of pests on plants using image processing", *Third International Conference on Computing Communication and Networking Technologies*, pp. 1-5, 2012. https://ieeexplore.ieee.org/document/6396012

[5] Hamed Tavakoli and Robin Gebbers. Assessing nitrogen and water status of winter wheat using a digitalcamera.Computers and Electronics in Agriculture, 157:558–567, 2019.

[6] K. Kamal, Z. Yin, B. Li, B. Ma and M. Wu, "Transfer learning for fine-grained crop disease classification based on leaf images", *Proc. 10th Workshop Hyperspectral Imag. Signal Process. Evol. Remote Sens. (WHISPERS)*, pp. 1-5, 2019. https://ieeexplore.ieee.org/document/8921213

[7] Konstantinos P Ferentinos. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145:311–318, 2018.

[8] S.B Dhaygude, N.P Kumbhar, "Agricultural Plant Leaf Disease Detection Using Image Processing", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2 Issue 1(2013).

[9] Guillermo L Grinblat, Lucas C Uzal, M ́onica G Larese, and Pablo M Granitto. Deep learning for plantidentification using vein morphological patterns.Computers and Electronics in Agriculture, 127:418–424, 2016.

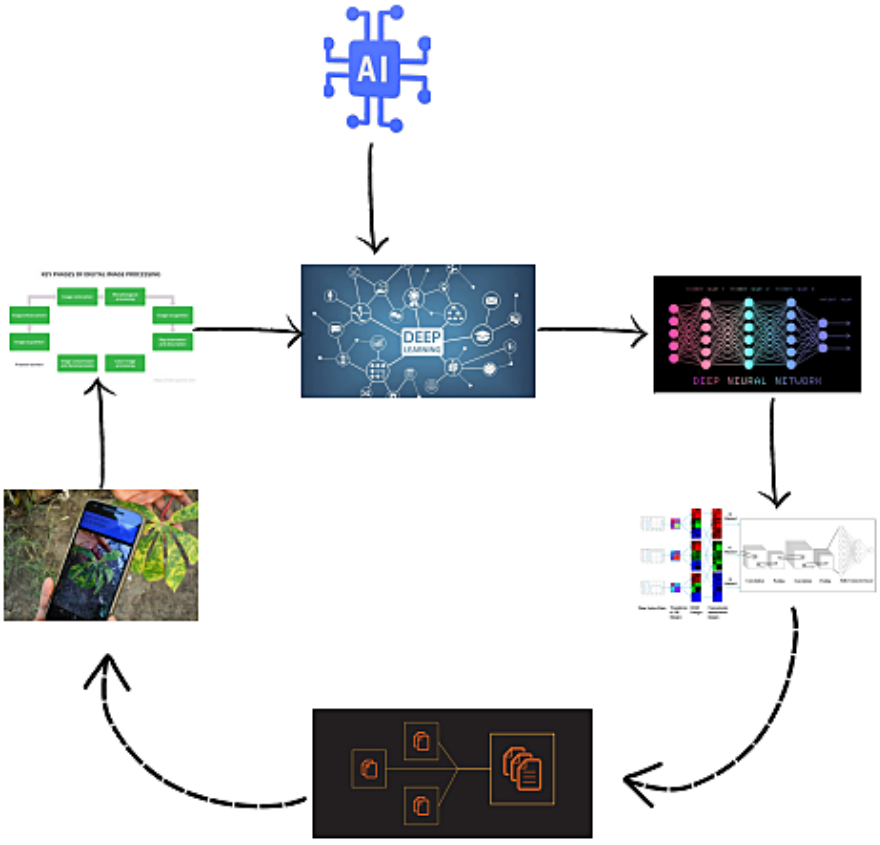
[10] B. Liu, Y. Zhang, D. He and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks", *Symmetry*, 2018.

PROBLEM STATEMENT In India, the agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most of the Indian population depends on agriculture for their livelihood. Agriculture gives an opportunity of employment to the village people to develop a country like India on large scale and give a push in the economic sector. The majority of farmers face the problem of planting an inappropriate crop for their land based on a conventional or non-scientific approach. This is a challenging task for a country like India, where agriculture feeds approximately 42% of the population. And the outcomes for the farmer of choosing the wrong crop for land is moving towards metro city for livelihoods, suicide, quitting the agriculture and give land on lease to industrialist or use for the nonagriculture purpose. The outcome of wrong crop selection is less yield and less profit. PROBLEM SOLUTION The solution to the problem is Machine learning, which is one of the applications of Artificial Intelligence, is being used to implement the proposed system. Crop recommendation is going to recommend you the best crop you can grow in your land as per the soil nutrition value and along with as per the climate in that region. And recommending the best fertilizer for every particular crop is also a challenging task. And the other and most important issue is when a plant gets caught by heterogeneous diseases that effect on less amount of agriculture production and compromises with quality as well. To overcome all these issues this recommendation has been proposed . Nowadays a lot of research and work is being implemented in the smart and modern agriculture domain. Crop recommendation is characterized by a soil database comprised of Nitrogen, Phosphorus, potassium. The ensembles technique is used to build a recommendation model that combines the prediction of multiple machine learning. Models to recommend the right crop based on s

Solution Architecture

In This Below Flow chart will showcase the Solution Architecture we are applied in the project

AI Architecture Flow Diagram::



Project Milestone & Activity Planning:

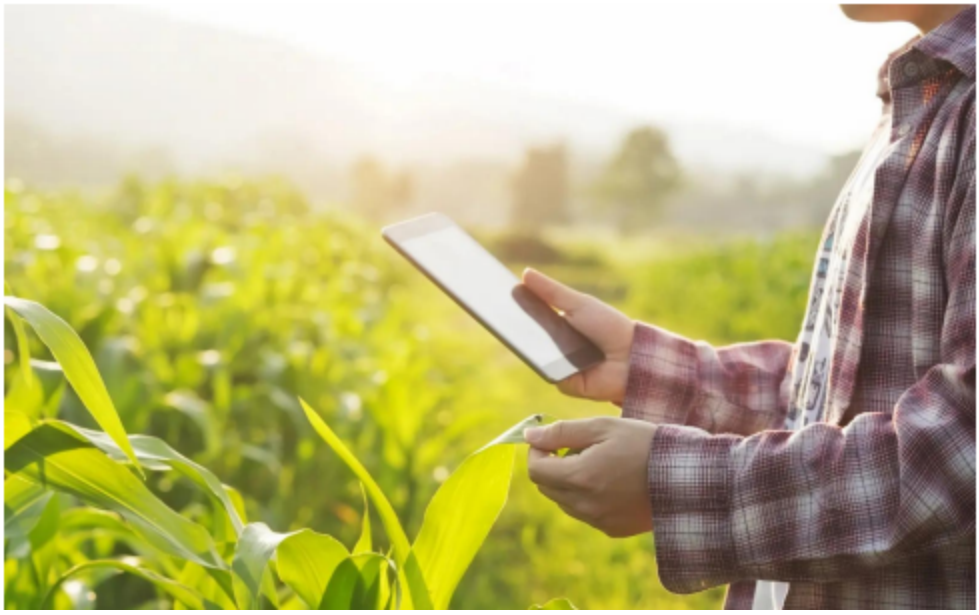
Milestone:

In Modern Technology are increasing and optimizing the Performance of the Artificial Intelligences (AI) Model.

In Based Crop Yield Disease Prediction System, it will helpful for farmers to prevent the crop from the various Disease which can identify the Disease with in a process of capturing the Image at the plant and Machine Learning Algorithm will give affected Disease Name.

In this Project Milestone will be given the Best Solution for the farmer using the complete friendly and simple user interface web application to fetching the solution by own.

In addition, process we are planned to add a valid Module that is Fertilizer recommendation for the Specific Disease. It can give both Artificial fertilizer and Natural Fertilizer in suggestion manner.



Activity List:

In Project Management Planning is an Important task to scheduling the phrase of the project to the Team Member.

In this Activity can shows the various activity are allocated and Done by the Team Members!

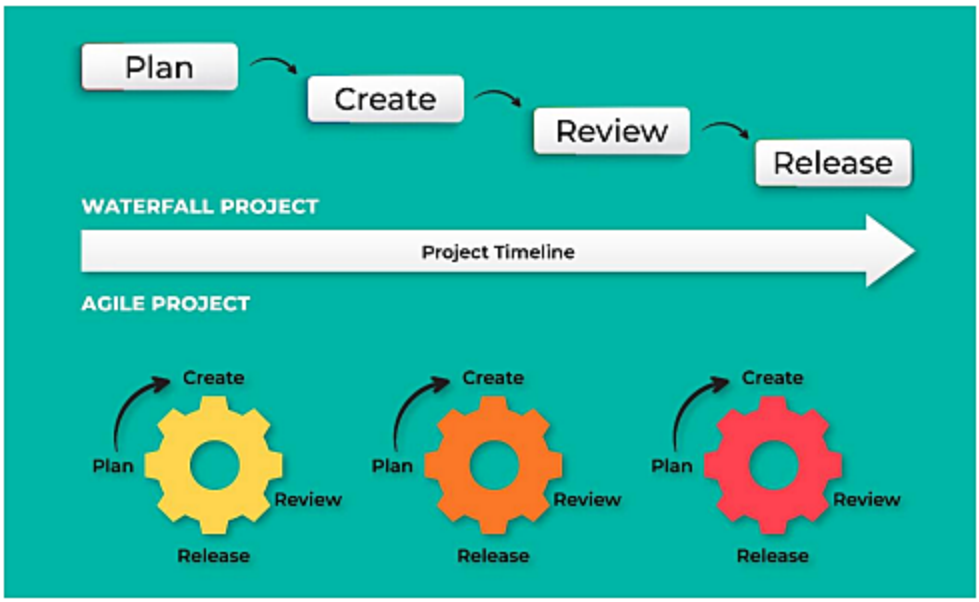
In Project we can Split into the Four Step of Phrases are

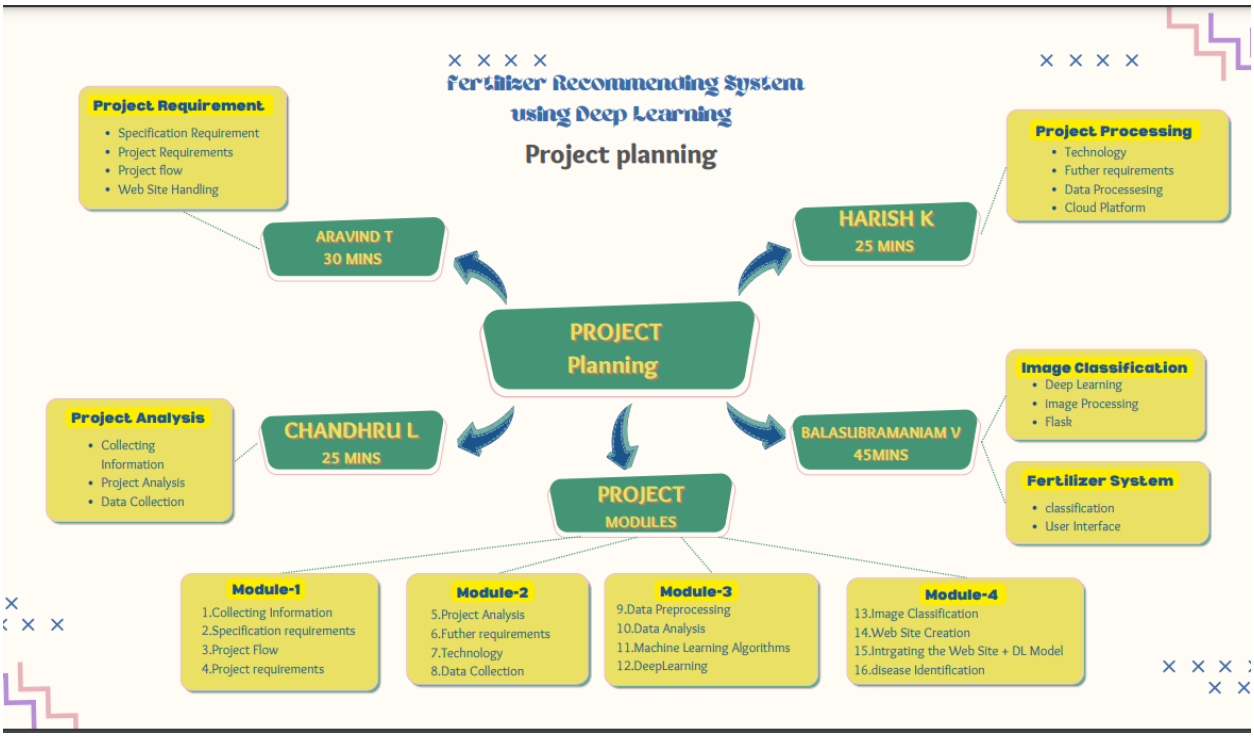
Phrase 1: Information Collection and Requirement Analysis

Phrase 2: Project Planning and Developing Modules

Phrase 3: Implementing the High Accuracy Deep Learning Algorithm to Perform

Phrase 4: Deploying the Model on Cloud and Testing the Model and UI Performance

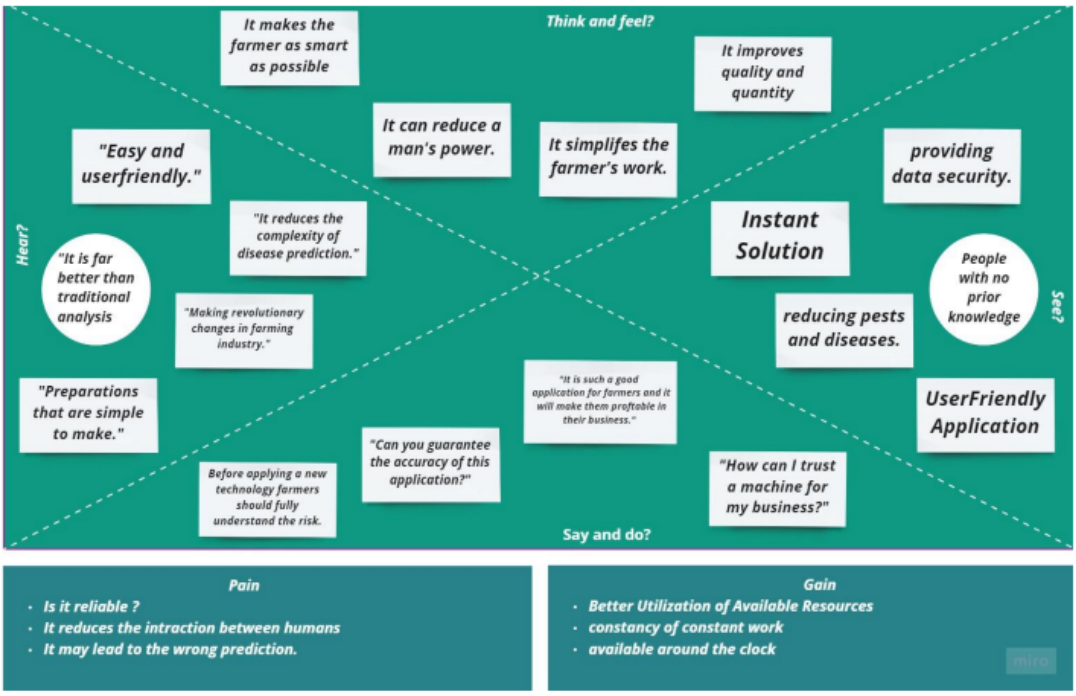



Fertilizer Recommending System using Deep Learning
Project planning
Project Requirement
Specification Requirement
Project Requirements
Project flow
Web Site Handling
ARAVIND T
30 MINS
HARISH K
25 MINS
Project Processing
Technology
Futher requirements
Data Processesing
Cloud Platform
PROJECT
Planning
Project Analysis
Collecting Information
Project Analysis
Data Collection
CHANDHRU L
25 MINS
BALASUBRAMANIAM V
45MINS
Image Classification
Deep Learning
Image Processing
Flask
Fertilizer System
classification
User Interface
PROJECT
MODULES
Module-1
1.Collecting Information
2.Specification requirements
3.Project Flow
4.Project requirements
Module-2
5.Project Analysis
6.Futher requirements
7.Technology
8.Data Collection
Module-3
9.Data Preprocessing
10.Data Analysis
11.Machine Learning Algorithms
12.DeepLearning
Module-4
13.Image Classification
14.Web Site Creation
15.Intrgating the Web Site + DL Model
16.disease Identification

Empathy Map for Fertilizer Recommendation System:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community

Empathy Map:



Pain

- Is it reliable ?
- It reduces the intraction between humans
- It may lead to the wrong prediction.

Gain

- Better Utilization of Available Resources
- constancy of constant work
- available around the clock

Digital Image:

A Digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels
Pixel Value Typically Represent gray levels,colors,heights,opacities etc

Common Image formats includes:

1 sample per point(B&W or Grayscales)
3 samples per point(Red,Green, and Blue)
4 samples per point(Red,Green,Blue, and "Alpha")

Image Processing in Deep Learning:

Image processing is an play in major role in Computer Vision.Computer Vision is an one of the part in Artificial Intelligences.Computer Vision can interact with a image data types such like Image Classification,object localization,object Detection,Segmentation etc

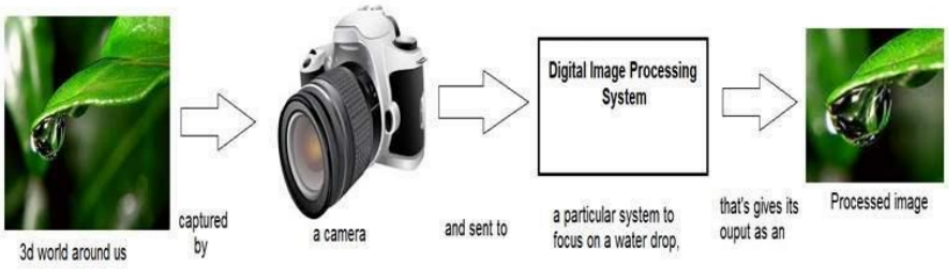
In nature of Computer vision using an Deep Learning Algorithms,Open CV,Pose Estimation Python Libraries etc, In Deep Learning it has been working on Neural Network which can interact and Extract the feature from the image and creating the Deep Learning Model it requires to build with high Accuracy and Less in Model Size can reduce the Computation power of the System,Mobile,cloud interface

Case Study of the Model Building in Deep Learning:

Image Processing:

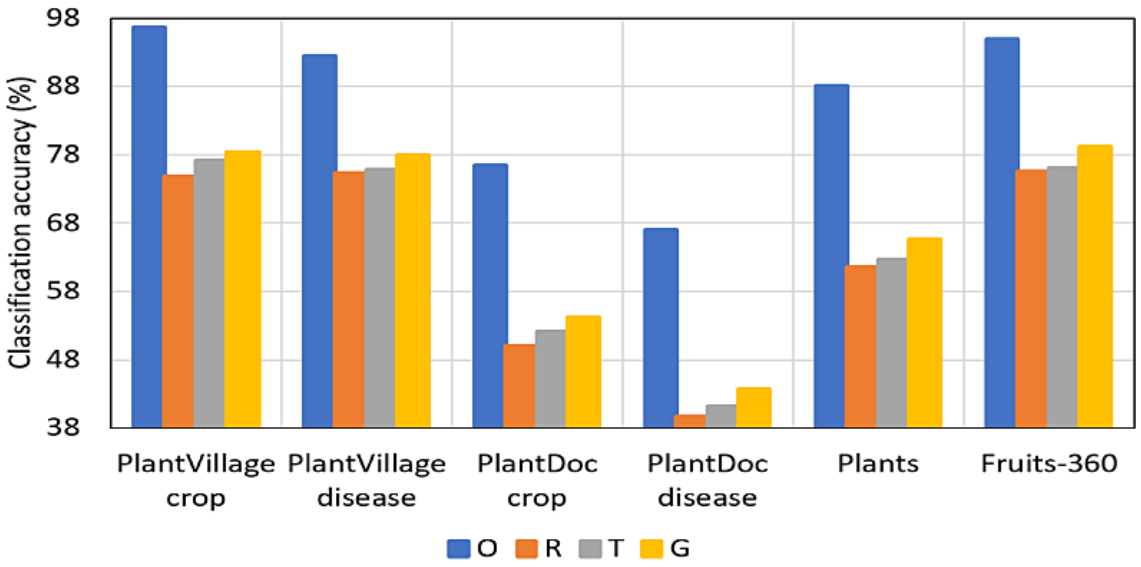
Image Processing is a method to convert an image into digit form and perform some operations on it, in order to get an enhanced image or to extract some useful information form it.

It is a type of signal Dispensation in which input is image,like video frames or photograph and output may be image or characteristics associated with that image



Project Dataset:

In those above two dataset are integrate into a single dataset for using our project to obtain the maximum accuracy form the given Dataset.



Project Flow :

Web application:

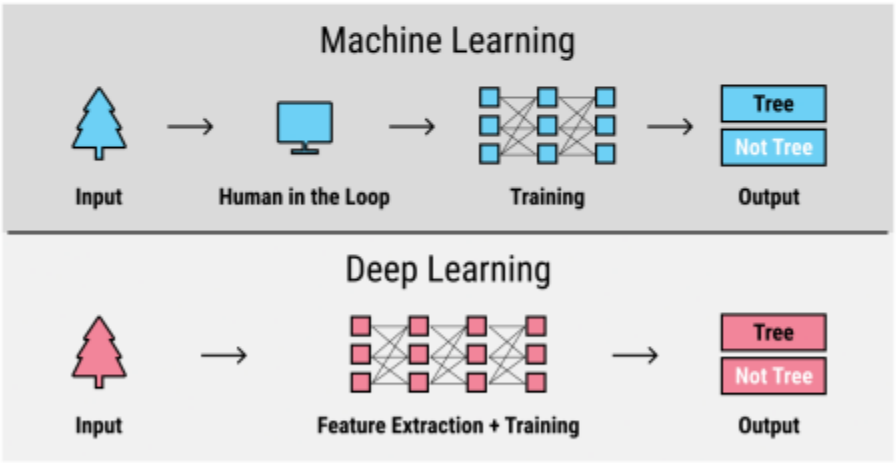
A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.

Farmers can interact with the portal build with a single window access of Information and services in Agricultural Sector in India for farmers and other stakeholders Use of icons/pictures/ images and graphical interfaces to represent links and information for quicker understanding of the farmers. Alternate lighter Home page and alternate text display for icons/pictures/images for faster access depending upon connectivity speed. Consistent and Easy to Use interface- Consistent design of the web pages for common look and feel. Design in way so that the desired document can be traced in maximum 3 clicks. Interacts with the user interface to upload images of diseased leaf. Well designed home page conveying theme and purpose. Single Sign-on to access all information and services on the portal. Self service- Interface to user to set/reset his/her own password. Our model built analyses the Disease and suggests the farmer with fertilizers are to be used. The portal should allow the user to fulfill his needs, wherever possible, through self-service.

Dataset:

A dataset is a collection of data in which data is arranged in some order. A dataset can contain any data from a series of an array to a database table. During the development of the project, the developers completely rely on the datasets. The datasets are divided into two parts:

- **Training Set:** A subset of dataset to train the machine learning model, and we already know the output.
- **Test set:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

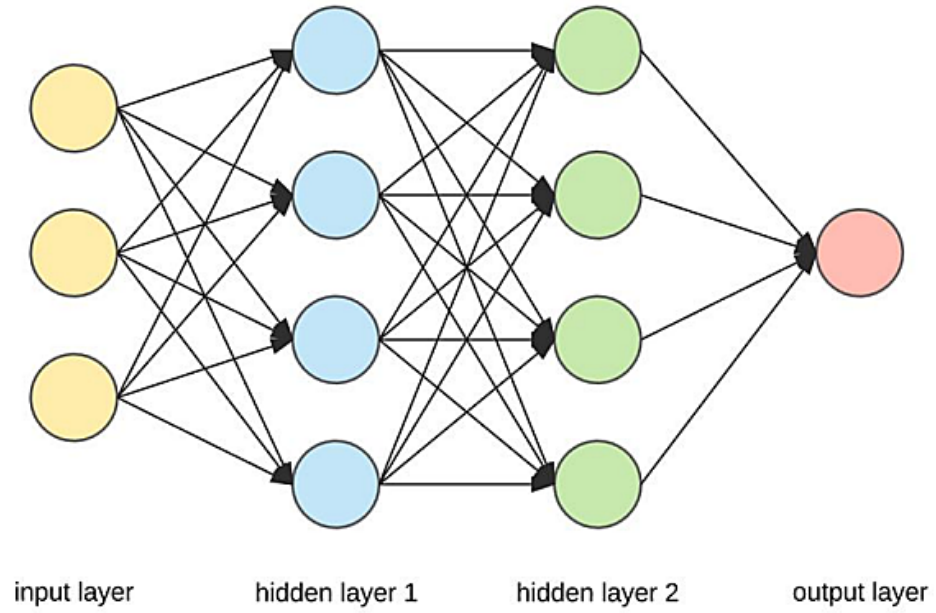


Some of the popular sites to download datasets are Kaggle Datasets, UCI Machine Learning Repository, Datasets via AWS and Google's Dataset Search Engine.

Neural network layers:

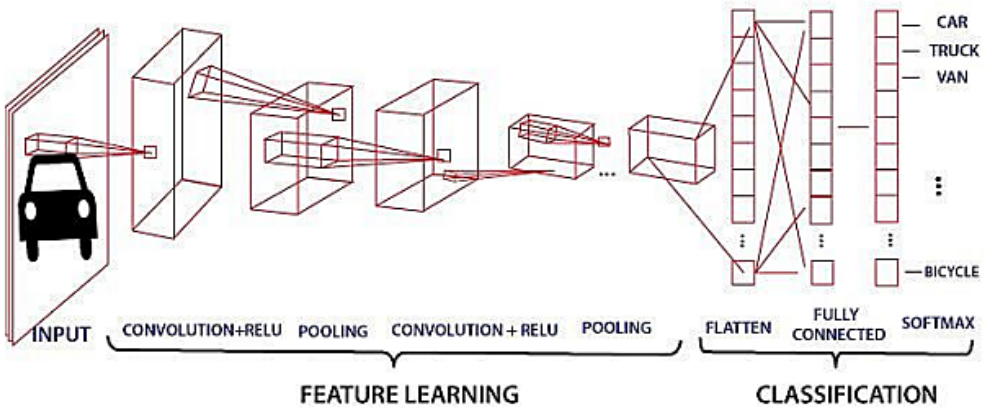
A neural network is made up of vertically stacked components called **Layers**. Each dotted line in the image represents a layer. There are three types of layers in a NN-

- **Input Layer**– First is the input layer. It will accept the data and pass it to the rest of the network.
- **Hidden Layer**– The second type of layer is called the hidden layer. Hidden layers are either one or more in number for a neural network.
- **Output layer**– The last type of layer is the output layer. The output layer holds the result or the output of the problem. Raw images get passed to the input layer and we receive output in the output layer.



Convolutional Neural Network:

It is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used. CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image.



Load the trained images and fit the model:

Loading and transformation are two main concepts which are essential to do image recognition in Keras. Loading and transformation of the images is the starting step of the recognition process.

We used a deep neural network to classify the endless dataset, and we found that it will not classify our data best. When we used the deep neural network, the model accuracy was not sufficient, and the model could improve. This improvement will be made with the help of the convolutional neural network.

Training and Testing the images:

In the training section, we trained our CNN model on the dataset (Endless dataset), and it seemed to reach a reasonable loss and accuracy. If the model can take what it has learned and generalize itself to new data, then it would be a true testament to its performance. This will be done in the same way as we have done in our previous topic.

After validation, we test our model by grabbing an image from the web page.

Save the model and its dependencies:

Keras separates the concerns of saving your model architecture and saving your model weights. Model weights are saved to an HDF5 format. This grid format is ideal for storing multi-dimensional arrays of numbers. The model structure can be described and saved using two different formats: JSON and YAML.

Deploying using Flask:

Fertilizers Recommendation System For Disease Prediction

Problem Statement :

In India, The Agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most of the Indian population depends on agriculture for their livelihood. Agriculture gives an opportunity of employment to the village people to develop a country like India on large scale and give a push in the economic sector. The majority of farmers face the problem of planting an inappropriate crop for their land based on a conventional or non-scientific approach. This is a challenging task for a country like India, where agriculture feeds approximately 42% of the population. And the outcomes for the farmer of choosing the wrong crop for land is moving towards metro city for livelihoods, suicide, quitting the agriculture and give land on lease to industrialist or use for the non-agriculture purpose. The outcome of wrong crop selection is less yield and less profit.

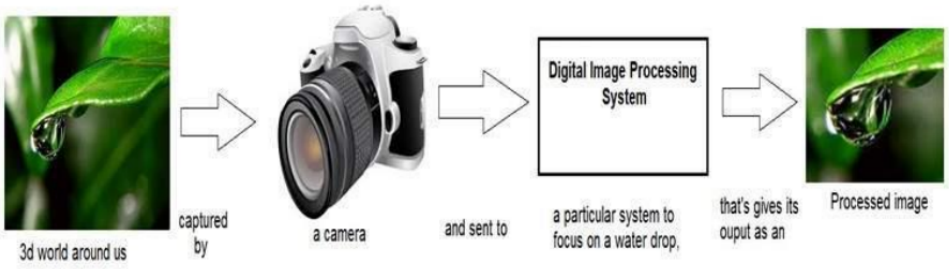
Problem Solution :

The solution to the problem is Machine learning, which is one of the applications of Artificial Intelligence, is being used to implement the proposed system. Crop recommendation is going to recommend you the best crop you can grow in your land as per the soil nutrition value and along with as per the climate in that region. And recommending the best fertilizer for every particular crop is also a challenging task. And the other and most important issue is when a plant gets caught by heterogeneous diseases that effect on less amount of agriculture production and compromises with quality as well. To overcome all these issues this recommendation has been proposed . Nowadays a lot of research and work is being implemented in the smart and modern agriculture domain. Crop recommendation is characterized by a soil database comprised of Nitrogen, Phosphorus, potassium. The ensembles technique is used to build a recommendation model that combines the prediction of multiple machine learning. Models to recommend the right crop based on soil value and the best fertilizer to use.

Image Processing:

Image Processing is a method to convert an image into digit form and perform some operations on it, in order to get an enhanced image or to extract some useful information form it.

It is a type of signal Dispensation in which input is image,like video frames or photograph and output may be image or characteristics associated with that image



Digital Image:

A Digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels
Pixel Value Typically Represent gray levels,colors,heights,opacities etc

Common Image formats includes:

1 sample per point(B&W or Grayscales)
3 samples per point(Red,Green, and Blue)
4 samples per point(Red,Green,Blue, and "Alpha")

Image Processing in Deep Learning:

Image processing is an play in major role in Computer Vision.Computer Vision is an one of the part in Artificial Intelligences.Computer Vision can interact with a image data types such like Image Classification,object localization,object Detection,Segmentation etc

In nature of Computer vision using an Deep Learning Algorithms,Open CV,Pose Estimation Python Libraries etc, In Deep Learning it has been working on Neural Network which can interact and Extract the feature from the image and creating the Deep Learning Model it requires to build with high Accuracy and Less in Model Size can reduce the Computation power of the System,Mobile,cloud interface

Case Study of the Model Building in Deep Learning:

MODEL BUILDING FOR FRUIT DISEASE PREDICTION

Convolution Layers:

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function

1.Convolutional Layer:

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).
The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.
The convolution layer in CNN passes the result to the next layer once applying the convolution operation in the input. Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact

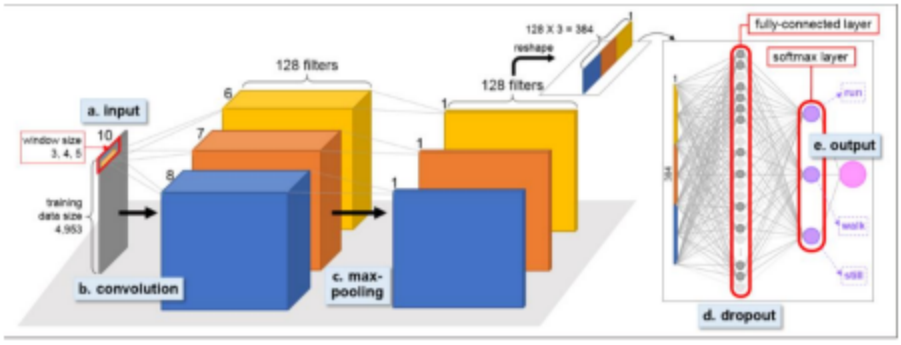


Fig 1.0 Convolution Operation

2. Pooling Layer:

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. It basically summarizes the features generated by a convolution layer.
In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.
This CNN model generalizes the features extracted by the convolution layer, and helps the networks to recognize the features independently. With the help of this, the computations are also reduced in a network.

In Pooling Layer consists two types of pooling Layer:

1. Max Pooling
2. Average Pooling

Max Pooling and Average Pooling:

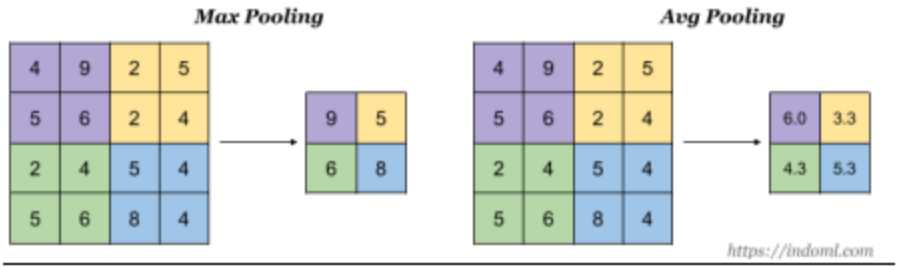


Fig 1.1 Pooling Layers in CNN

3. Fully Connected Layer or Dense Layer:

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place. The reason two layers are connected is that two fully connected layers will perform better than a single connected layer. These layers in CNN reduce the human supervision
Hidden Layer also called as Fully Connected layer,

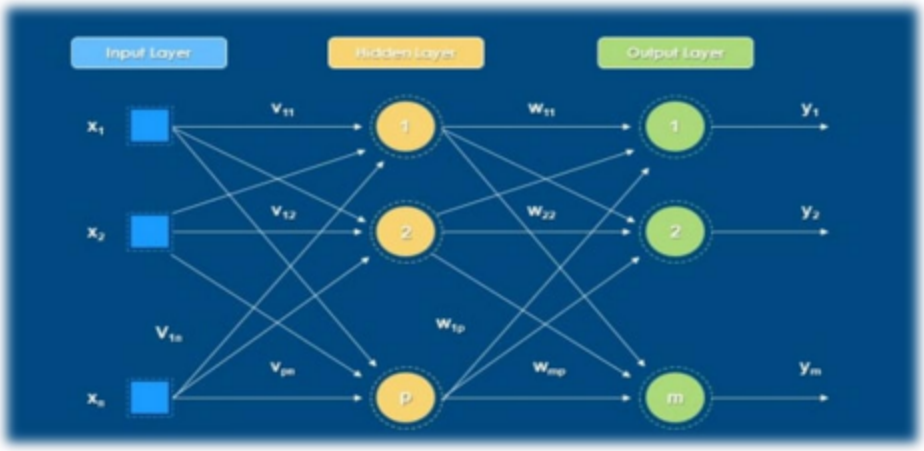
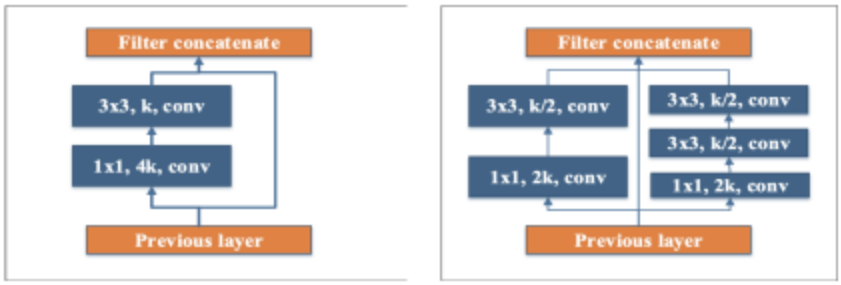


Fig 1.2 Fully Connected layer – Hidden layer

Two Types of Dense Layers are:



(a) original dense layer (b) 2-way dense layer

Fig 1.3 Two types of Dense Layer

4. Dropout:

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

```
from flask import Flask, render_template, request, Markup,url_for, redirect
import numpy as np
import pandas as pd
from utils.disease import disease_dic
from utils.fertilizer import fertilizer_dic
import requests
import config
import pickle
import io
import torch
from torchvision import transforms
from PIL import Image
from utils.model import ResNet9
from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user,
current_user
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt

#
================================================================================
=================

# ----------------------LOADING THE TRAINED MODELS -----------------------------------------------

# Loading plant disease classification model

disease_classes = ['Apple___Apple_scab',
```

'Apple___Black_rot',
'Apple___Cedar_apple_rust',
'Apple___healthy',
'Blueberry___healthy',
'Cherry_(including_sour)___Powdery_mildew',
'Cherry_(including_sour)___healthy',
'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
'Corn_(maize)___Common_rust_',
'Corn_(maize)___Northern_Leaf_Blight',
'Corn_(maize)___healthy',
'Grape___Black_rot',
'Grape___Esca_(Black_Measles)',
'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
'Grape___healthy',
'Orange___Haunglongbing_(Citrus_greening)',
'Peach___Bacterial_spot',
'Peach___healthy',
'Pepper,_bell___Bacterial_spot',
'Pepper,_bell___healthy',
'Potato___Early_blight',
'Potato___Late_blight',
'Potato___healthy',
'Raspberry___healthy',
'Soybean___healthy',
'Squash___Powdery_mildew',
'Strawberry___Leaf_scorch',
'Strawberry___healthy',
'Tomato___Bacterial_spot',
'Tomato___Early_blight',
'Tomato___Late_blight',
'Tomato___Leaf_Mold',

```
            'Tomato___Septoria_leaf_spot',
            'Tomato___Spider_mites Two-spotted_spider_mite',
            'Tomato___Target_Spot',
            'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
            'Tomato___Tomato_mosaic_virus',
            'Tomato___healthy']

disease_model_path = 'models/plant_disease_model.pth'
disease_model = ResNet9(3, len(disease_classes))
disease_model.load_state_dict(torch.load(
    disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()


# Loading crop recommendation model

crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))


#
==========================================================================
============

# Custom functions for calculations


def weather_fetch(city_name):
    """
    Fetch and returns the temperature and humidity of a city
```

```
    :params: city_name
    :return: temperature, humidity
    """
    api_key = config.weather_api_key
    base_url = "http://api.openweathermap.org/data/2.5/weather?"

    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()

    if x["cod"] != "404":
        y = x["main"]

        temperature = round((y["temp"] - 273.15), 2)
        humidity = y["humidity"]
        return temperature, humidity
    else:
        return None


def predict_image(img, model=disease_model):
    """
    Transforms image to tensor and predicts disease label
    :params: image
    :return: prediction (string)
    """
    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
```

```
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)

    # Get predictions from model
    yb = model(img_u)
    # Pick index with highest probability
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    # Retrieve the class label
    return prediction

#
================================================================================
==================
# ---------------------------------- FLASK APP ----------------------------------------------


app = Flask(__name__)

db = SQLAlchemy(app)
bcrypt = Bcrypt(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['SECRET_KEY'] = 'thisisasecretkey'


login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'


@login_manager.user_loader
```

```
def load_user(user_id):
    return User.query.get(int(user_id))


class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)


class RegisterForm(FlaskForm):
    username = StringField(validators=[
                        InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})

    password = PasswordField(validators=[
                         InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

    submit = SubmitField('Register')

    def validate_username(self, username):
        existing_user_username = User.query.filter_by(
            username=username.data).first()
        if existing_user_username:
            raise ValidationError(
                'That username already exists. Please choose a different one.')


class LoginForm(FlaskForm):
    username = StringField(validators=[
```

```
                    InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})

    password = PasswordField(validators=[
                     InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})

    submit = SubmitField('Login')


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user:
            if bcrypt.check_password_hash(user.password, form.password.data):
                login_user(user)
                return redirect (url_for('dashboard'))
    return render_template('login.html', form=form)


@app.route('/dashboard', methods=['GET', 'POST'])
@login_required
def dashboard():
    return render_template('dashboard.html')
```

```
@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('login'))


@ app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()

    if form.validate_on_submit():
        hashed_password = bcrypt.generate_password_hash(form.password.data)
        new_user = User(username=form.username.data, password=hashed_password)
        db.session.add(new_user)
        db.session.commit()
        return redirect(url_for('login'))

    return render_template('register.html', form=form)

# render home page


# render crop recommendation form page


@ app.route('/crop-recommend')
def crop_recommend():
    title = 'Crop Recommendation'
```

```
    return render_template('crop.html', title=title)

# render fertilizer recommendation form page


@ app.route('/fertilizer')
def fertilizer_recommendation():
    title = 'Fertilizer Suggestion'

    return render_template('fertilizer.html', title=title)

# render disease prediction input page




#
================================================================================
==================

# RENDER PREDICTION PAGES

# render crop recommendation result page


@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'Crop Recommendation'

    if request.method == 'POST':
        N = int(request.form['nitrogen'])
```

```
        P = int(request.form['phosphorous'])
        K = int(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
            my_prediction = crop_recommendation_model.predict(data)
            final_prediction = my_prediction[0]

            return render_template('crop-result.html', prediction=final_prediction, title=title)

        else:

            return render_template('try_again.html', title=title)

# render fertilizer recommendation result page


@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'Fertilizer Suggestion'

    crop_name = str(request.form['cropname'])
    N = int(request.form['nitrogen'])
    P = int(request.form['phosphorous'])
    K = int(request.form['pottasium'])
```

```
# ph = float(request.form['ph'])

df = pd.read_csv('Data/fertilizer.csv')

nr = df[df['Crop'] == crop_name]['N'].iloc[0]
pr = df[df['Crop'] == crop_name]['P'].iloc[0]
kr = df[df['Crop'] == crop_name]['K'].iloc[0]

n = nr - N
p = pr - P
k = kr - K
temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
max_value = temp[max(temp.keys())]
if max_value == "N":
    if n < 0:
        key = 'NHigh'
    else:
        key = "Nlow"
elif max_value == "P":
    if p < 0:
        key = 'PHigh'
    else:
        key = "Plow"
else:
    if k < 0:
        key = 'KHigh'
    else:
        key = "Klow"

response = Markup(str(fertilizer_dic[key]))
```

```
    return render_template('fertilizer-result.html', recommendation=response, title=title)

# render disease prediction result page


@app.route('/disease-predict', methods=['GET', 'POST'])
def disease_prediction():
    title = 'Disease Detection'

    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files.get('file')
        if not file:
            return render_template('disease.html', title=title)
        try:
            img = file.read()

            prediction = predict_image(img)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)


#
================================================================================
==================
if __name__ == '__main__':
```

app.run(debug=True)

- Tanh or hyperbolic tangent Activation Function.
- ReLU (Rectified Linear Unit) Activation Function.
- Leaky ReLU.

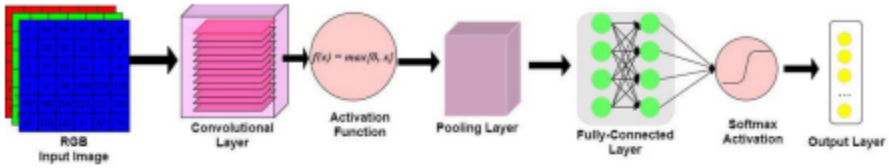


Fig 1.5 Activation Function in CNN

Buliding the CNN model in Image Classification for the Project:

```
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(4,4),input_shape=(28, 28, 1), activation='relu',))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 25, 25, 32)        544
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 128)               589952
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 591,786
Trainable params: 591,786
Non-trainable params: 0
```