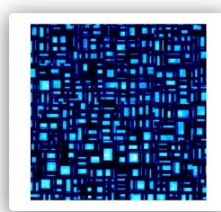


## Data Collection and Preprocessing Phase

Date	1 July 2024
Team ID	SWTID1720176710
Project Title	Visual Diagnostics: Detecting Tomato Plant Diseases Through Leaf Image Analysis
Maximum Marks	6 Marks

### Preprocessing Template

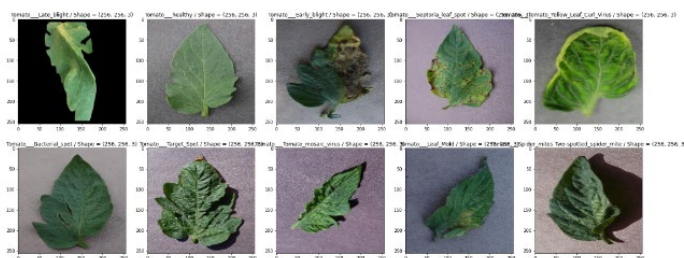
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	<p>Collect images of Tomato Leaves. Images are then organized into subdirectories based on their respective names as shown in the project structure.</p> <p>In this project, we have collected images of 10 types of Tomato Leaf images like Heatly, Spider Mites, Yellow leaf curl, etc. and they are saved in the respective sub directories with their respective names.</p> <p>You can download the dataset used in this project using the below link</p> <p>Dataset: <a href="https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf">https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf</a></p> <div data-bbox="636 1566 1380 1814">  <p>Tomato leaf disease detection   Kaggle..</p> <p>Tomato leaf disease detection using CNN.</p> <p><a href="https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf">https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf</a></p> </div>

## Resizing

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

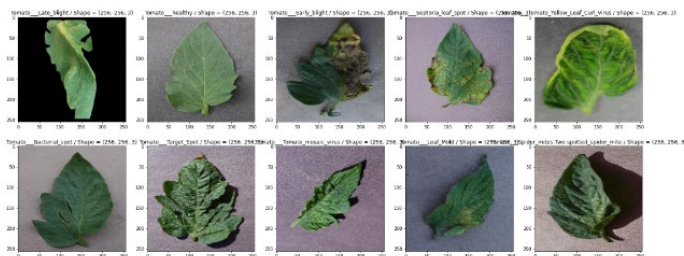
for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



## Normalization

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



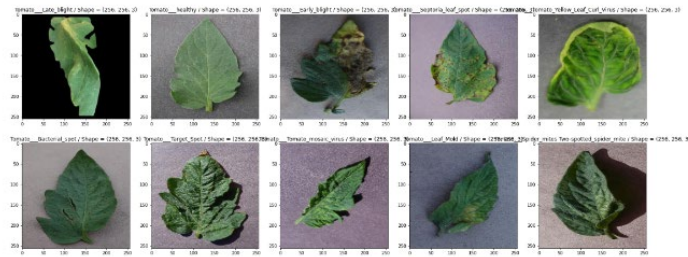
```
[6]: # Training data visualization

classes = os.listdir('./train')

plt.figure(figsize=(25,10))

for i in enumerate(classes):
    pic = os.listdir('./train/'+i)[0]
    image = Image.open('./train/'+i+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i*1+1)
    plt.title('{} / Shape = {}'.format(i[1], image.shape))
    plt.imshow(image)

plt.show()
```



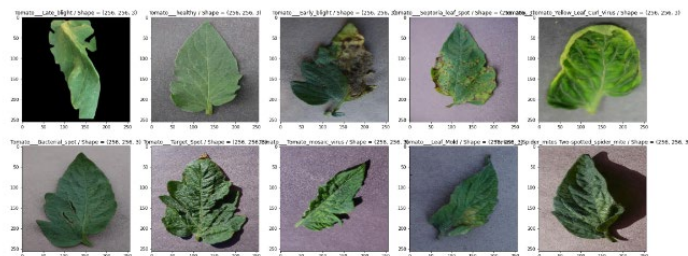
```
[6]: # Training data visualization

classes = os.listdir('./train')

plt.figure(figsize=(25,10))

for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = np.asarray(image.open('./train/'+i[1]+'/' +pic))
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{} / Shape = {}'.format(i[1], image.shape))
    plt.imshow(image)

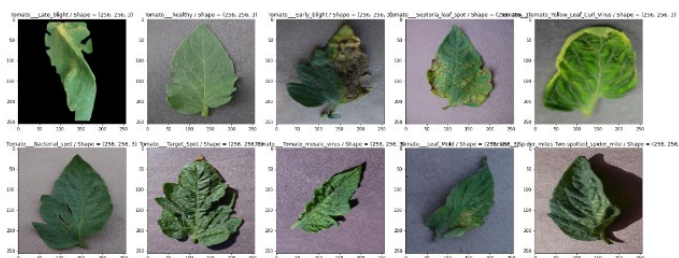
plt.show()
```



## Edge Detection

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

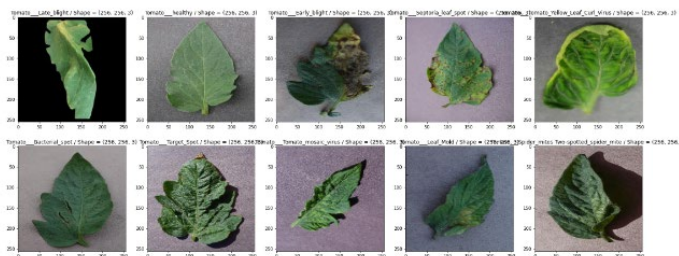
for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



## Color Space Conversion

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

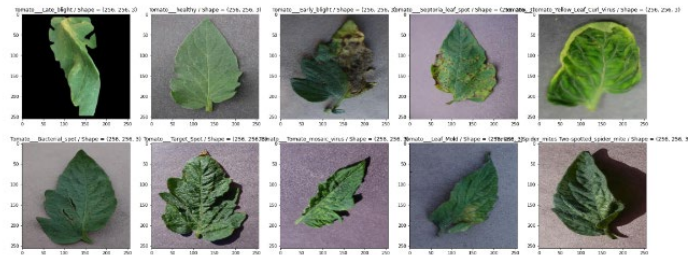
for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



## Image Cropping

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

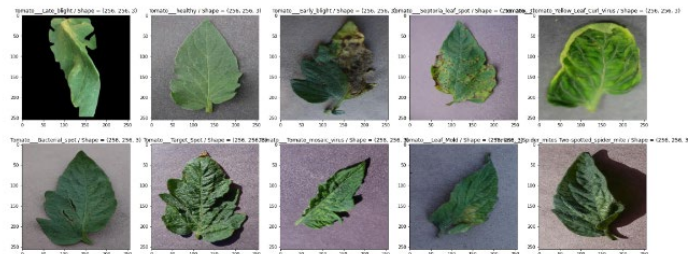
for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



## Batch Normalization

```
[6]: # Training data visualization
classes = os.listdir('./train')
plt.figure(figsize=(25,10))

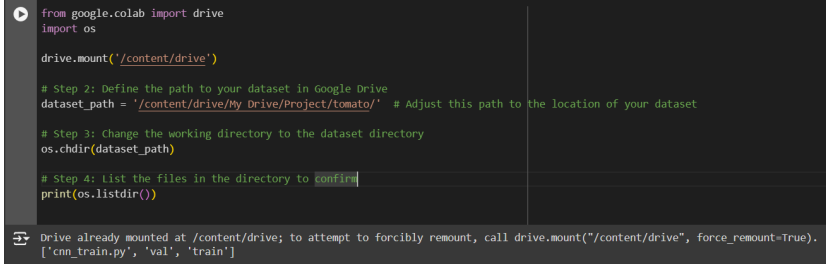
for i in enumerate(classes):
    pic = os.listdir('./train/'+i[1])[0]
    image = Image.open('./train/'+i[1]+'/' +pic)
    image = np.asarray(image)
    plt.subplot(2,5,i[0]+1)
    plt.title('{0} / Shape = {1}'.format(i[1], image.shape))
    plt.imshow(image)
plt.show()
```



## Data Preprocessing Code Screenshots

### Loading Data

```
from google.colab import drive
import os
drive.mount('/content/drive')
dataset_path = '/content/drive/My Drive/Project/tomato/'
os.chdir(dataset_path)
print(os.listdir())
```

	 <pre> from google.colab import drive import os  drive.mount('/content/drive')  # Step 2: Define the path to your dataset in Google Drive dataset_path = '/content/drive/My Drive/Project/tomato/' # Adjust this path to the location of your dataset  # Step 3: Change the working directory to the dataset directory os.chdir(dataset_path)  # Step 4: List the files in the directory to confirm print(os.listdir())  Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True). ['cnn_train.py', 'val', 'train'] </pre>
Resizing	Give the code snippet as an image (copy and paste the picture in this block).
Normalization	Give the code snippet as an image (copy and paste the picture in this block).
Data Augmentation	Give the code snippet as an image (copy and paste the picture in this block).
Denoising	Give the code snippet as an image (copy and paste the picture in this block).
Edge Detection	Give the code snippet as an image (copy and paste the picture in this block).
Color Space Conversion	Give the code snippet as an image (copy and paste the picture in this block).
Image Cropping	Give the code snippet as an image (copy and paste the picture in this block).
Batch Normalization	Give the code snippet as an image (copy and paste the picture in this block).