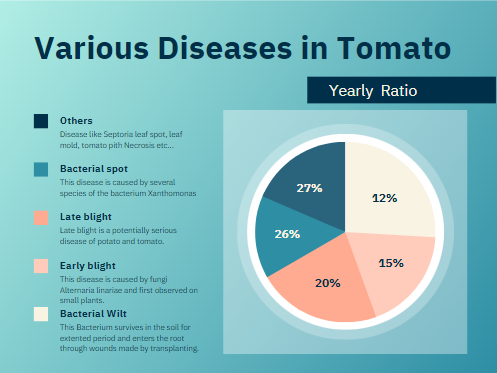
**Smart Tomato Cultivation and Disease Management System**

1.1 Problem Statement:

* The main problem of tomato crises is climate change, it ruined the growth of tomato. The perfect temperature to grow a tomato crop is between 21\* to 27\* celcius.
* Also if some tomatoes are still good from harvesting a tomato to exporting it, There are huge amount of tomatoes are wasting in many ways such as,
  + Unripe
  + Over ripe
  + Spoils during transporting
* Causes of this huge amount of wasted tomatoes, it impacts on high demand for tomato in the recent days.
* High cost because of labour.
* After harvesting, major number of tomatoes will spoil during transportation.
* Also tomato was affected by Bacterial wilt, Early blight, late blight, Bacterial spot etc…



* These diseases and disorder causes by bacteria and fungi and affect the growth, shape, colour and taste of the tomato.

1.2 Concept (or) Solution:

* We uses soil moisture sensor to see the moisture and temperature sensor to see the temperature.
* The crop growing is under polyhouse technique, so that the outer temperature cannot do a major difference on crop growth.
* Using micro controller, We can know it’s dry or wet, if it’s wet(below 21\*Celcius), the system will automatically dry the soil and if it’s dry (above 24\*celcius), it will automatically Irrigate the soil.
* We uses camera to see the picture of tomato.
* Uses Machine learning algorithms to identify, if it’s ripped (or) un ripped (or) over ripped.
* If it’s ripped, we use robotics hand to harvest the ripped tomato.
* While transportation, we uses plastic trays to contain tomatoes.
* We uses Machine learning, Deep learning algorithm and scikit-learn library that uses python code and data collected by camera to recognize the disease in the tomato plant.
* Use organic pesticides for preventing a crop from those diseases.

1.3 Machine learning Algorithms:

1.3.1 To Recognise ripped and unripped tomatoes:

PROGRAM:

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import LabelEncoder

X = np.array([[0, 5, 0],

[1, 3, 1],

[1, 4, 0],

[0, 2, 1],

[1, 6, 0]])

# Labels: 0=unripe, 1=ripe

y = np.array([0, 1, 1, 0, 1])

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create and train a Decision Tree classifier

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = clf.predict(X\_test)

# Calculate accuracy

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

# Classify a new tomato (replace with your own tomato's features)

new\_tomato = np.array([[0, 4, 0]])

prediction = clf.predict(new\_tomato)

if prediction[0] == 0:

print("The tomato is unripe.")

else:

print("The tomato is ripe.")

ALGORITHM:

**Certainly! Here's the step-by-step algorithm for the provided program:**

**1. Import Libraries:** Import the required libraries for your machine learning tasks, such as NumPy for numerical operations and scikit-learn for machine learning tools.

**2. Generate Example Data**: Define example data in a 2D NumPy array X where each row represents a tomato and its features (color, size, texture), and a 1D NumPy array y containing labels (0 for unripe, 1 for ripe).

**3. Split the Data:** Use train\_test\_split to split the data into training and testing sets: X\_train, X\_test, y\_train, and y\_test. Specify the test size and a random seed for reproducibility.

**4. Create and Train a Decision Tree Classifier:**

* + Create a DecisionTreeClassifier instance clf.
  + Train the classifier using the training data (X\_train and y\_train) using the fit method.

**5. Make Predictions and Calculate Accuracy:**

* + Use the trained classifier to make predictions on the test set (X\_test) using the predict method. Store the predicted labels in y\_pred.
  + Calculate the accuracy of the predictions by comparing y\_pred with the true labels (y\_test) using accuracy\_score.

**6. Classify a New Tomato:**

* + Create a new tomato's features in a NumPy array new\_tomato.
  + Use the trained classifier (clf) to predict whether the new tomato is unripe (0) or ripe (1) based on its features.
  + Print the classification result based on the prediction.

And that's it! The program generates example data, trains a Decision Tree classifier, evaluates its accuracy, and then demonstrates the classification of a new tomato based on its features.

1.3.2 To Recognise Disease:

PROGRAM:

import os

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from sklearn.preprocessing import StandardScaler

from sklearn.pipeline import Pipeline

from skimage.io import imread

from skimage.transform import resize

# Replace these paths with the paths to your dataset

dataset\_path = 'path\_to\_dataset\_folder'

disease\_classes = os.listdir(dataset\_path)

# Load and preprocess the dataset

X = []

y = []

for class\_idx, disease\_class in enumerate(disease\_classes):

class\_path = os.path.join(dataset\_path, disease\_class)

for image\_name in os.listdir(class\_path):

image\_path = os.path.join(class\_path, image\_name)

image = imread(image\_path)

resized\_image = resize(image, (100, 100)) # Resize the image to a consistent size

X.append(resized\_image.flatten())

y.append(class\_idx)

X = np.array(X)

y = np.array(y)

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a pipeline with preprocessing and SVM classifier

svm\_model = Pipeline([

('scaler', StandardScaler()),

('svm', SVC(kernel='linear', C=1.0))

])

# Train the SVM model

svm\_model.fit(X\_train, y\_train)

# Predict disease classes on the test set

y\_pred = svm\_model.predict(X\_test)

# Calculate accuracy

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

ALGORITHM:

Absolutely, let's break down the steps in the algorithm for the given program:

**1. Import Libraries**: Import the required libraries for your machine learning tasks, such as scikit-learn for the classifier and preprocessing, NumPy for numerical operations, os for file handling, and skimage for image processing.

**2. Specify Dataset Path**: Set the dataset\_path variable to the path where your dataset is located. This should be a folder containing subfolders for each disease class.

**3. Load and Preprocess the Dataset:**

* Initialize empty lists X and y to store features and labels respectively.
* Loop through each disease class in disease\_classes:
* Combine the dataset\_path with the current disease class to get class\_path.
* Inside the class loop, loop through each image in the class folder:
* Create the full image\_path by combining class\_path with the image name.
* Read the image using imread.
* Resize the image to a consistent size (e.g., 100x100) using resize.
* Flatten the resized image into a 1D array using .flatten().
* Append the flattened image to the X list.
* Append the corresponding class\_idx (disease class label) to the y list.

**4. Convert to Arrays**: Convert the X and y lists to NumPy arrays for easier manipulation.

**5. Split the Dataset**:

* Use train\_test\_split to split the dataset into training and testing sets. This function randomly shuffles and partitions the data.
* Specify the test size (e.g., 20% of the data) and a random seed for reproducibility.

**6. Create a Pipeline:**

* Build a scikit-learn pipeline using Pipeline:
* The first step is feature scaling using StandardScaler to normalize the data.
* The second step is an SVM classifier with a linear kernel (SVC with 'linear' kernel) and regularization parameter C set to 1.0.

**7. Train the Model:**

* Fit (train) the SVM model using the training data (X\_train and y\_train) using the fit method on the pipeline.

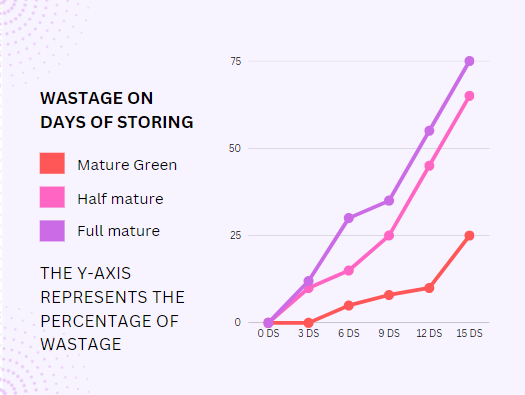
**8. Predict and Evaluate:**

* Use the trained model to predict disease classes on the test set (X\_test) using the predict method on the pipeline.
* Calculate the accuracy of the predictions by comparing them to the true labels (y\_test) using accuracy\_score.
* Print the accuracy.

This algorithm covers loading and preprocessing the dataset, splitting it into training and testing sets, creating a machine learning pipeline, training the SVM model, making predictions, and evaluating the model's accuracy.

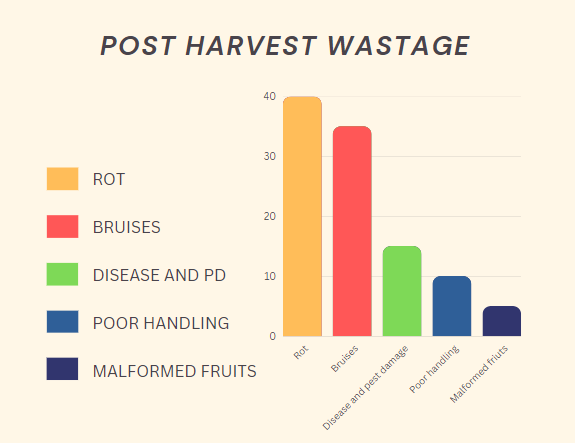
1.4 Tomato Wastage Ratios:

1.4.1 Days of Storing:



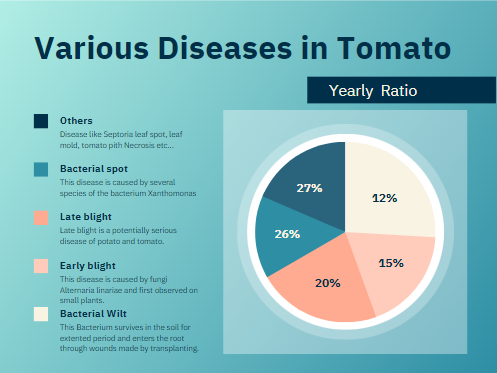
* It clearly indicates the Ripped tomatoes can’t be stored for a long period of time.
* So, for a long distance of transporting tomatoes. We should harvest Mature green to stay in good ripped state when it’s transported.
* And also for small distance transport, we should use full mature that will transported in less time
* For transporting a decent medium distance, we transport half mature.
* Using a trap while transporting while highly reduce the spoilage of tomatoes.
* So, after our idea the spoilage reduce rate would be 80% of the spoilage tomatoes are safe now.

1.4.2 Post harvest wastage:



* 90% of rotten problems are caused because of harvesting of over-ripped tomatoes, transportation, poor handling.
* These all can be prevented using the previous solutions of safe transportation and automated harvesting.
* Bruises and pest damage in tomatoes caused by insects that are eating a tomato from inside, that should be pesticide before harvesting in a organic way.
* Poor handling is a silly mistake but cause a large damage on the tomatoes. So handling a tomato while transporting will play a major role in it.
* This will entirely stop the tomato from post harvest problems.

1.4.3 Various of Disease and Disorder:



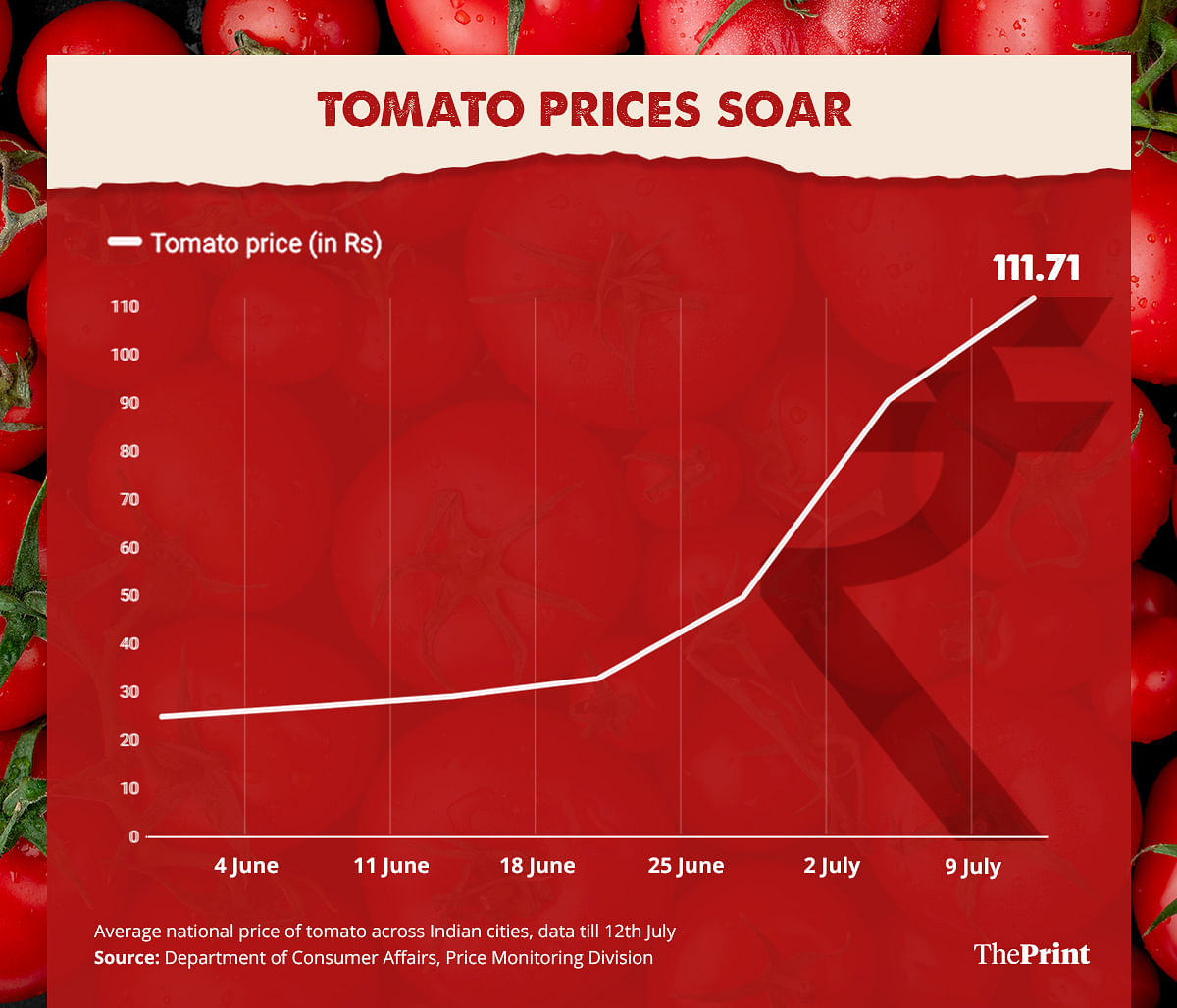
* Control of bacterial wilt is difficult in some plants like corn, beans and cabbage. In tomatoes better remove (or) destroy all the infectious plants to stop spreading this disease to other plants, because there is no chemical for this disease.
* Use resistant or tolerant tomato cultivars to prevent early blights from tomatoes. Don’t irrigate the already wet tomatoes, this will easy prevented by our automation irrigation and drying system.
* To prevent late blight, keep the leaves dry and keep distance between every plants and pull out then destroy the affected plant.
* If the plant affected by bacterial spots then use fungicide products to cure it. To prevent from it, only use certified seeds and plants.

1.5 Uniqueness of the Idea:

* Automation Drying or irrigating respect to the temperature.
* No labour needed.
* Automation harvesting respect to ripeness of the tomato.
* Recognizes the disease and disorder using Cameras and Machine learning Techniques.
* Harvesting the tomatoes upon there transportation distance, so that the number of rotten tomatoes will be totally reduced.
* Using trays to avoid wound and spoilage of tomatoes while transporting.
* Agriculture takes place under polyhouse, so that the outer natural doesn’t impact much to the growth of the plant.

1.6 Survey status of the problem:

* In the recent times, India facing a large amount of tomato crises.
* If we had a solution to minimize this wastage of tomato, we won’t get these crises.
* Recently, In Mumbai tomato prizes are 200/kg, rain ravage crops and price raises 400% extra.



Listening to the chatter at Delhi’s vegetable markets, only one question is on everyone’s lips: just how much will a tomato cost today?

Prices of tomatoes, a staple of Indian cooking, have soared by more than 400% in recent weeks as the country has been gripped by a nationwide shortage.

The shortage has been attributed to the irregular weather that has ravaged India during this year’s season for tomatoes, including unseasonable high rainfall in recent months, which devastated the growing crops and fuelled a deadly fungal disease.

While those in cities such as Mumbai or Delhi used to pay 40 rupees (40p) for a kilo of tomatoes, prices have shot up to 160 rupees and higher, making them largely unaffordable for an average low-income household.

Traders have warned prices could hit record highs of 200 rupees a kilo in the next few days as the recent heavy monsoon rains have spoiled more stock.

While July is often a more expensive season for buying tomatoes, as it falls between harvests, consumers said they had never seen prices so high.

The shortage has even hit outlets of the fast-food chain McDonald’s in India. In branches of McDonald’s across north, east and south India, signs were put up to state that tomatoes would no longer be put in burgers and other dishes, due to a lack of availability.

1.7 Outcome of our Idea:

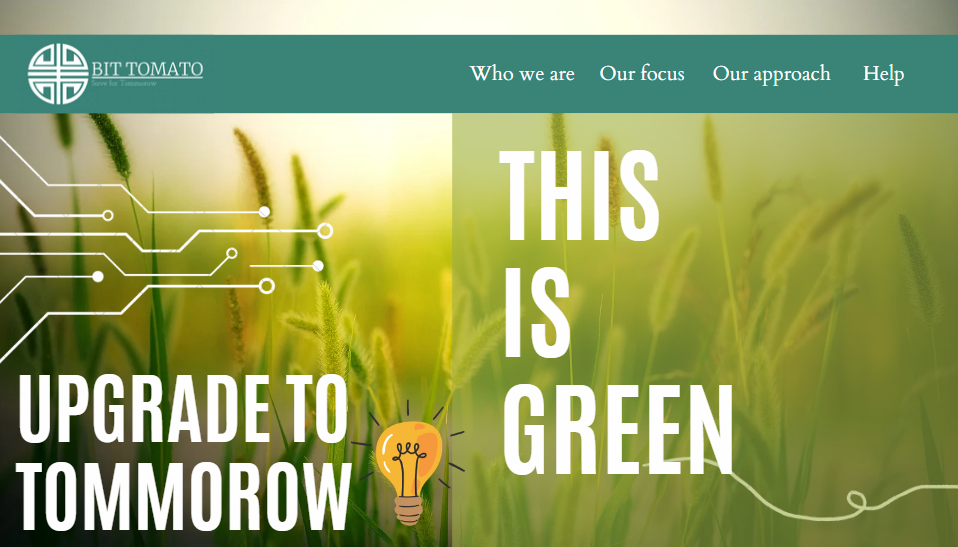
* There will be no tomato crises anymore.
* The wastage of tomatoes during pre-processing and post harvesting will be totally reduced.
* Less labour and low cost.
* Transportation wastage of tomatoes will be terminated because of using trays and harvesting tomatoes with respect to the distance travelled for transport.
* Diseases and disorder will be pre-diagnosed because of our cameras send a images and data of the tomato plants then Machine learning program will detect the disease in the crop.
* Automation irrigation and drying system.
* Automation harvesting using a robotic hands, depends upon a ripeness of the tomato.

1.8 Business Plans and Marketing on this Idea:

To make this idea into a business plan and market it. We can sell our ideas to other new agricultural startups and sell our products to them.



* For marketing, we create a website to buy our plans and products(Ex: Tomato trays, equipments). And we named a company as Bit-Tomato.
* In our website, customers can get a free consultation from our team members and can buy modern agricultural products



* It’s how the front page looks like in our website. We can achieve on social medias through this website upon our idea.
* If customers buy our plans we will directly go to their place and our team member will install the sensors and technology used in computers. So that the customer need not to know about our modern technology.
* The services of our installed mechanics also available on our company. The customer just need to call our company and we’ll provide a service person to it.



* The website is available in: [“Ctrl + Click here !”](https://suriyaopm.github.io/ress/) to view a website of our Bit-tomato.

1.9 Block Diagram:

