Introduction

- Plant Disease Detection in Image Segmentation is aimed at identifying and diagnosing diseases in crops by analyzing images of plant leaves.
- This combines computer vision and image processing techniques to assist farmers and researchers in monitoring and managing crop health efficiently.

About the project

- Images of tomato leaves are processed and segmented using CNN.
- Image segmentation divides an image into distinct regions or segments, making it possible to isolate specific areas of interest.
- For plant disease detection, this segmentation is crucial for precisely identifying and locating regions of leaves affected by diseases or pests

About the Dataset

- Plant village dataset acquired from Kaggle has been used in the project.
- The data set contains 16,012 images of diseased tomato leaves.
- The dataset has ten labels i.e. different types of leaf diseases, for example, Target spot, Mosaic virus.

Preprocessing

- Preprocessing is a crucial step in the image analysis pipeline for plant disease detection in image segmentation. Its primary purpose is to prepare the acquired images by enhancing their quality, reducing noise, and making them more suitable for subsequent analysis.

- **Converting to Grayscale:** The code converts the RGB images to grayscale using cv2.cvtColor. Grayscale images have a single channel (intensity), which simplifies processing and reduces computational complexity.

Preprocessing

- **Histogram Equalization :**

The code checks if the image is in grayscale format (i.e., it has only one channel). If it is, histogram equalization is applied using cv2.equalizeHist(). Histogram equalization enhances the contrast of the grayscale image, making it easier to detect features.

- **Pixel Value Normalization:**

After processing, the code normalizes the pixel values of the image to be between 0 and 1. This is often done by dividing each pixel value by 255.0, as pixel values typically range from 0 to 255 in grayscale images.

Preprocessing

- **Resizing the Image:**

The code resizes the image to a fixed size of 32x32 pixels using cv2.resize(). This resizing ensures that all images have the same dimensions, which is often required for machine learning models.

Finally, the preprocessed image is added to the images list associated with its respective directory. This list will later be used for building a dataset for training or evaluation

Convolutional Neural Network (CNN)

- Convolutional layers (Conv2D): These layers apply convolution operations to extract features from the images. Different convolutional layers with varying numbers of filters are used to capture hierarchical features in the images. The 'relu' activation function is applied to introduce non-linearity, and 'same' padding maintains the spatial dimensions.

- Max-pooling layers (MaxPooling2D): These layers downsample the feature maps to reduce the computational load and improve translation invariance. They are interspersed between convolutional layers.

Convolutional Neural Network (CNN)

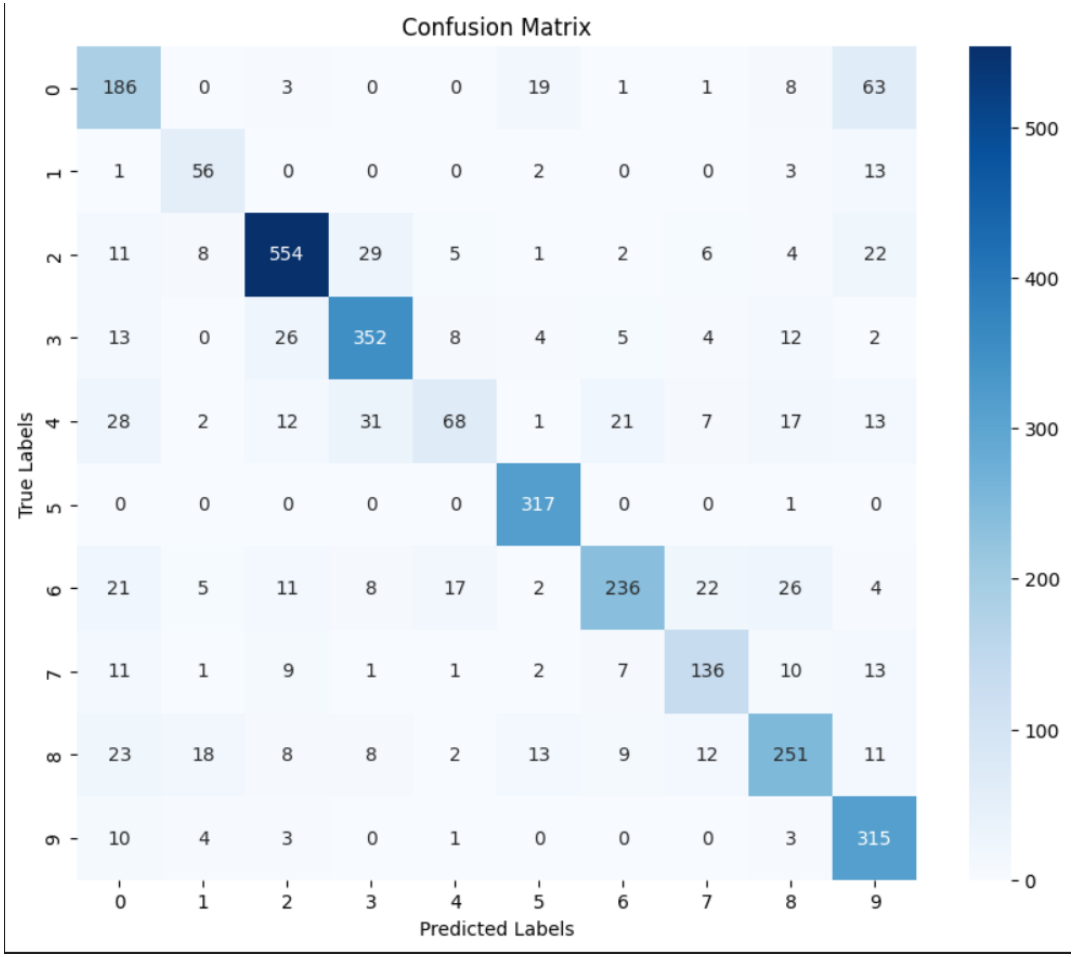
- Dropout layers (Dropout): Dropout is applied to regularize the model and prevent overfitting by randomly deactivating a fraction of neurons during training.
- Fully connected layers (Dense): After feature extraction, the flattened feature maps are fed into fully connected layers to make classification decisions.
- The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss, which is suitable for multi-class classification problems. The chosen metrics for evaluation are accuracy.

Exploratory Data Analysis

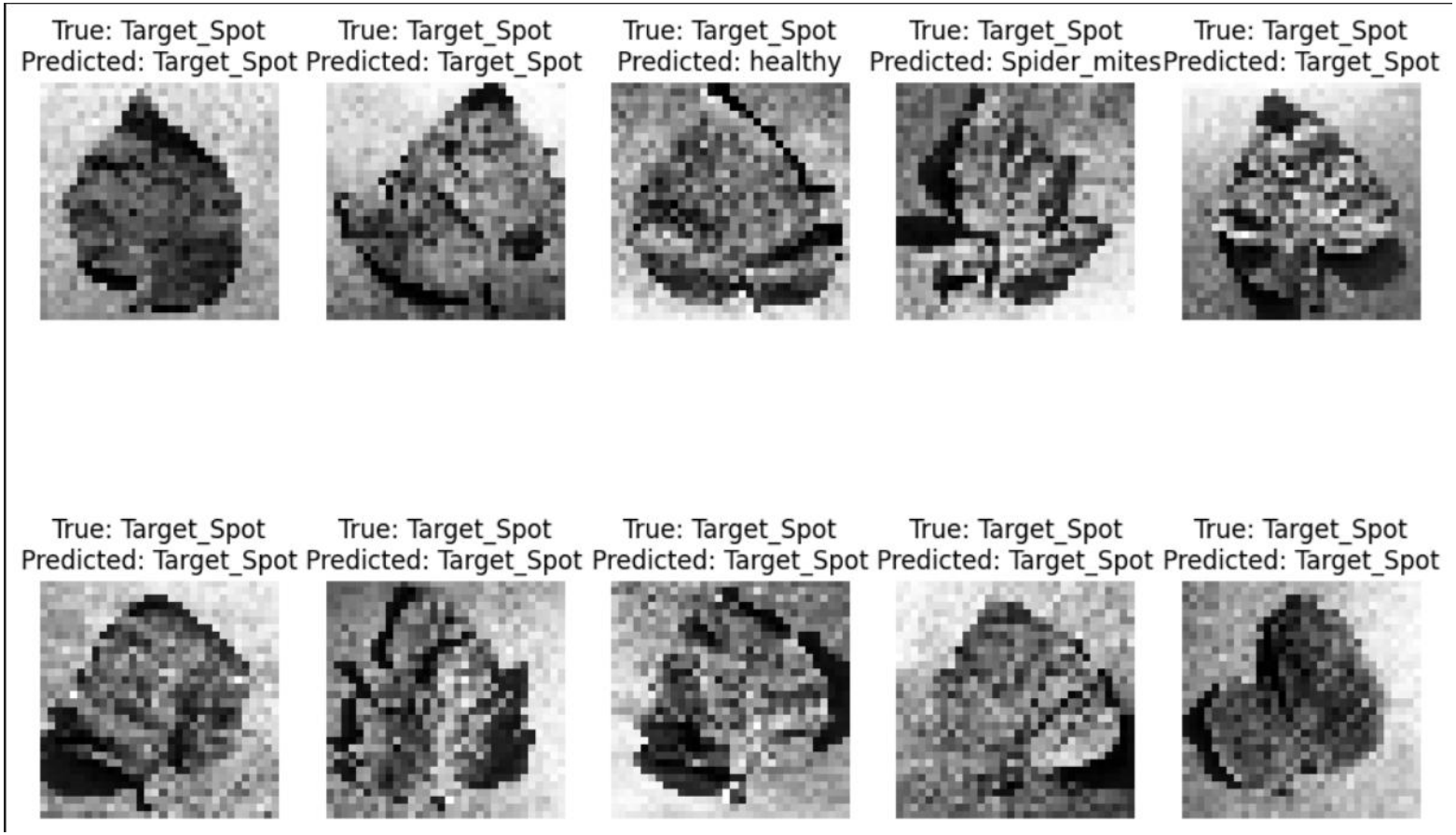
- After training the convolutional neural network (CNN) model and evaluating it on the test data, next we generated a confusion matrix to gain insights into the model's performance.

- A confusion matrix is a useful tool for understanding how well the model is classifying different classes and where it might be making errors.

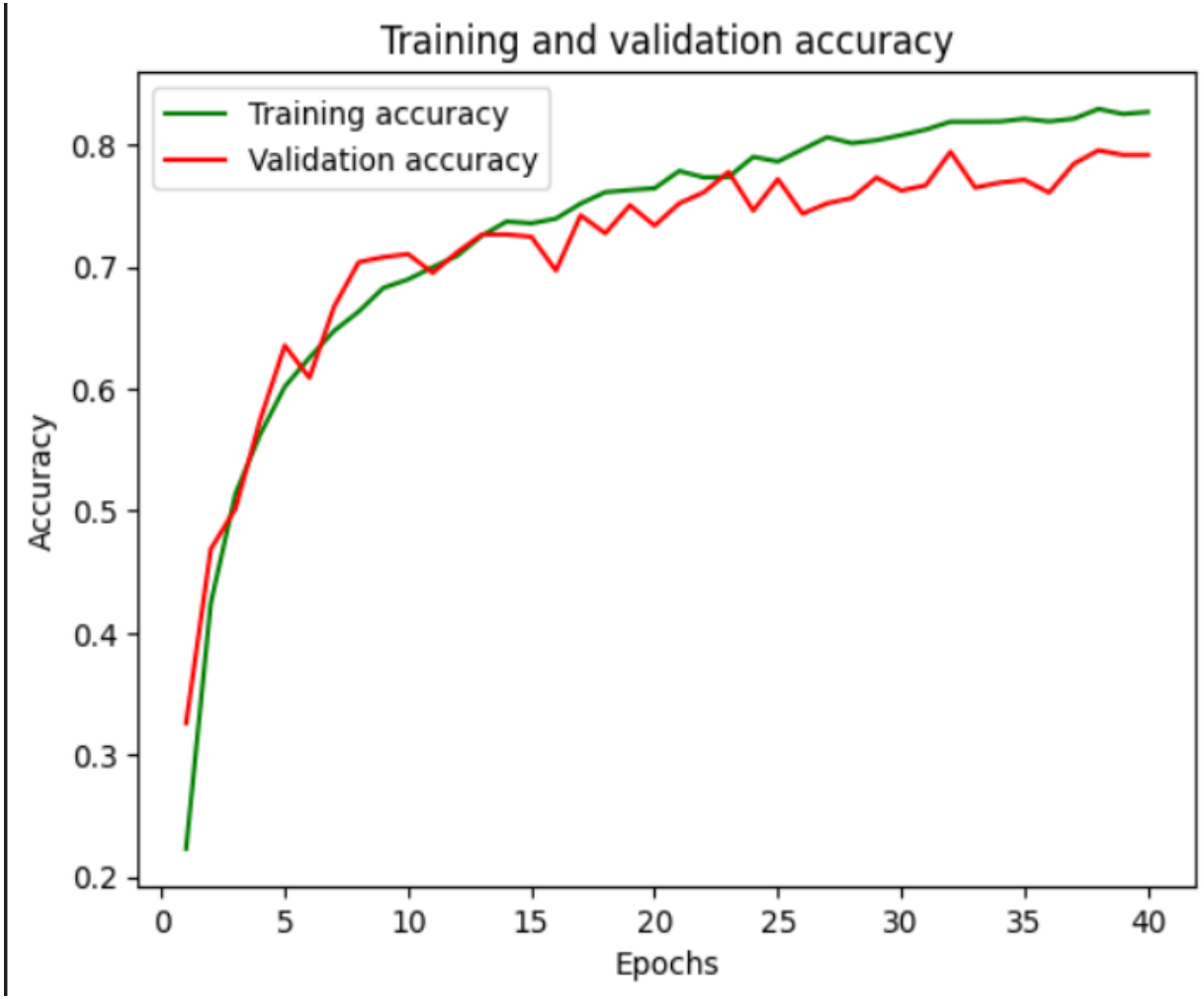
Confusion Matrix Visualization for Image Classification



Visualization of Predicted vs. True Labels for Plant Disease Detection



Training and Validation Accuracy Visualization



Conclusion

- In conclusion, the Plant Disease Detection in Image Segmentation project combines computer vision, image processing, and deep learning techniques to aid farmers and researchers in efficiently monitoring and managing crop health. Using a Convolutional Neural Network (CNN) on the Plant Village dataset from Kaggle, the project successfully identifies and diagnoses diseases in tomato leaves, crucially relying on image segmentation to precisely locate affected regions.

- The project achieves promising results, and the use of a confusion matrix helps evaluate the model's classification performance effectively. Overall, this project contributes to improved crop disease detection and agricultural sustainability.

Thank You