PLUMBAGO ZEYLANICA

Tanishk N Shinde

TABLE OF CONTENTS

DATA AND INFORMATION VISUALIZATION

Class Distribution

Stacked Area Chart
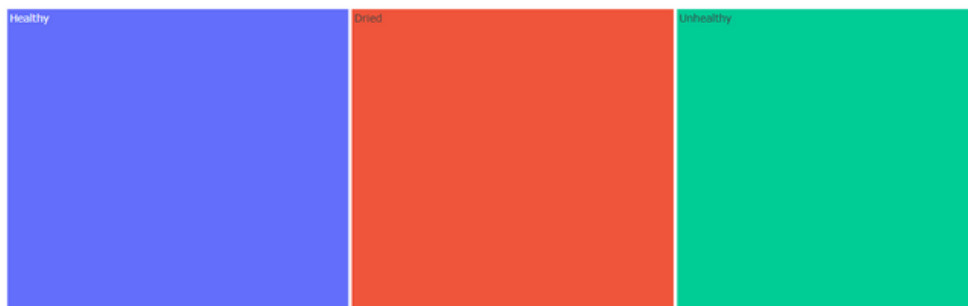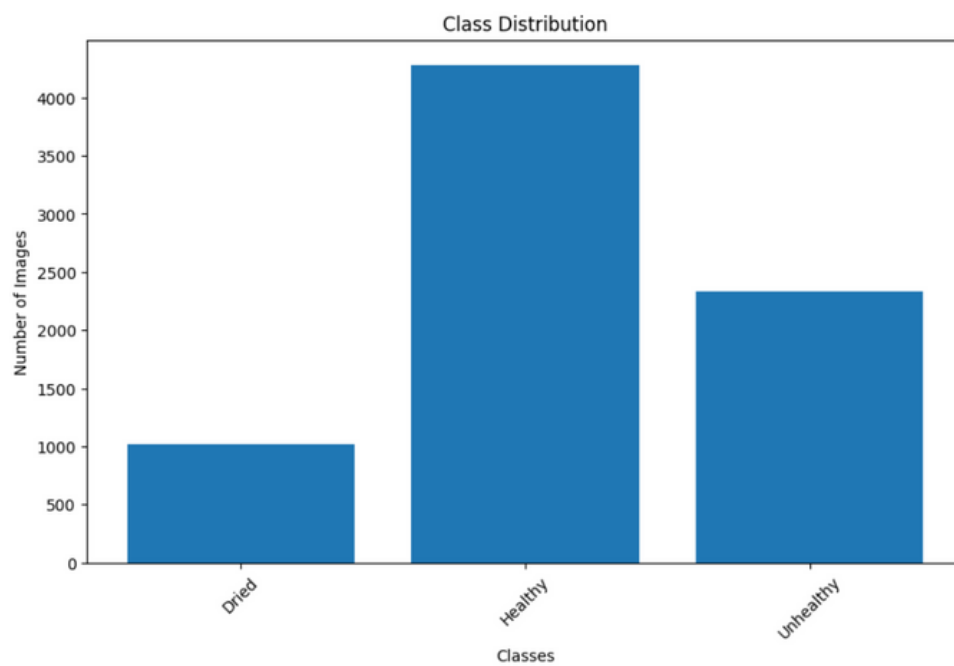
DATA AND INFORMATION VISUALIZATION
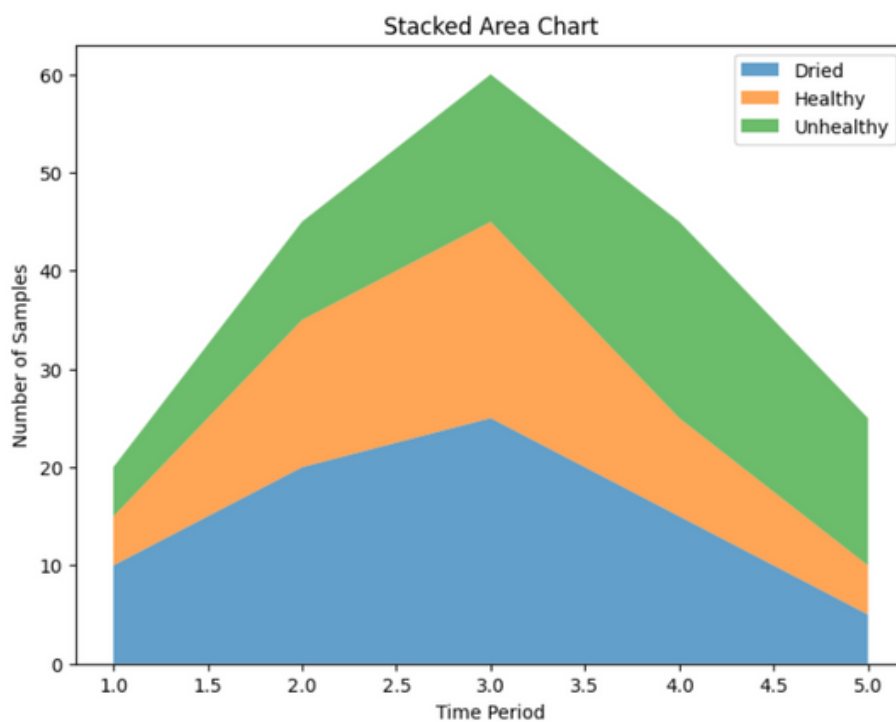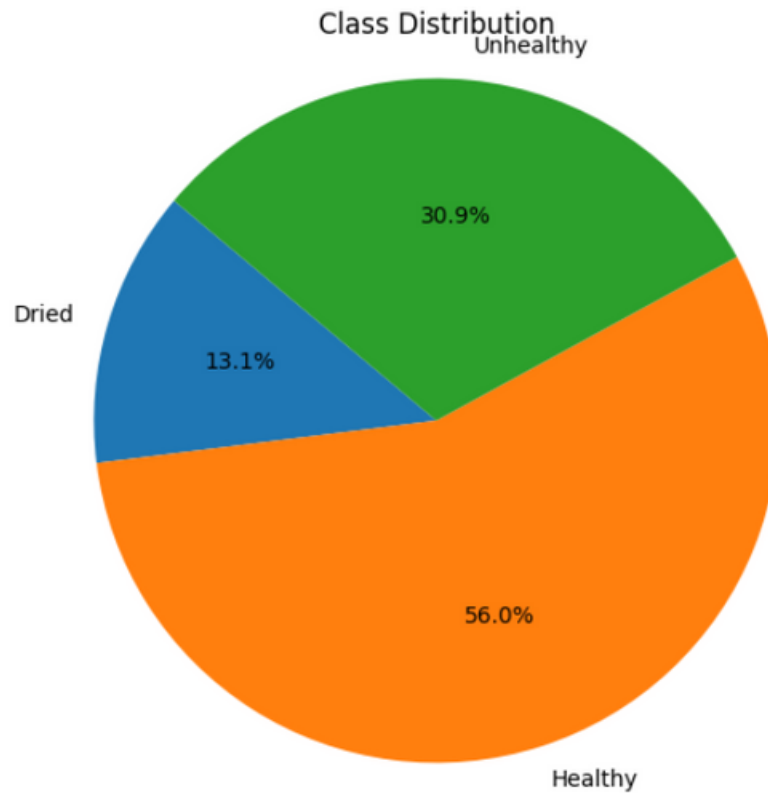
Class Distribution

EFFICIENTNET-B0

EfficientNet-B0 is a convolutional neural network (CNN) architecture that has gained prominence in the field of deep learning for computer vision tasks. It was introduced by researchers in 2019 and is known for its efficiency and impressive performance in image classification tasks.
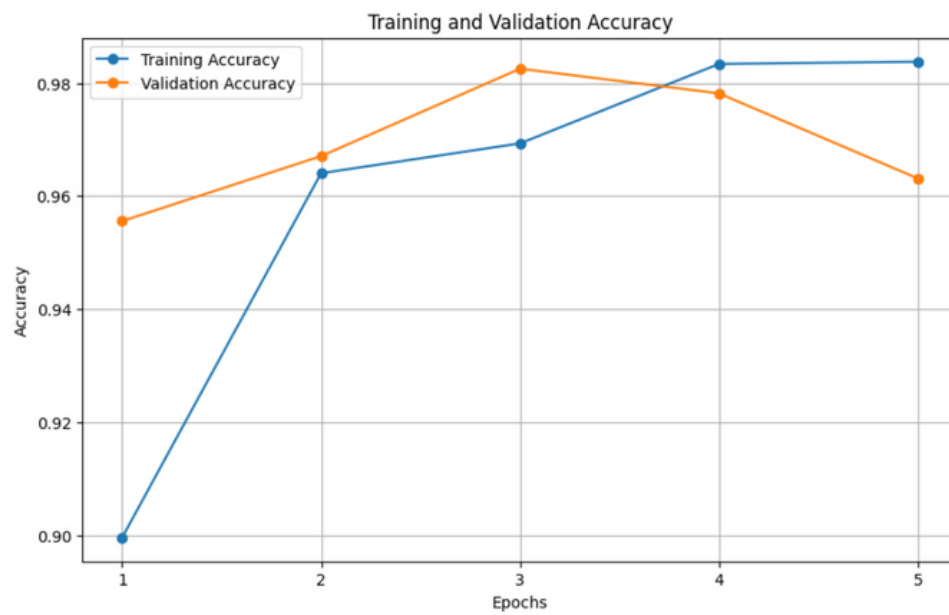
EfficientNet-B0 is part of a family of models, with 'B0' denoting the base model. The 'EfficientNet' series is designed to be highly efficient in terms of both computational resources and model size while delivering state-of-the-art performance. The key innovation behind EfficientNet is the compound scaling, which involves optimizing the network's depth, width, and resolution simultaneously. This scaling strategy allows EfficientNet to achieve a superior balance between accuracy and computational cost.

EfficientNet-B0 employs a combination of depthwise separable convolutions and a "swish" activation function, which contributes to its compactness and effectiveness. The model architecture consists of multiple stacked blocks, including inverted residual blocks and linear bottlenecks, which help it capture intricate features in the data.
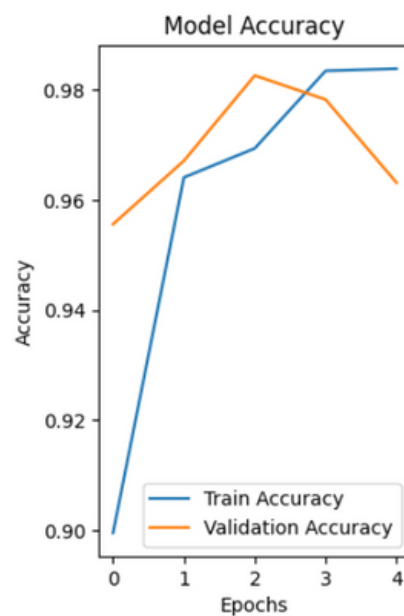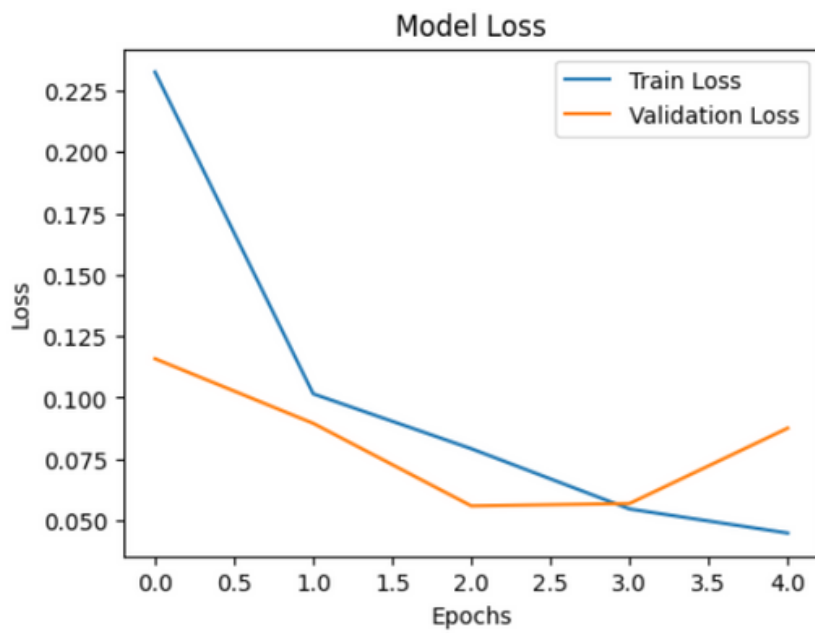
One of the remarkable aspects of EfficientNet-B0 is its transfer learning capability. By pre-training on large image datasets like ImageNet and then fine-tuning on specific tasks, it can adapt to a wide range of image classification challenges. This transfer learning approach has made EfficientNet-B0 a popular choice for various computer vision applications.
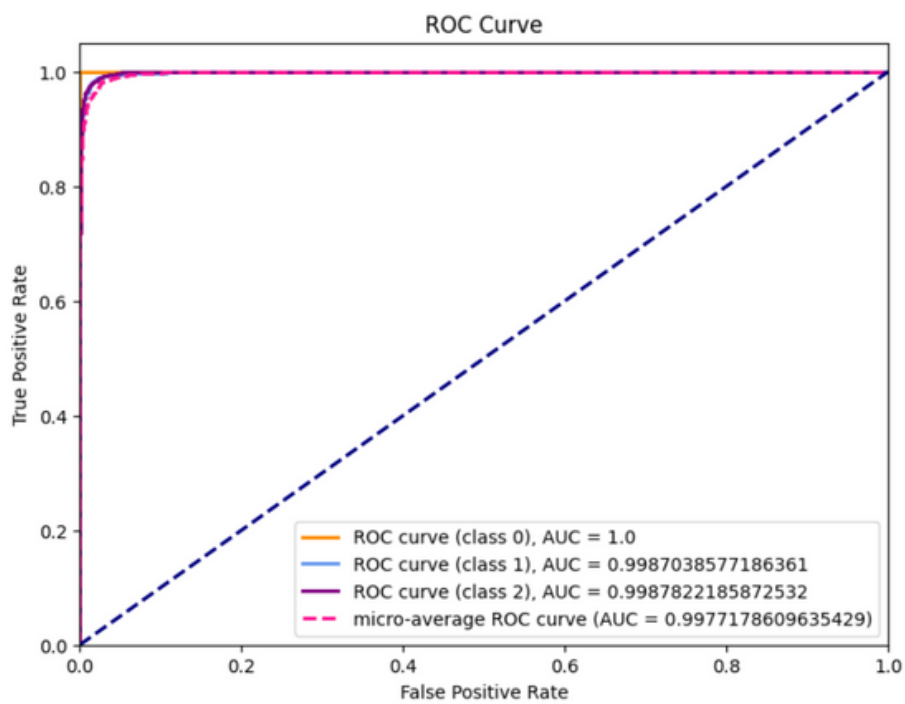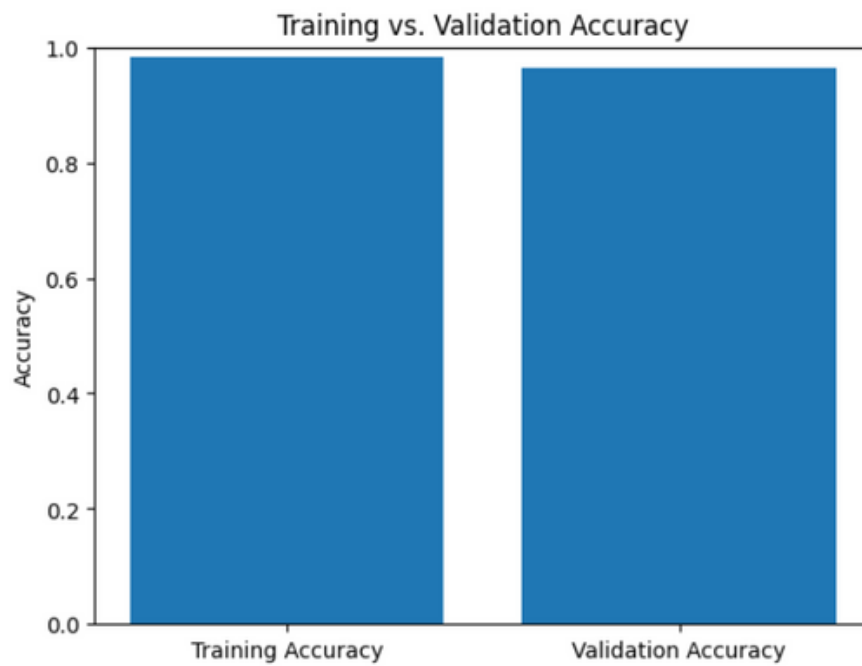
EfficientNet-B0 has served as a baseline model for other EfficientNet variants (B1, B2, B3, etc.) that are larger and more powerful. The choice of variant depends on the specific requirements of a task, with larger variants offering better accuracy but requiring more computational resources.

In summary, EfficientNet-B0 is a highly efficient and effective convolutional neural network architecture for image classification tasks. Its innovative compound scaling approach, compact design, and transfer learning capabilities make it a valuable tool in the deep learning community for a wide range of computer vision applications.

Training and Validation Accuracy

Final Training Accuracy: 0.9838
Final Validation Accuracy: 0.9631

Model Loss

Model Accuracy

Training vs. Validation Accuracy

ROC Curve

ROC curve (class 0), AUC = 1.0
ROC curve (class 1), AUC = 0.9987038577186361
ROC curve (class 2), AUC = 0.9987822185872532
micro-average ROC curve (AUC = 0.9977178609635429)

Confusion Matrix

EfficientNet-B0

Accuracy Val_Accuracy Recall F1-Score ROC Curve Loss Val_Loss
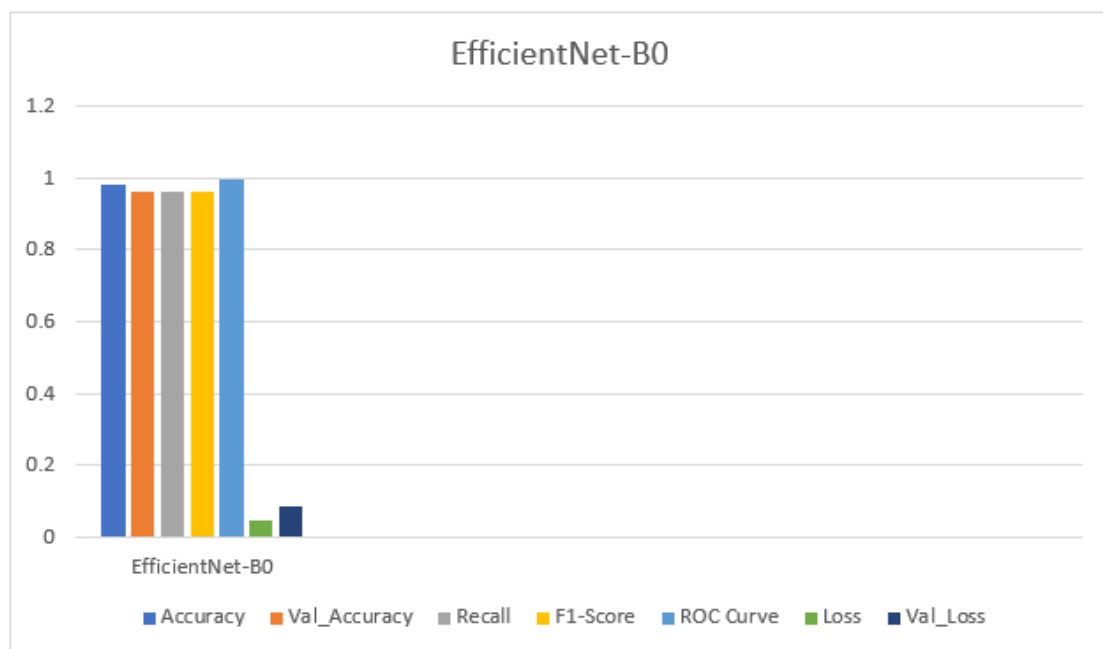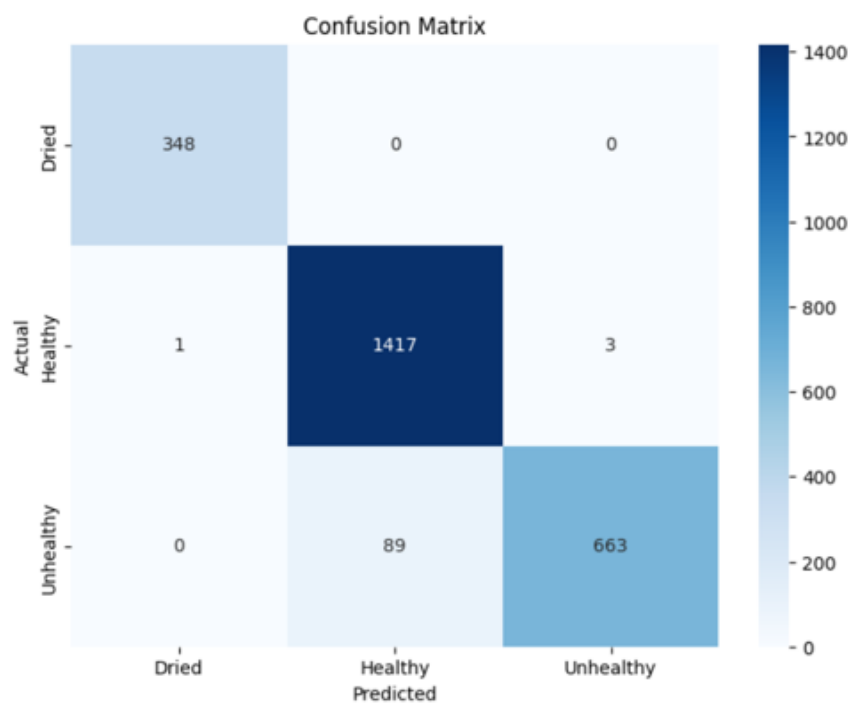
MOBILENETV2

MobileNetV2 is a highly efficient convolutional neural network (CNN) architecture designed for mobile and embedded devices. It was developed by Google as a successor to the original MobileNet, with the primary goal of enabling state-of-the-art deep learning performance on resource-constrained platforms. Here's an overview in 300 words:
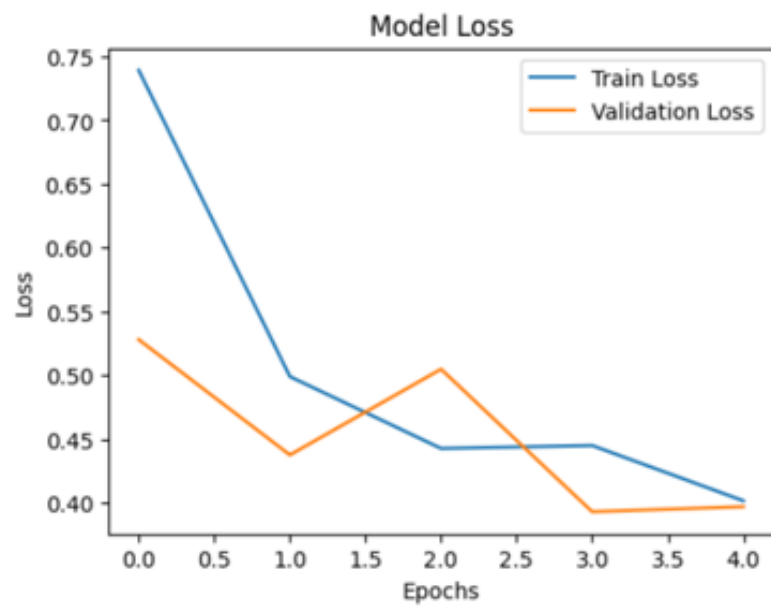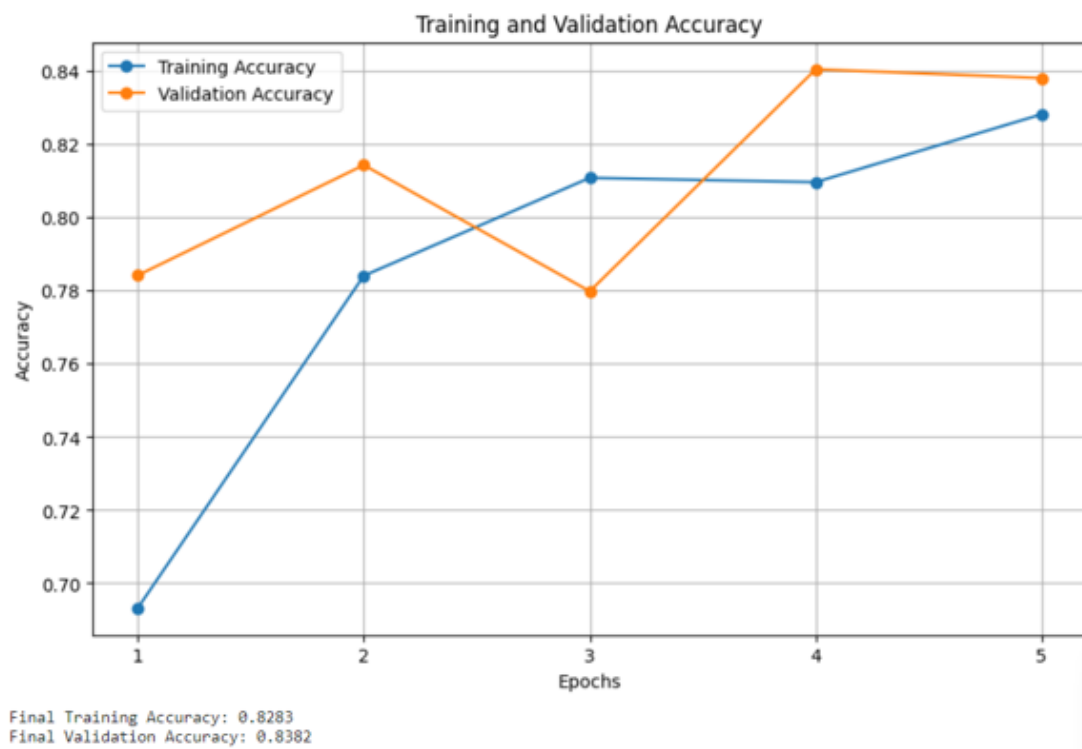
MobileNetV2 is characterized by its lightweight and compact design, making it ideal for applications like real-time object recognition on smartphones, IoT devices, and other edge computing environments. The architecture is known for its exceptional speed and accuracy, making it a popular choice for tasks such as image classification, object detection, and more.
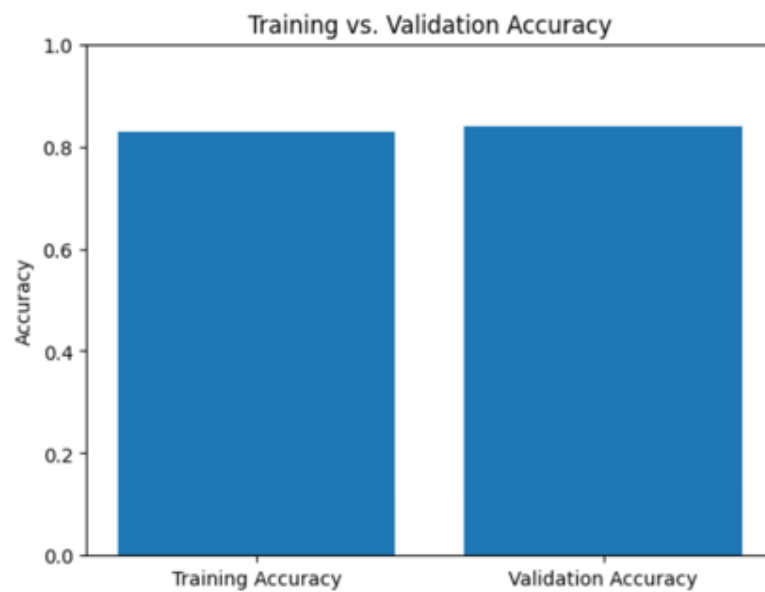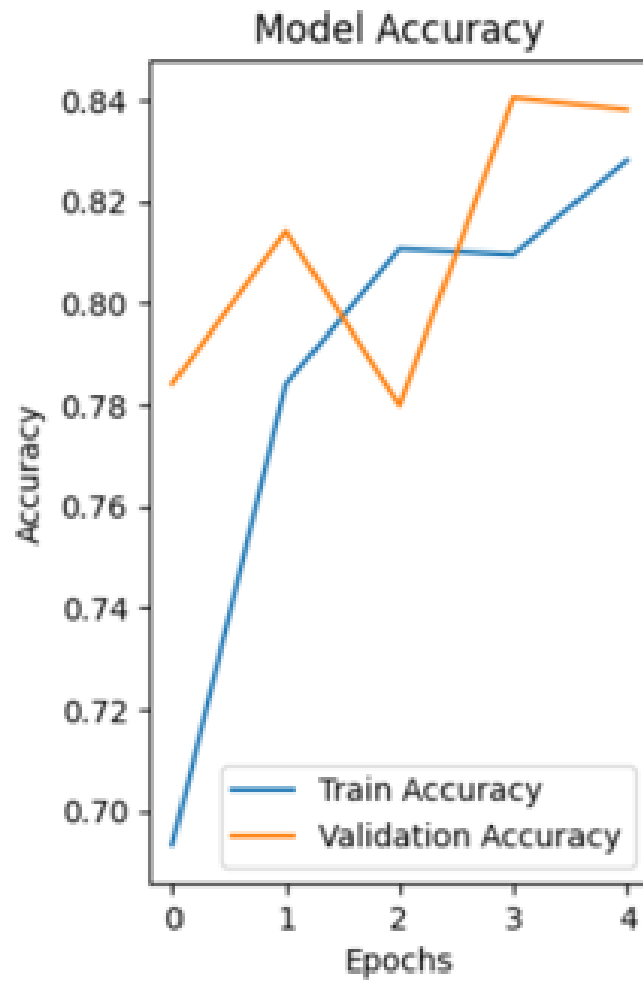
The key innovation in MobileNetV2 is the use of inverted residual blocks, which include a bottleneck layer with fewer filters followed by expansion and projection layers. This design reduces the computational load while maintaining high representation capacity. MobileNetV2 also employs depthwise separable convolutions, which split standard convolutions into depthwise and pointwise convolutions. This separation further reduces the number of parameters and computational requirements.
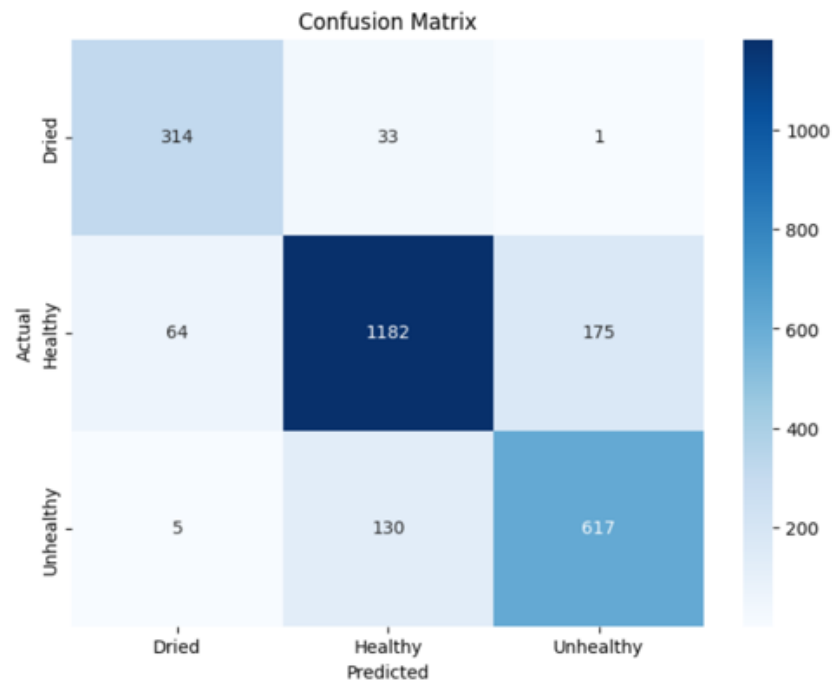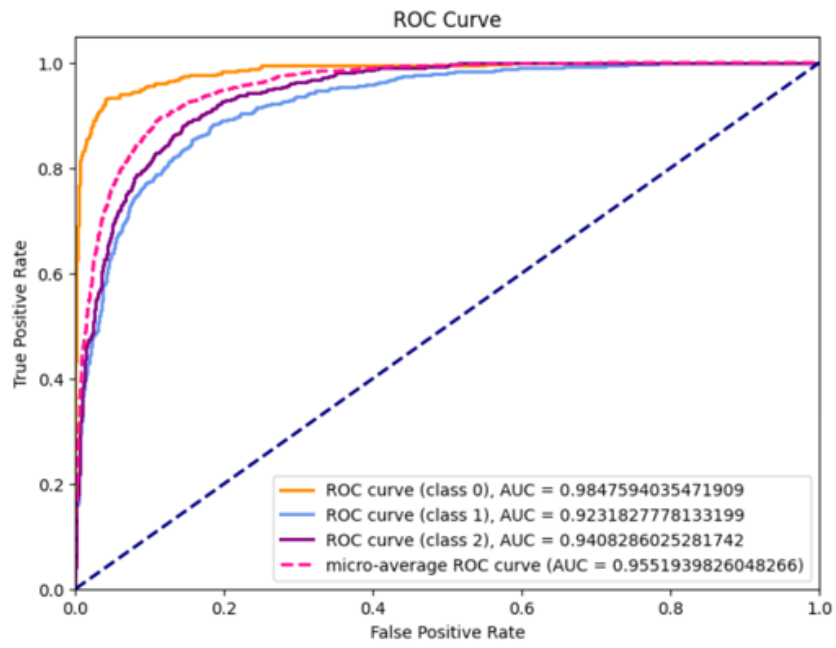
In addition to its efficiency, MobileNetV2 includes a feature called "linear bottlenecks," which helps prevent information loss during the network's forward pass. It employs skip connections to facilitate the flow of gradients during training, promoting faster convergence and better training stability.

MobileNetV2 comes in various pre-trained models with different complexities, allowing developers to choose the right trade-off between speed and accuracy for their specific application. This adaptability is one of its significant advantages.

MobileNetV2 has achieved impressive results in benchmarking competitions, and it's often used as a backbone architecture in larger models for tasks like object detection and semantic segmentation. The network has a wide range of practical applications, including image and video processing, autonomous vehicles, and even augmented reality.

Training and Validation Accuracy

Final Training Accuracy: 0.8283
Final Validation Accuracy: 0.8382

Model Loss

Model Accuracy

Training vs. Validation Accuracy

ROC Curve

ROC curve (class 0), AUC = 0.9847594035471909
ROC curve (class 1), AUC = 0.9231827778133199
ROC curve (class 2), AUC = 0.9408286025281742
micro-average ROC curve (AUC = 0.9551939826048266)

Confusion Matrix

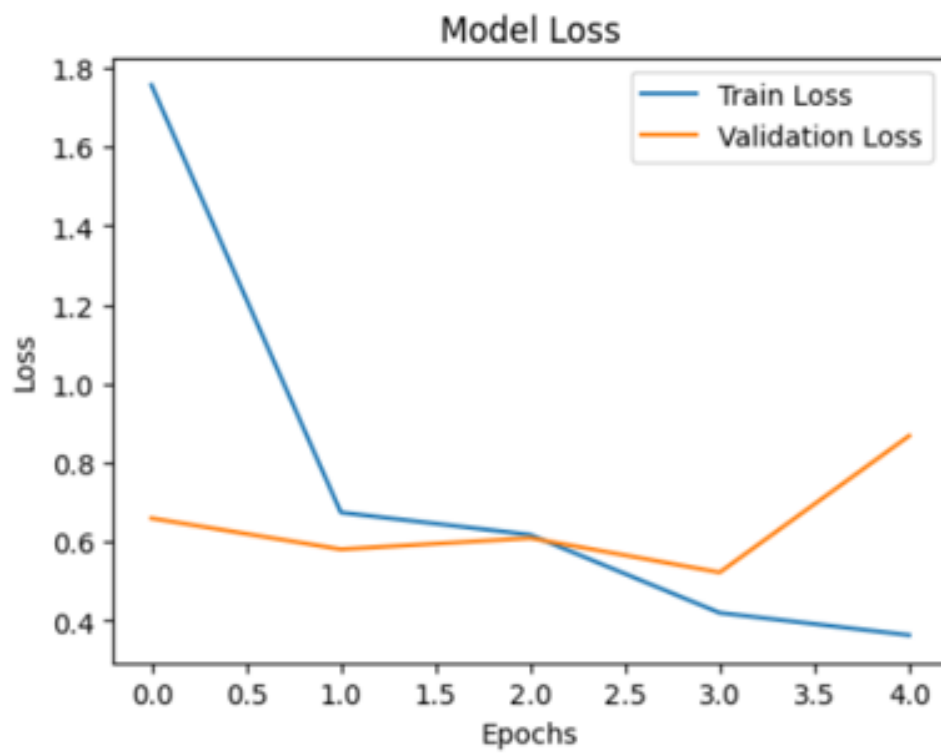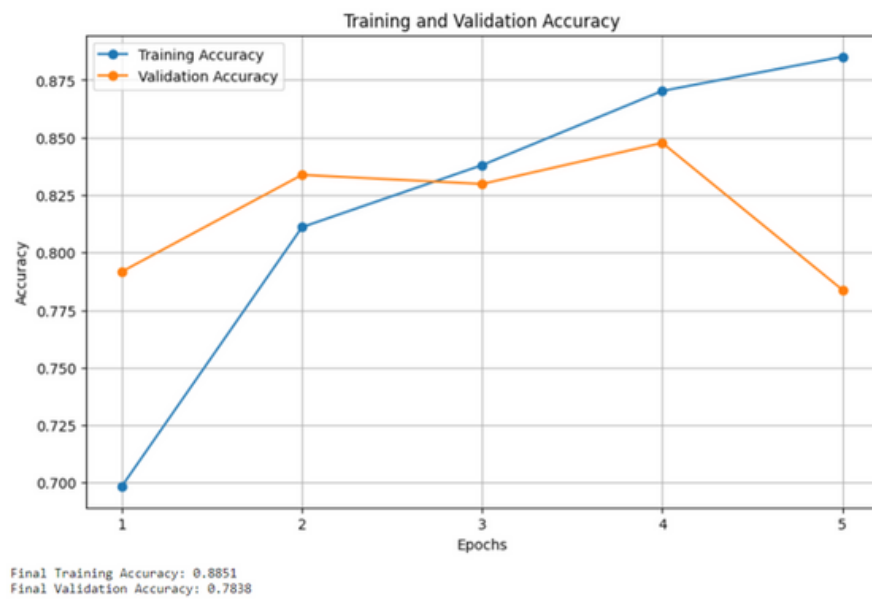| Actual \ Predicted	Dried	Healthy	Unhealthy
Dried	314	33	1
Healthy	64	1182	175
Unhealthy	5	130	617

MobileNetV2

DENSENET201

DenseNet-201 is a deep convolutional neural network architecture that was developed to address the vanishing gradient problem in deep neural networks. It is an extension of the original DenseNet, which stands for "Densely Connected Convolutional Networks." This architecture was designed by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger and has been widely used in computer vision tasks, particularly image classification.
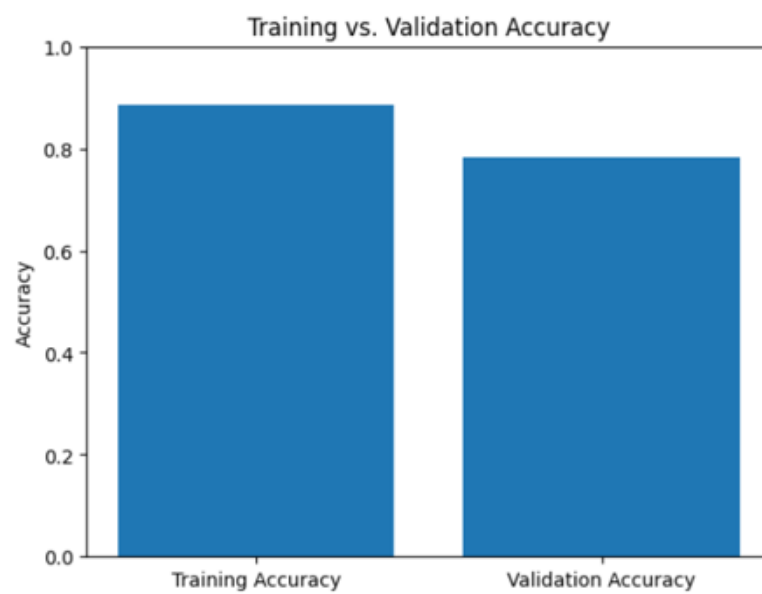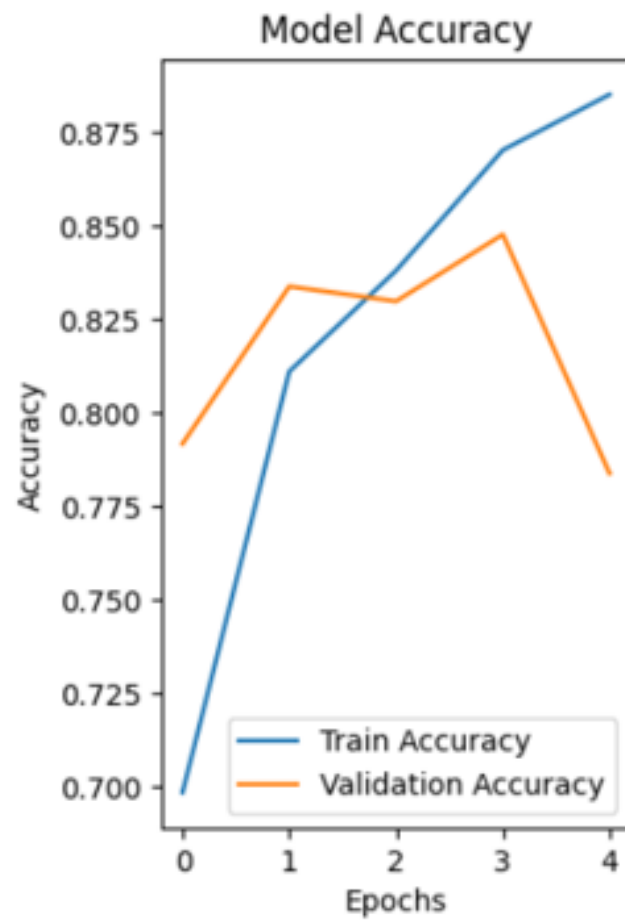
The key innovation in DenseNet-201 is its dense connectivity pattern. In traditional convolutional neural networks, information flows sequentially from one layer to the next. DenseNet, on the other hand, connects each layer to every subsequent layer in a feedforward fashion. This dense connectivity helps in feature reuse and alleviates the vanishing gradient problem, enabling the network to train effectively even when it's very deep.
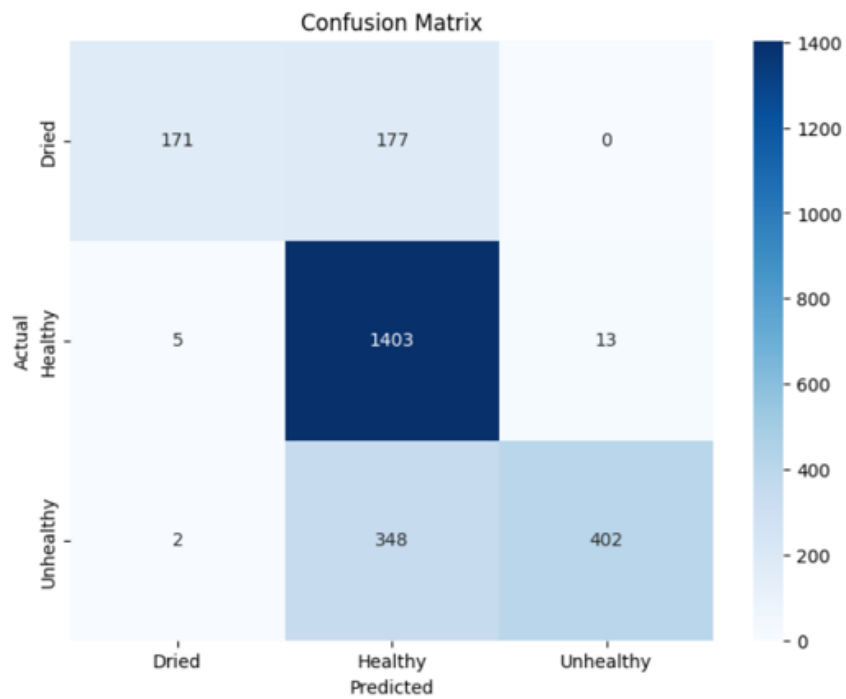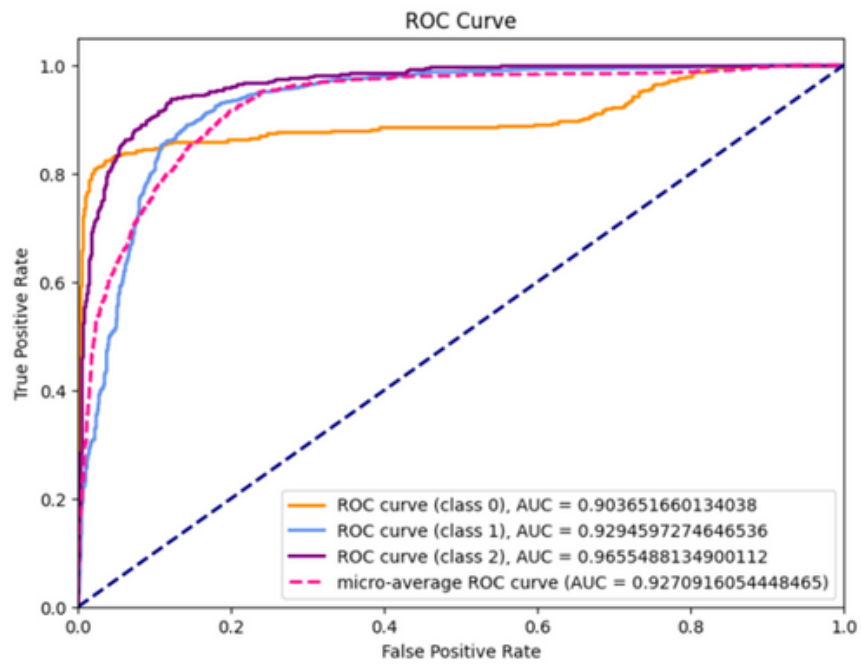
DenseNet-201 specifically consists of 201 layers and is often chosen for tasks that require a high level of feature extraction, such as object recognition in images. It has a composite structure that includes dense blocks, which contain several convolutional layers with bottleneck layers to reduce computational complexity, and transition layers that control the spatial dimensions of feature maps as they flow through the network.
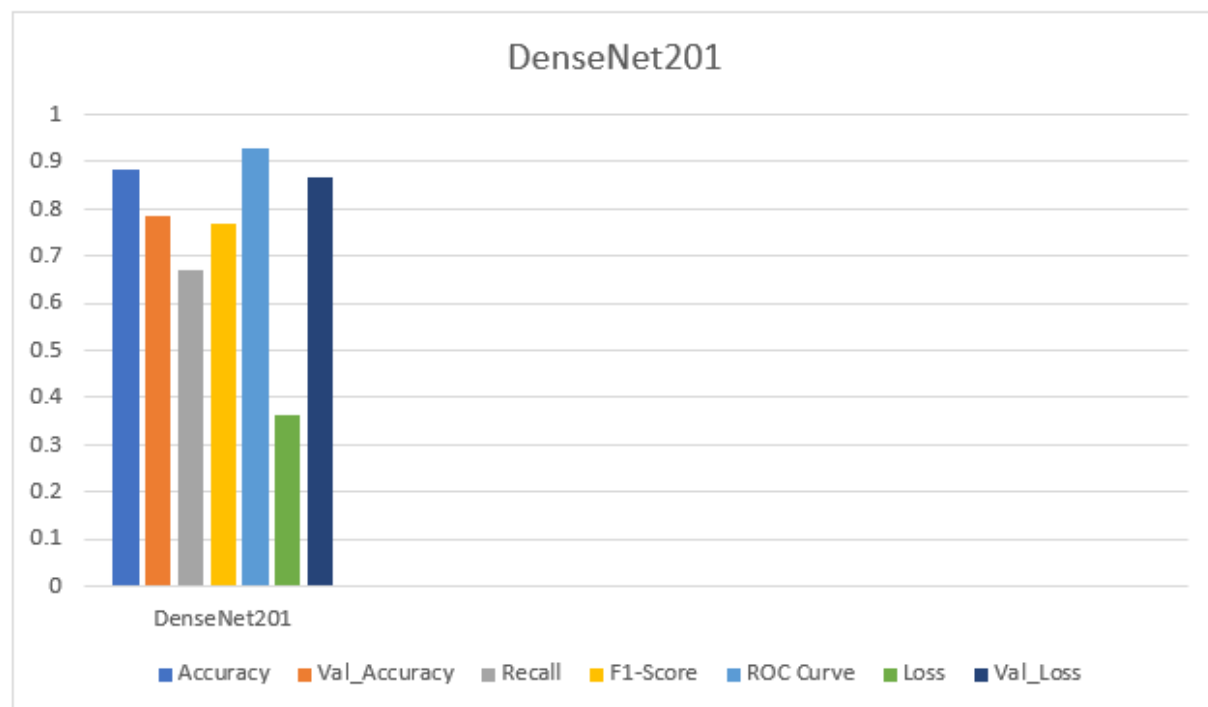
The densely connected nature of DenseNet-201 allows it to achieve remarkable performance with fewer parameters compared to other architectures, making it more computationally efficient. It excels in various computer vision tasks, including image classification, object detection, and segmentation. Researchers and practitioners often employ transfer learning techniques with pre-trained DenseNet-201 models, fine-tuning them on specific tasks, which can significantly reduce training time and data requirements.

In summary, DenseNet-201 is a powerful convolutional neural network architecture known for its dense connectivity pattern, which addresses the vanishing gradient problem and allows for efficient feature reuse. It has been successful in various image-related tasks, making it a popular choice in the field of deep learning and computer vision.

Training and Validation Accuracy

Final Training Accuracy: 0.8851
Final Validation Accuracy: 0.7838

Model Loss

Model Accuracy

Training vs. Validation Accuracy

ROC Curve

ROC curve (class 0), AUC = 0.903651660134038
ROC curve (class 1), AUC = 0.9294597274646536
ROC curve (class 2), AUC = 0.9655488134900112
micro-average ROC curve (AUC = 0.9270916054448465)

Confusion Matrix

DenseNet201

Accuracy Val_Accuracy Recall F1-Score ROC Curve Loss Val_Loss
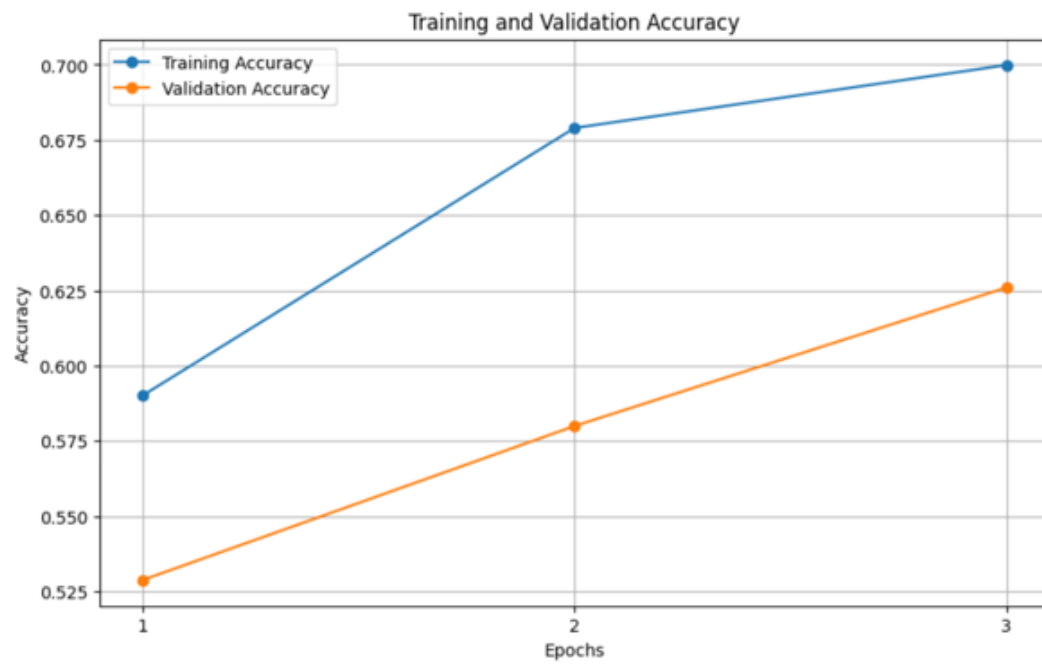
XCEPTION

Xception, short for "Extreme Inception," is a deep convolutional neural network (CNN) architecture designed for image classification and object recognition tasks. It was introduced by François Chollet, the creator of the popular deep learning framework Keras, in 2016. Xception is known for its innovative and highly efficient architecture, which is a significant advancement in the field of deep learning.
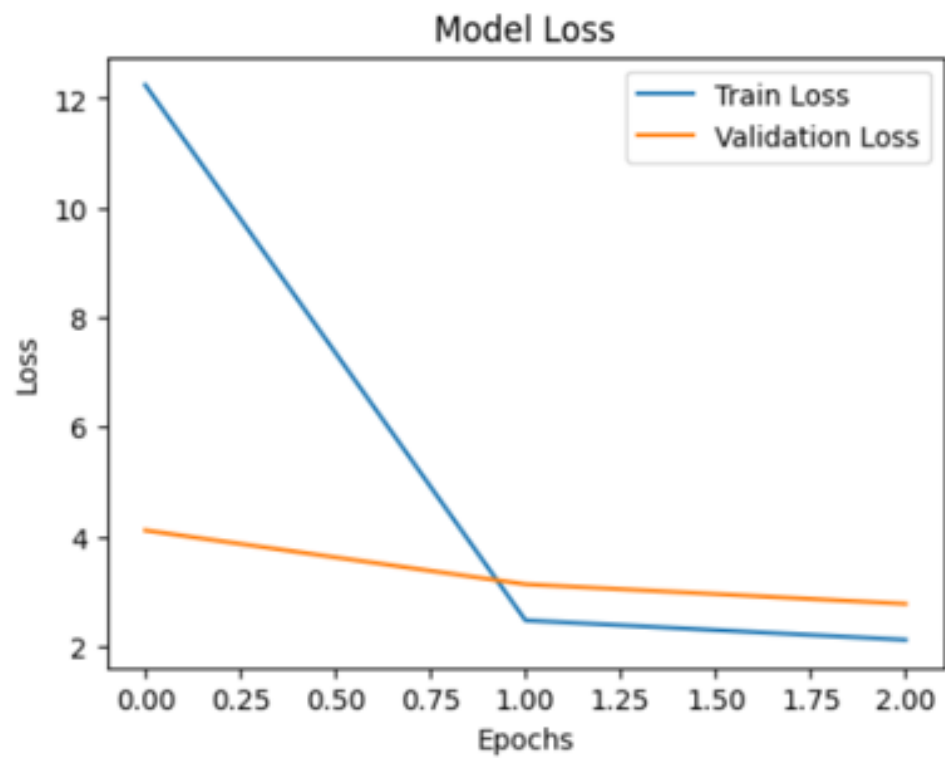
At its core, Xception is based on the concept of depthwise separable convolutions. Traditional CNNs use standard convolutions, which apply filters across all input channels, making them computationally intensive. Xception, on the other hand, separates the depthwise convolution (applying a single filter to each input channel) from the pointwise convolution (combining the results). This separation significantly reduces the number of parameters and computations, making the network faster and more efficient.
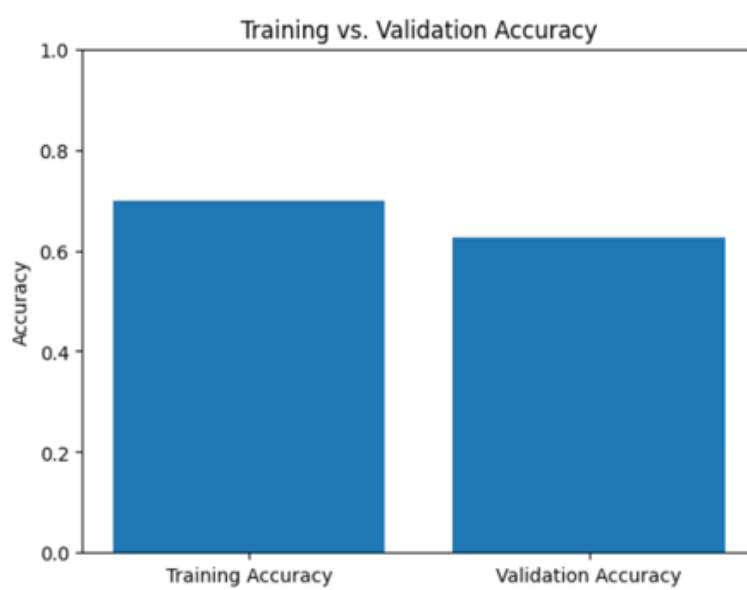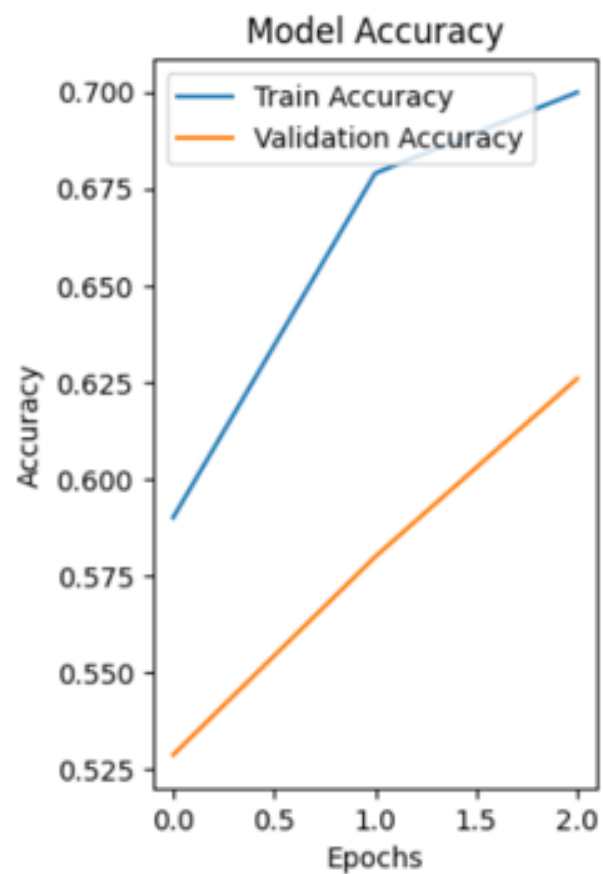
The Xception architecture contains multiple convolutional blocks, each with depthwise separable convolutions, batch normalization, and skip connections. These blocks are stacked on top of each other, creating a deep network that can learn intricate features from images. Xception's depthwise separable convolutions help it excel in various computer vision tasks, including image classification, object detection, and semantic segmentation.
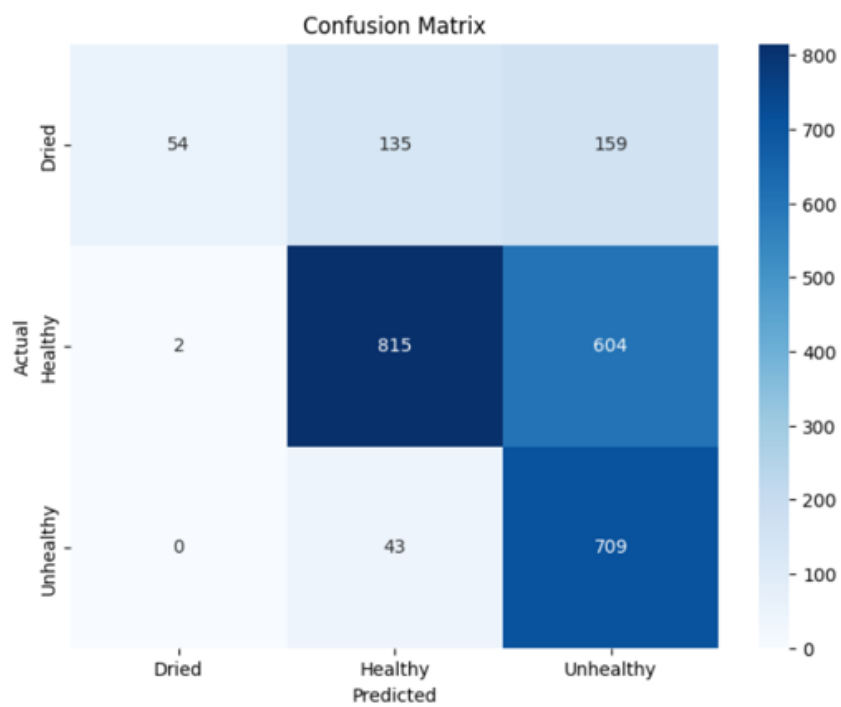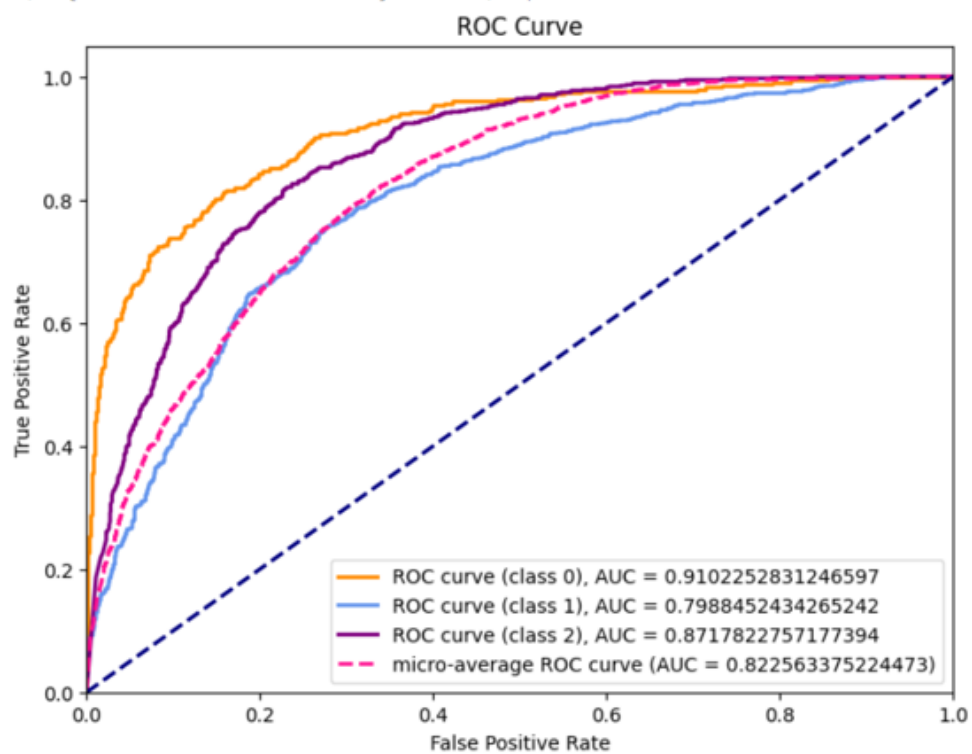
Xception has demonstrated remarkable performance on benchmark datasets and has been widely adopted in transfer learning scenarios. Researchers and practitioners often use pre-trained Xception models as a starting point for various computer vision tasks, fine-tuning them for specific applications. Its efficiency and strong generalization capabilities have made Xception a popular choice in the deep learning community, contributing to advancements in image analysis and understanding.
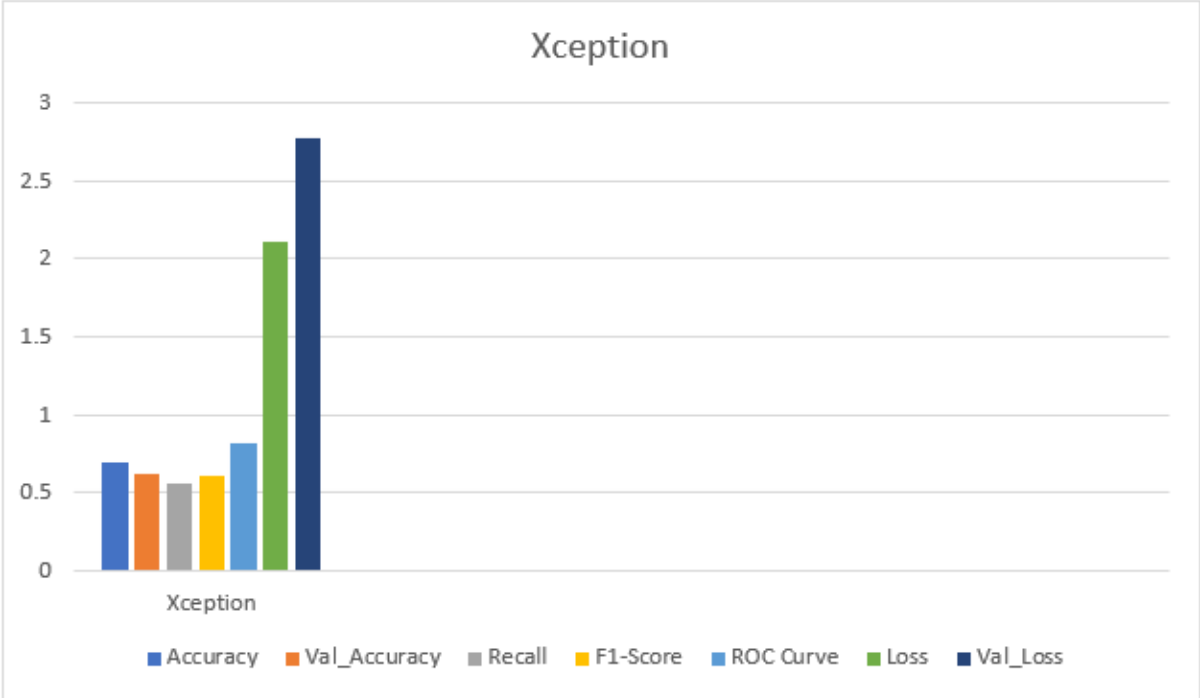
Training and Validation Accuracy

Final Training Accuracy: 0.6999
Final Validation Accuracy: 0.6259

Model Loss

Model Accuracy

Training vs. Validation Accuracy

ROC Curve

ROC curve (class 0), AUC = 0.9102252831246597
ROC curve (class 1), AUC = 0.7988452434265242
ROC curve (class 2), AUC = 0.8717822757177394
micro-average ROC curve (AUC = 0.822563375224473)

Confusion Matrix

Xception

COMPARISON