CONVOLUTIONAL NETWORKS

AND

TRANSFER LEARNING

- **ResNet50V2**
- **ResNet101V2**
- **VGG16**
- **VGG19**

DATASET: LEAF (WHITE SPOT, ALGAL, BROWN BLIGHT)

Sachith M. Gunawardane

Table of Contents

1. Summary

The ultimate iteration, Final Model 02, utilizing ResNet50V2 (without training weights in the pre-trained network), achieved a remarkable validation accuracy of **0.8845** and test accuracy of **0.8327** surpassing the validation accuracy of 0.7570 or test accuracy of 0.7490 attained by the manually designed Model 01 from scratch. The progression through this process unfolded as follows:

1. Dataset loading took place without data augmentation, accompanied by visual inspection. The training set comprised 296 images, the validation set included 72 images, and the testing set consisted of 368 images (total number of images). Due to limitation of data, for testing full data set has been used. This is not a good practise and does not provide accurate picture. However, this issue has been addressed in augmented dataset which is used to report final values.
2. Model 01 was built from scratch, employing Keras Tuner to optimize parameters such as the number of convolutional/pooling layers, fully connected (FC) layers, dropout rates, nodes in each layer, and the training rate.
3. Model 02 was devised using various pre-trained networks (ResNet50V2, ResNet101V2, VGG16, and VGG19) to extract features, with a subsequent critical evaluation of results to identify the most suitable pre-trained network for the given problem.
4. The FC layers of Model 02, employing the best pre-trained architecture identified in step 3, were redesigned using Keras Tuner. The performance was then compared with that of Model 01.
5. Novel data for training and validation was generated through image augmentation, and the augmented data was visually inspected. The augmented dataset comprised 1171 training images, 251 validation images, and 251 testing images. Total 1673 images.
6. The best models from steps 2 (Model 01) and 4 (Model 02) were trained on the augmented data.

Key Insights:

- Transfer learning with pre-trained networks significantly enhances feature extraction, facilitating the rapid design of machine learning models with higher accuracy.
- The quantity of images in the training dataset holds critical importance for model performance, particularly in terms of generalization. In computer vision, employing image data augmentation and pre-processing techniques proves instrumental in creating a more extensive and diverse training dataset.

Refer .ipynb to more details and explanation for each step.

2. Data Visualization

The given dataset contains images for 3 classes (algal leaf, brown blight, white spot) which are diseases of tea leaf. Total number of images in this dataset is 368 having around 100+ images in each class. I have decided initially to use data as-is without augmentation or upscaling. Training and validation are having 80% and 20% data split. Hence training got 296 images, validation with 72 images and testing is full set of images 368.

All models use same image size which is width 224 and height 224.

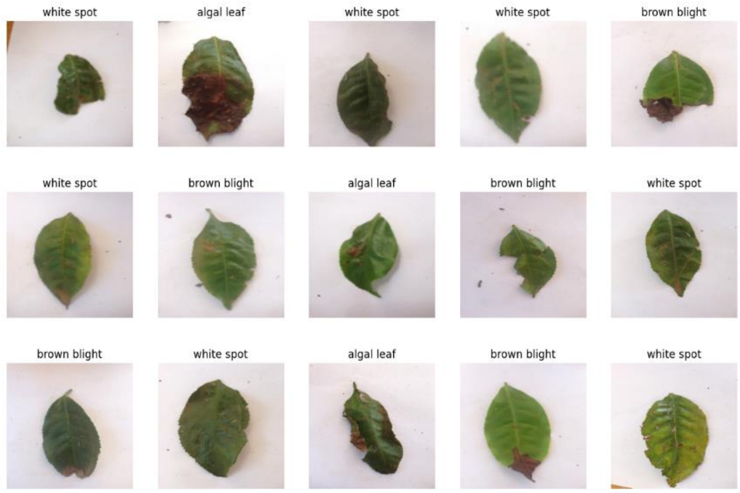

Figure 2-1 - Original Dataset

3. Model 01

The model 01 is CNN model from scratch. Here I have use Keras tuner to design optimal model. Keras tuner will optimise,

- No of Convolutional and Pooling layers
- Convolutional layer filters, kernel size
- Pooling layer, pooling size
- No of Fully Connected (FC) layers
- Nodes in FC layer
- Percentage of dropout
- Learning rate

Hyperparameters	Best Model

```
conv_1_filter: 128
conv_1_kernel: 3
pool_1_size: 2
n_conv_layers: 3
conv_2_filter: 128
conv_2_kernel: 5
pool_2_size: 2
n_FC_layers: 1
1_FC_layer: 64
dropout_layer: 0.2
learning_rate: 0.0001
conv_3_filter: 256
conv_3_kernel: 5
pool_3_size: 2
conv_4_filter: 128
conv_4_kernel: 3
pool_4_size: 2
2_FC_layer: 64
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 128)     3584

 max_pooling2d (MaxPooling2  (None, 111, 111, 128)     0
 D)

 conv2d_1 (Conv2D)           (None, 107, 107, 128)     409728

 max_pooling2d_1 (MaxPoolin  (None, 53, 53, 128)       0
 g2D)

 conv2d_2 (Conv2D)           (None, 49, 49, 256)       819456

 max_pooling2d_2 (MaxPoolin  (None, 24, 24, 256)       0
 g2D)

 conv2d_3 (Conv2D)           (None, 22, 22, 128)       295040

 max_pooling2d_3 (MaxPoolin  (None, 11, 11, 128)       0
 g2D)

 flatten (Flatten)           (None, 15488)             0

 dense (Dense)               (None, 64)                991296

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 2519299 (9.61 MB)
Trainable params: 2519299 (9.61 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3-1 - Model 1 Architecture

Keras Tuner used Random Search with reducing validation loss as its objective. At end of optimisation Keras Tuner provide following model with best validation loss of 0.9380.

Model accuracy and loss as follows,

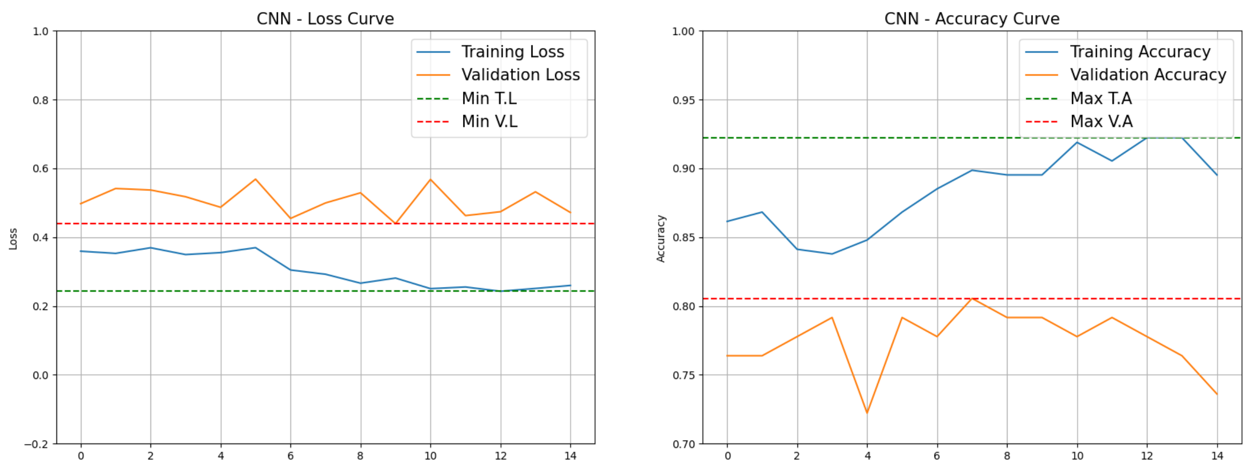

Figure 3-2 - Model 01 Accuracy & Loss

Conclusion - Model 01

Model 01, a Convolutional Neural Network (CNN) classification model crafted from the ground up to predict three distinct tea leaf diseases, exhibits a designed architecture yielding a validation accuracy of 0.79 and a test accuracy of 0.90. Notably, throughout the training process, Model 01 attained a commendable test accuracy of 0.95 but encountered difficulty surpassing a validation accuracy slightly above 0.80. This observed disparity between training and validation accuracy underscores challenges related to the model's generalization capabilities for the provided dataset.

4. Model 02

In this segment, I intend to train the CNN model utilizing the subsequent pre-trained networks for feature extraction:

- ResNet50V2
- ResNet101V2
- VGG16
- VGG19

The design of the fully connected (FC) layers mirrors that of Model 01 (absent a pre-trained network). This approach ensures a like-to-like comparison of the capabilities of pre-trained networks. Additionally, the training rate employed is identical to that of Model 01.

```
Model Summary

Model: "Resnet50V2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50v2 (Functional)     (None, 7, 7, 2048)        23564800

 flatten_2 (Flatten)         (None, 100352)            0

 dense_4 (Dense)             (None, 64)                6422592

 dropout_2 (Dropout)         (None, 64)                0

 dense_5 (Dense)             (None, 3)                 195

=================================================================
Total params: 29987587 (114.39 MB)
Trainable params: 6422787 (24.50 MB)
Non-trainable params: 23564800 (89.89 MB)
_________________________________________________________________
```

Figure 4-1 - Model 2 Architecture for ResNet50V2

All four models undergo evaluation using the worst-model exclusion strategy to determine the optimal architecture for a pre-trained model. The depicted model architecture pertains to ResNet50V2. ResNet50V2 demonstrated superior performance when compared to all other pre-trained networks. The image below illustrates a comparison between the accuracy and loss of ResNet50V2 and ResNet101V2.

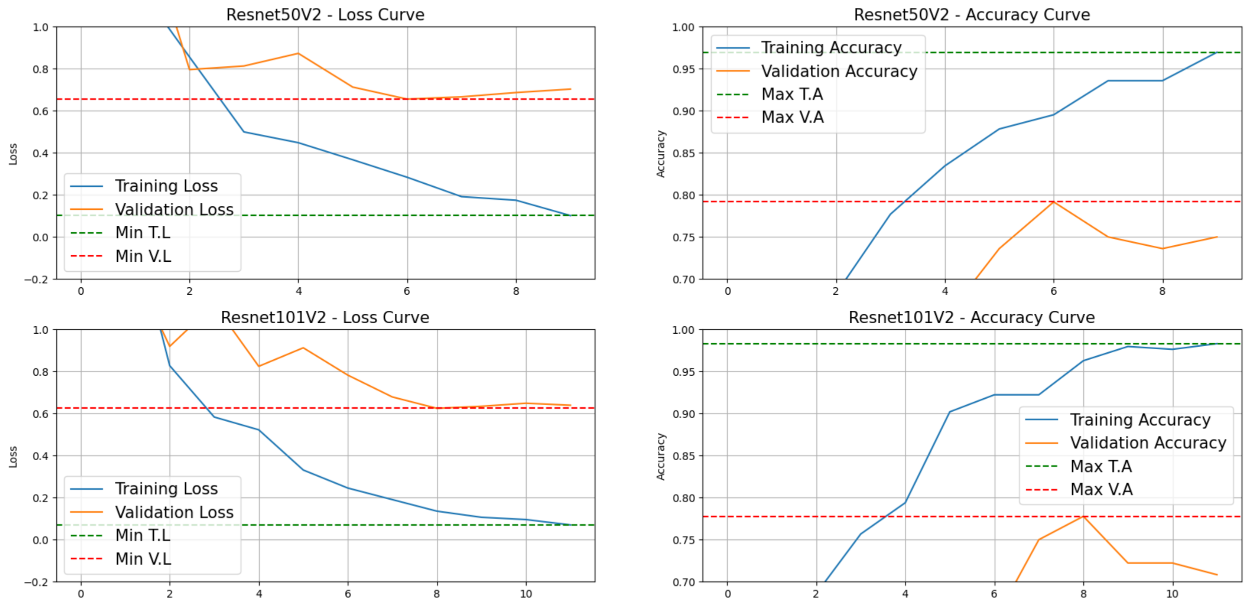

Figure 4-2 - Model 2 Accuracy and Loss for ResNet

Upon a meticulous examination of loss and accuracy metrics between ResNet50V2 and ResNet101V2, it becomes evident that ResNet50V2 outperforms ResNet101V2 on the utilized dataset, as indicated by a best loss of 0.6557 and a best accuracy of 79.2%.

This experiment involved solely the replacement of the feature extraction layer with transfer learning, without training any weights, while maintaining fully connected layers akin to Model 01 (CNN from scratch). For Model 01, constructed from scratch, the loss stands at 0.4394 with an accuracy of 0.7916. In contrast, Model 02, employing transfer learning, exhibits a loss of 0.6557 and an accuracy of 0.792. Surprisingly, despite Model 02 boasting a more intricate architecture with heightened feature extraction capabilities, Model 01 appears to outperform it.

This discrepancy in performance could potentially be attributed to the fully connected (FC) layer utilized in Model 02. The FC layer was fine-tuned using Keras Tuner for Model 01. To validate this hypothesis, the next step involves redesigning the FC layer of Model 02 using Keras Tuner.

4.1 Model 02 - Redesign

Following table display tunned hyperparameters and architecture for model 02. Further it’s accuracy and loss.

Hyperparameters for FC	Best Model

```
n_layers: 4
1_units: 256
rate: 0.2
learning_rate: 0.0001
2_units: 256
3_units: 512
4_units: 256
```

```
Model: "ResNet50V2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50v2 (Functional)     (None, 7, 7, 2048)        23564800

 flatten (Flatten)           (None, 100352)            0

 dense (Dense)               (None, 256)               25690368

 dense_1 (Dense)             (None, 256)               65792

 dense_2 (Dense)             (None, 512)               131584

 dense_3 (Dense)             (None, 256)               131328

 dropout (Dropout)           (None, 256)               0

 dense_4 (Dense)             (None, 3)                 771

=================================================================
Total params: 49584643 (189.15 MB)
Trainable params: 26019843 (99.26 MB)
Non-trainable params: 23564800 (89.89 MB)
_________________________________________________________________
```

Figure 4.1-1 - Model 2 Redesign Architecture

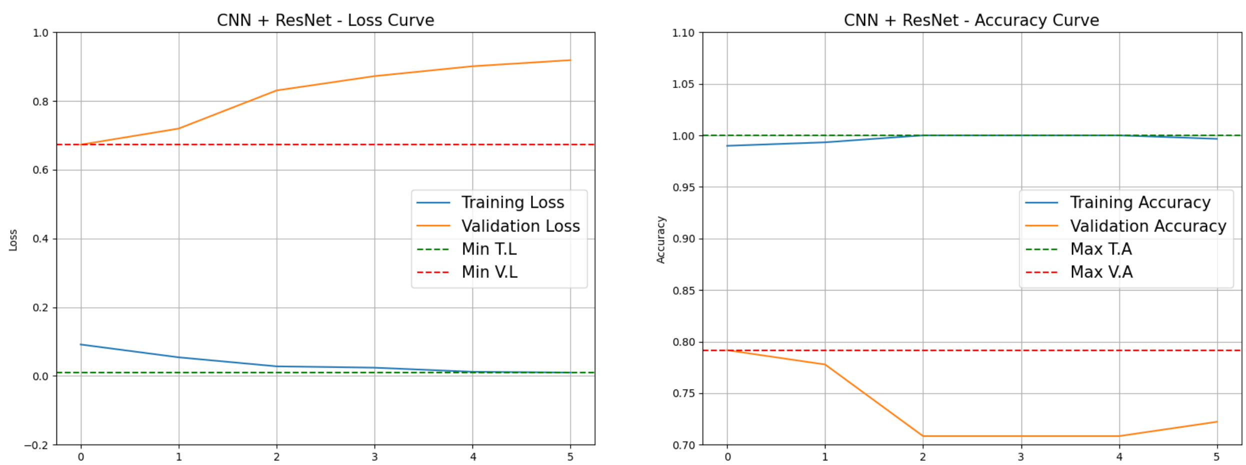

Figure 5-1 - Model 2 Redesign Architecture

Conclusion - Model 02 Optimization

After optimizing FC layers in Model 02, a notable improvement was observed, resulting in an accuracy of 0.791. The key takeaways from this optimization are as follows:

1. Model 01 and Model 02 exhibited comparable performance, indicating that the specific architectures no longer significantly impact accuracy with the current dataset.
2. Despite achieving 100% accuracy during training, both models struggled to attain higher accuracy on the validation set, suggesting challenges in generalizing the prediction function.

To enhance generalization, the following strategies can be employed:

- Utilizing simpler model architectures.
- Increasing the volume of training data.

The analysis highlights a potential issue with generalization, pointing towards a shortage of training data (300+ samples with each class having 100 samples). Consequently, the next step involves augmenting the training data through techniques such as image augmentation.

5 Data Augmentation

Using image augmentation, I have increase images in each class. After augmentation counts as follows,

- Algal leaf – 654 images
- Brown blight – 661 images
- White spots - 826 images

With 70%, 15% and 15% split between training, validation and test data brings to 1171 images in training 251 in validation and 251 in test . Data visualization for augmented dataset as follows:

Figure 5-1 – Augmented Dataset

5.1 Model 01

The Model 01 accuracy and loss with augmented data is as follows:

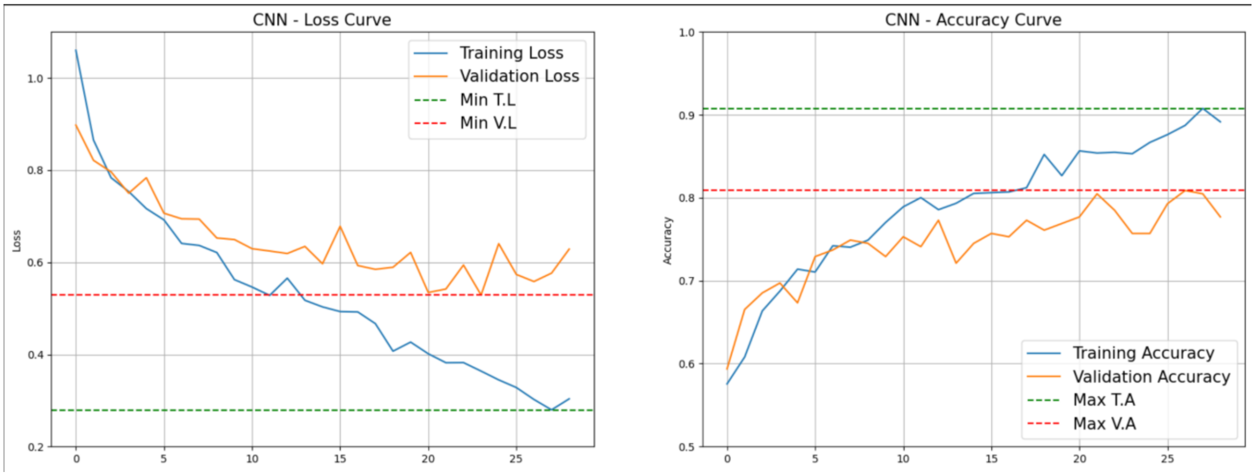

Figure 5.1-2 – Model 01 Accuracy and Loss

5.2 Model 02

The Model 02 accuracy and loss with augmented data is as follows:

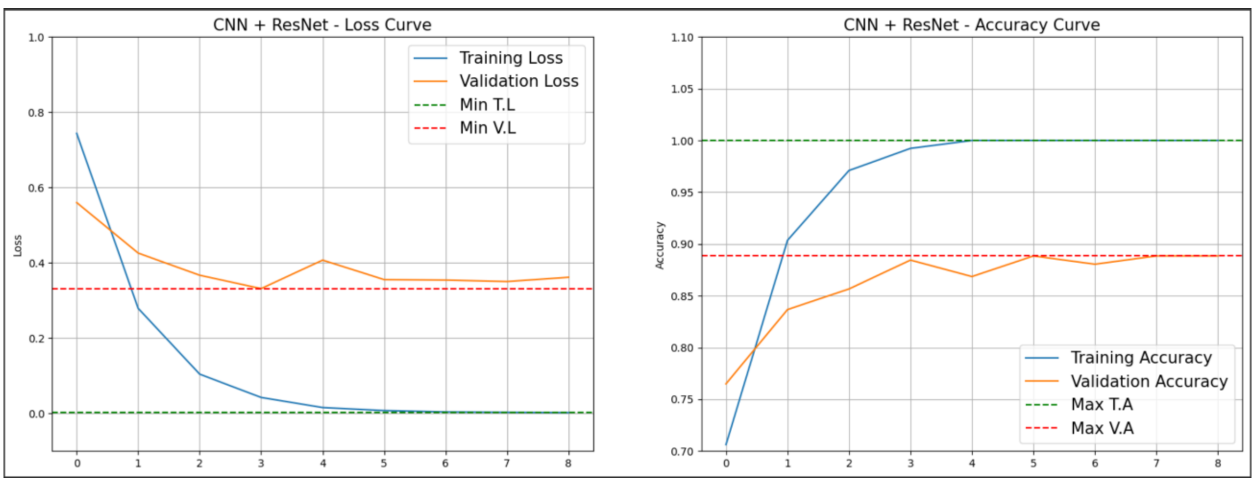

Figure 5.2-3 – Model 02 Accuracy and Loss

6 Conclusion

As anticipated, the inclusion of more data proved beneficial in narrowing the gap between training and validation accuracy.

For Model 01, the accuracy declined from 0.7916 to 0.7570. However, this validation accuracy encompasses both training and testing (0.7490) accuracies, indicating potential enhancements in model architectures for superior outcomes.

This notion is substantiated by the performance of Model 02, which achieved a validation accuracy of **0.8845** and a testing accuracy of 0.8327. Leveraging the ResNet50V2 feature extraction capabilities, characterized by its intricate architecture, and utilizing improved datasets, Model 02 outperformed all trained models.

In conclusion, this exercise underscores two key insights:

1. Transfer learning plays a pivotal role in feature extraction, enabling the rapid development of machine learning models with heightened accuracy.
2. The volume of images in the training dataset significantly influences model performance, particularly concerning generalization. In the realm of computer vision, employing techniques such as image data augmentation and pre-processing proves instrumental in expanding and diversifying the training dataset.

7 TensorBoard