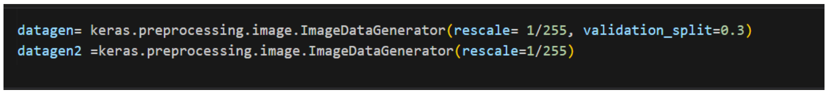
Data Collection and Preprocessing Phase

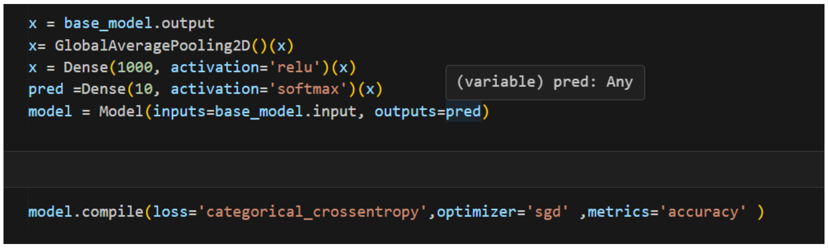
Date	1 July 2024
Team ID	SWTID1720176710
Project Title	Visual Diagnostics: Detecting Tomato Plant Diseases Through Leaf Image Analysis
Maximum Marks	6 Marks

Preprocessing Template

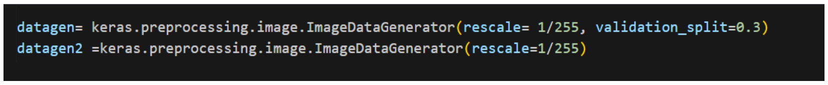
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network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

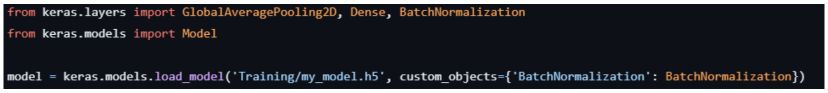
Section	Description
Data Overview	Collect images of Tomato Leaves. Images are then organized into subdirectories based on their respective names as shown in the project structure. In this project, we have collected images of 10 types of Tomato Leaf images like Heatly, Spider Mites, Yellow leaf curl,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

Resizing	
	datagen= keras.preprocessing.image.ImageDataGenerator(rescale= 1/255, validation_split=0.3)
	datagen2 =keras.preprocessing.image.ImageDataGenerator(rescale=1/255)

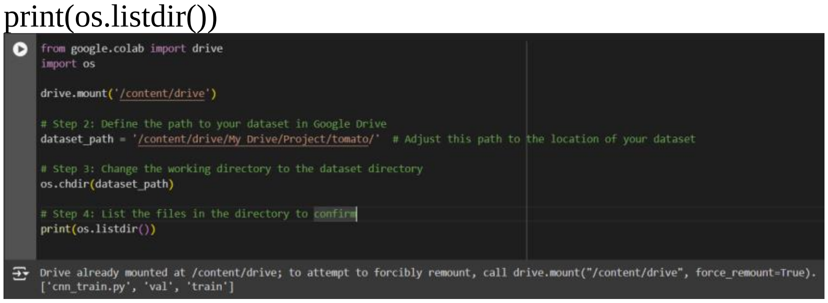
Edge Detection	
	x = base_model.output
	x= GlobalAveragePooling2D()(x)
	x = Dense(1000, activation='relu')(x)
	pred =Dense(10, activation='softmax')(x)
	model = Model(inputs=base_model.input, outputs=pred)
	(variable) pred: Any
	model.compile(loss='categorical_crossentropy',optimizer='sgd' ,metrics='accuracy')

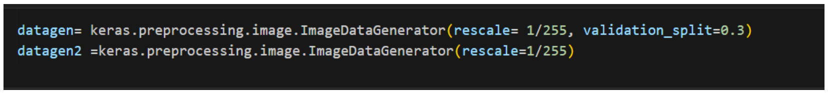
Color Space Conversion	No need

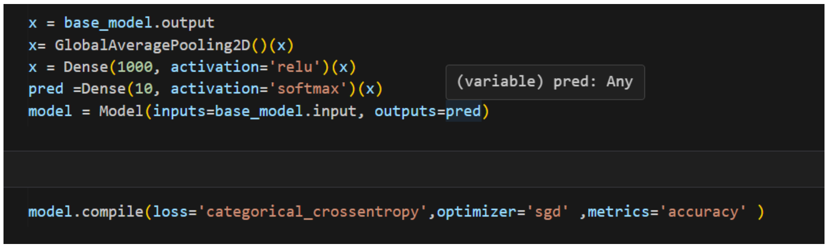
Image Cropping	
	datagen= keras.preprocessing.image.ImageDataGenerator(rescale= 1/255, validation_split=0.3)
	datagen2 =keras.preprocessing.image.ImageDataGenerator(rescale=1/255)

Batch Normalization	
	from keras.layers import GlobalAveragePooling2D, Dense, BatchNormalization
	from keras.models import Model
	model = keras.models.load_model('Training/my_model.h5', custom_objects={'BatchNormalization': BatchNormalization})

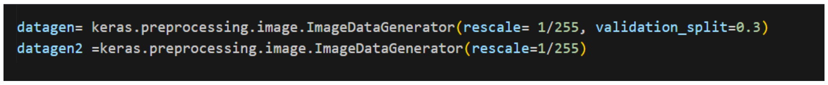
Data Preprocessing Code Screenshots	

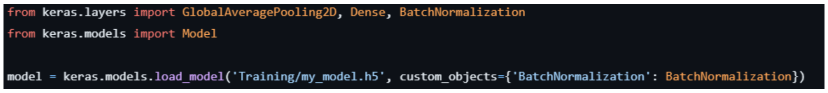
Loading Data	from google.colab import drive
	import os
	drive.mount('/content/drive')
	dataset_path = '/content/drive/My Drive/Project/tomato/'
	os.chdir(dataset_path)
	print(os.listdir())
	
	from google.colab import drive
	import os
	drive.mount('/content/drive')
	# Step 2: Define the path to your dataset 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']

Resizing	
	datagen= keras.preprocessing.image.ImageDataGenerator(rescale= 1/255, validation_split=0.3)
	datagen2 =keras.preprocessing.image.ImageDataGenerator(rescale=1/255)

Edge Detection	
	`x = base_model.output`
	`x= GlobalAveragePooling2D()(x)`
	`x = Dense(1000, activation='relu')(x)`
	`pred =Dense(10, activation='softmax')(x)`
	`model = Model(inputs=base_model.input, outputs=pred)`
	
	`model.compile(loss='categorical_crossentropy',optimizer='sgd' ,metrics='accuracy' )`

Color Space Conversion	No need

Image Cropping	
	`datagen= keras.preprocessing.image.ImageDataGenerator(rescale= 1/255, validation_split=0.3)`
	`datagen2 =keras.preprocessing.image.ImageDataGenerator(rescale=1/255)`

Batch Normalization	
	`from keras.layers import GlobalAveragePooling2D, Dense, BatchNormalization`
	`from keras.models import Model`
	`model = keras.models.load_model('Training/my_model.h5', custom_objects={'BatchNormalization': BatchNormalization})`