

# EfficientNet-B3 Model Documentation

This document presents a detailed documentation of the image classification model based on EfficientNet-B3. The focus is on the model architecture, training strategy, evaluation results, and interpretation, without discussing data preprocessing steps.

## 1. Model Overview

EfficientNet-B3 is a convolutional neural network architecture that scales depth, width, and input resolution in a balanced manner using compound scaling. It achieves high accuracy with fewer parameters compared to traditional CNN architectures.

## 2. Training Process

The model was trained for multiple epochs using transfer learning. Initially, the base layers were frozen, and later fine-tuning was applied by unfreezing part of the network. The best model was selected based on validation accuracy.

```
Starting training for 20 epochs
Epoch 1/20
Train - Loss: 1.3776, Acc: 76.00%
Val - Loss: 1.2080, Acc: 81.30%
Learning Rate: 0.000000
Best model saved (Accuracy: 81.30%)
Epoch 2/20
Train - Loss: 1.2048, Acc: 88.97%
Val - Loss: 1.2080, Acc: 83.91%
Learning Rate: 0.000000
Best model saved (Accuracy: 83.91%)
Epoch 3/20
Train - Loss: 1.2130, Acc: 82.41%
Val - Loss: 1.2130, Acc: 84.71%
Learning Rate: 0.000000
Best model saved (Accuracy: 84.71%)
Epoch 4/20
Train - Loss: 1.1937, Acc: 82.96%
Val - Loss: 1.2080, Acc: 85.00%
Learning Rate: 0.000000
Best model saved (Accuracy: 84.96%)
Epoch 5/20
Train - Loss: 1.1729, Acc: 83.82%
Val - Loss: 1.2080, Acc: 85.48%
Learning Rate: 0.000000
Best model saved (Accuracy: 85.48%)
Epoch 6/20
Unfreezing 50% of layers...
Train - Loss: 0.9544, Acc: 93.34%
Val - Loss: 0.8111, Acc: 93.88%
Learning Rate: 0.000000
Best model saved (Accuracy: 93.88%)
Epoch 7/20
Train - Loss: 0.7961, Acc: 94.85%
Val - Loss: 0.8109, Acc: 93.51%
Learning Rate: 0.000000
Epoch 8/20
Train - Loss: 0.7779, Acc: 97.12%
Val - Loss: 0.8432, Acc: 94.67%
```

## 3. Training and Validation Curves

The following plots illustrate the training and validation loss, accuracy, and learning rate schedule across epochs.

The screenshot shows a Jupyter Notebook interface with three plots: Loss Curves, Accuracy Curves, and Learning Rate Schedule. The Loss Curves plot shows Train Loss (blue) decreasing from ~1.4 to ~0.7 and Validation Loss (orange) decreasing from ~1.3 to ~0.8 over 18 epochs. The Accuracy Curves plot shows Train Accuracy (blue) increasing from ~80% to ~98% and Validation Accuracy (orange) increasing from ~85% to ~96% over 18 epochs. The Learning Rate Schedule plot shows a constant learning rate of approximately 1e-4 across all 18 epochs.

```

# EfficientNet B3.jupyter
# Best Validation Accuracy: 99.87% (at epoch 17)
# Corresponding Train Accuracy: history['train_acc'][best_epoch-1], 2d%]
# Total Epochs Trained: len(history['train_loss']) / 15

# Test Training Results:
print("Best Training Results:")
print(f"Best Validation Accuracy: {history['val_acc'][best_epoch]} (% at epoch {best_epoch})")
print(f"Corresponding Train Accuracy: {history['train_acc'][best_epoch-1]}, {2d}%")
print(f"Total Epochs Trained: {len(history['train_loss'])} / 15")

# Load the best model
print("Loading the best model...")
checkpoint = torch.load(test_model_path, map_location=device)
model.load_state_dict(checkpoint['model'])
model.eval()

print("Model loaded from epoch: ", checkpoint['epoch'])
print(f"Tested validation accuracy: {checkpoint['val_acc']}, {2d}%")

# Test the model on Test Set
print("Testing model on Test Set...")

# Variables Terminal 5:58 AM T4 (Python 3)

```

## 4. Evaluation Results

The trained model was evaluated on a held-out test dataset. The results show high generalization performance, achieving strong accuracy and low loss values.

The screenshot shows a Jupyter Notebook interface with a classification report and a confusion matrix. The classification report table includes columns: precision, recall, f1-score, and support for various fruit classes. Most classes have high precision, recall, and f1-score values near 1.0. The confusion matrix shows the overall accuracy of 98.98% and a weighted average of 98.98%.

	precision	recall	f1-score	support
Apple___Bald_star	0.991	1.000	0.995	109
Apple___Black_rust	0.990	1.000	0.995	97
Apple___Coral_infection	0.990	1.000	0.995	23
Apple___Healthy	1.000	1.000	1.000	255
Apple___Rotten	1.000	1.000	1.000	255
Cherry_(including_sour)___Powered_mildew	1.000	1.000	1.000	153
Cherry_(including_sour)___Unpowered_mildew	0.990	1.000	0.995	242
Chestnut___Leaf_spots	0.950	0.963	0.979	71
Chestnut___Northern_leaf_light	0.990	0.970	0.980	129
Corn_(maize)___Crosses_leaves	0.999	0.970	0.977	136
Corn_(maize)___Northern_leaf_light	0.989	0.960	0.974	124
Corn_(maize)___Northern_leaf_healthy	0.989	0.960	0.974	106
Grape___Black_Mold	0.991	0.995	0.993	215
Grape___Leaf_spots	0.990	0.995	0.995	240
Grape___Mildew	1.000	1.000	1.000	58
Orange_Hanging	0.990	0.995	0.995	820
Peach___Bacterial_soot	1.000	0.997	0.999	363
Peach___Brown_spots	0.990	0.995	0.995	50
Pepper_dell_Bacterial_soot	0.981	0.991	0.987	152
Pepper_dell_Fungal_spots	0.981	0.991	0.986	237
Potato_Farci_Slight	1.000	0.976	0.999	137
Potato_Fungal_spots	0.990	0.995	0.995	147
Potato_Healthy	1.000	1.000	1.000	23
Raspberry_Healthy	1.000	1.000	1.000	43
Schizanthus_Powdery_mildew	1.000	1.000	1.000	747
Strawberry_Healthy	1.000	1.000	1.000	123
Strawberry_Rust	1.000	1.000	1.000	80
Tomato_Bacterial_blight	0.990	0.995	0.995	132
Tomato_Farci_Light	0.930	0.991	0.965	158
Tomato_Fungal_spots	0.990	0.995	0.995	243
Tomato_Leaf_folds	0.992	0.991	0.976	127
Tomato_Tomato_brown_rot	0.990	0.995	0.995	290
Tomato_Spider_mites_Two-spotted_spider_mite	0.976	0.999	0.982	249
Tomato_Tomato_brown_rot_Crl_Virus	0.990	0.995	0.995	831
Tomato_Tomato_mosaic_virus	0.915	0.980	0.946	74
Tomato_Tomato_spider_mite	0.990	0.995	0.995	214
Background	1.000	1.000	1.000	135

Generating Confusion Matrix...

## 5. Classification Report

A detailed classification report was generated, including precision, recall, F1-score, and support for each class. Most classes achieved very high performance, indicating robust feature learning.

The screenshot shows a Google Colab notebook titled "EfficientNet B3.jptn". The interface includes a sidebar with file navigation, a top bar with tabs for WhatsApp, EfficientNet B3.jptn, evaluation ٤٦٢٣, Stream, Projects-Ideas.pdf, and Upload files - malak01. The main area displays a table of predicted labels and their accuracy, followed by Python code for generating Grad-CAM visualizations.

```

Predicted Labels

Top 10 Classes (Highest Accuracy):
1. Background : 100.00%
2. Tomato_Tomato_mosaic_virus : 100.00%
3. Tomato_healthy : 100.00%
4. Raspberry_healthy : 100.00%
5. Strawberry_healthy : 100.00%
6. Tomato_Bacterial_spot : 100.00%
7. Strawberry_Leaf_scorch : 100.00%
8. Apple_Apple_scab : 100.00%
9. Tomato_Bacterial_scorch_Powdery_mildew : 100.00%
10. Apple_Cedar_apple_rust : 100.00%

Bottom 10 Classes (Lowest Accuracy):
1. Corn_(maize)_Cercospora_leaf_spot_Gray_leaf_spot: 80.28%
2. Tomato_Bacterial_spot : 93.91%
3. Tomato_Bacterial_scorch : 93.95%
4. Tomato_Leaf_Mold : 96.00%
5. Tomato_healthy : 96.01%
6. Tomato_Bacterial_scarfspot_leaf_spot : 97.40%
7. Corn_(maize)_Northern_Leaf_Blight : 97.79%
8. Tomato_Bacterial_scorch_healthy : 98.00%
9. Grape_Grape_leaves_brown_spots : 98.00%
10. Tomato_Spider_mites_Two-spotted_spider_mite: 98.00%

[49]: 
from pytorch_grad_cam import GradCAM
from pytorch_grad_cam.utils.model_targets import ClassifierOutputTarget
from pytorch_grad_cam.utils.image import show_cam_on_image

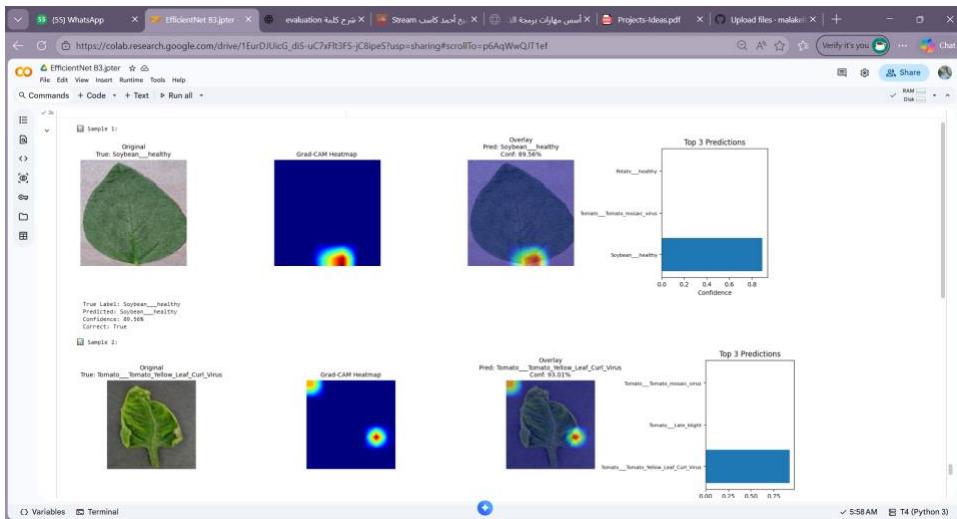
def visualize_gradcam(model, image_tensor, label, class_name, target_layer=None):
    Visualize Grad-CAM for a given image
    ...

```

Variables Terminal ✓ 5:58 AM T4 (Python 3)

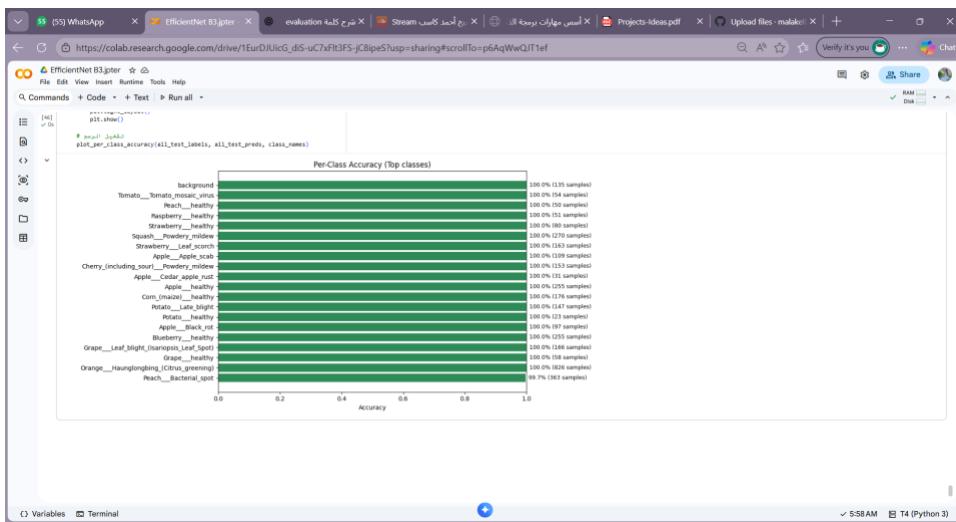
## 6. Per-Class Performance

Per-class accuracy analysis highlights the strongest and weakest classes. This helps identify which categories the model handles best and where further improvements could be applied.



## 7. Model Interpretation (Grad-CAM)

Grad-CAM visualizations were used to interpret model predictions. They highlight the regions of the image that contributed most to the predicted class, enhancing explainability and trust in the model.



## 8. Conclusion

The EfficientNet-B3 model demonstrates excellent performance in image classification tasks. Through transfer learning, fine-tuning, and thorough evaluation, the model achieves high accuracy and reliable predictions, making it suitable for academic and practical applications.