كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Helwan University

Deep Learning

&

Neural Network

DR.

Eslam Gamal

Contents

Overview:

This document provides an in-depth explanation of three prominent neural network architectures: ResNet, Xception, and DenseNet. Each section covers the architecture’s core design principles, step-by-step workings, and visual representations. References to the original research papers are included.

Drawbacks:

Our dataset presents several challenges, such as (class imbalance, synthetic backgrounds, lack of contextual information, etc...). We have made significant efforts in preprocessing to address these issues and enhance model performance.

Class Imbalance:

Our dataset has imbalanced classes, which can affect model performance, but we address this issue during preprocessing.

Synthetic Backgrounds:

In our dataset, the images in the color folder often have uniform or artificial backgrounds, which can lead to overfitting. We mitigate this issue by incorporating data augmentation and background removal techniques.

Dataset Size:

Although the dataset is large, it might still be insufficient for training very deep models without data augmentation or transfer learning, particularly for classes with fewer samples.

Overfitting:

In our dataset, overfitting occurs because of the imbalanced classes, the excessive similarity between many classes, and the likely imbalance ratio of data samples being likely 10:1.

Architecture:

ResNet (Residual Networks):

Overview:

ResNet was introduced to address the vanishing gradient problem in deep neural networks. It uses residual learning by introducing shortcut connections (skip connections) that allow gradients to flow directly through the network.

Key Components:

1. **Residual Block**: A small sub-network with skip connections that add the input directly to the output.
2. **Skip Connections**: Enable identity mapping, ensuring that information flows without degradation.
3. **Batch Normalization**: Stabilizes training and accelerates convergence.

Step-by-Step Details:

1. **Input Layer**: Accepts an image or feature map.
2. **Convolutional Layers**: Extract features using filters.
3. **Residual Block**:
    - Two convolutional layers followed by batch normalization and ReLU activation.
    - A shortcut connection adds the input to the output of the block.
4. **Stacking Blocks**: Multiple residual blocks are stacked to form a deep network.
5. **Fully Connected Layer**: Outputs class probabilities or regression results.

Graph:

Training and Validation Accuracy:

We used this Graph to track the accuracy of the model on both the training and validation datasets over epochs.

AUC (Area Under the Curve):

AUC is plotted to evaluate the model's ability to distinguish between classes. It provides a measure of the model's performance across all classification thresholds.

ROC Curve (Receiver Operating Characteristic):

The ROC curve is plotted to visualize the trade-off between true positive rate and false positive rate. It helps assess the model's classification performance at different thresholds.

Pros of using ResNet on PlantVillage Dataset:

Mitigation of Vanishing Gradient Problem:

Residual connections allow gradients to flow through the network more easily during backpropagation, enabling training of very deep networks without performance degradation.

Deep Feature Extraction:

ResNet’s architecture, with its residual connections, enables it to learn deep and complex features, making it highly effective for tasks like disease classification in images.

Cons of using ResNet on PlanetVillage Dataset:

Overfitting to Synthetic Backgrounds:

The dataset contains uniform backgrounds, which causes ResNet to overfit on these features rather than concentrate on the leaf patterns. This issue is particularly evident when the model is fine-tuned without additional preprocessing, and Since we built our model from scratch and do not have techniques like dropout or regularization, we had to rely on callbacks, such as early stopping, to mitigate overfitting.

Sensitivity to Dataset Bias:

ResNet can amplify biases in the dataset, such as class imbalance or over-representation of certain features, leading to suboptimal generalization.

DenseNet (Dense Convolutional Network)

Overview:

DenseNet introduces dense connections between layers, where each layer receives the outputs of all preceding layers as inputs. This reduces redundancy, improves feature reuse, and mitigates the vanishing gradient problem.

Key Components:

1. **Dense Block:** A sequence of layers connected to each other.
2. **Transition Layers:** Reduce feature map dimensions using 1x1 convolutions and pooling.
3. **Growth Rate:** Controls the number of new features added by each layer.

Step-by-Step Details:

1. **Input Layer:** Accepts an image or feature map.
2. **Dense Block:**
    - Each layer takes inputs from all preceding layers.
    - Outputs are concatenated instead of added.
3. **Transition Layer:** Applies 1x1 convolutions and pooling to reduce dimensions.
4. **Stacking Blocks:** Multiple dense blocks separated by transition layers.
5. **Global Average Pooling:** Reduces spatial dimensions before the final classification layer.

Graph:

Training and Validation Accuracy:

We used this Graph to track the accuracy of the model on both the training and validation datasets over epochs.

AUC (Area Under the Curve):

AUC is plotted to evaluate the model's ability to distinguish between classes. It provides a measure of the model's performance across all classification thresholds.

ROC Curve (Receiver Operating Characteristic):

The ROC curve is plotted to visualize the trade-off between true positive rate and false positive rate. It helps assess the model's classification performance at different thresholds.

Pros of using DenseNet on PlantVillage Dataset:

Efficient Feature Propagation:

DenseNet connects each layer to every other layer, enabling efficient reuse of features and reducing redundancy. This is particularly useful for learning subtle patterns in leaf textures and diseases.

Fewer Parameters:

DenseNet is parameter-efficient compared to other architectures like ResNet, as it avoids learning redundant feature maps. This makes it suitable for datasets like PlantVillage, which may not be extremely large.

Better Gradient Flow:

The dense connections improve gradient flow during backpropagation, reducing the risk of vanishing gradients and enabling better training for deeper networks.

Pretrained Weights Availability:

Pretrained weights (e.g., from ImageNet) can be leveraged, allowing for transfer learning to improve performance even with limited training data.

Cons of using DenseNet on PlanetVillage Dataset:

Sensitivity to Backgrounds:

DenseNet might overfit to the synthetic backgrounds in the PlantVillage dataset if not properly addressed, as its dense feature extraction can pick up irrelevant details.

Computational Overhead:

DenseNet requires more memory and computation due to its dense connections.

Xception (Extreme Inception):

Overview:

Xception builds upon the Inception architecture by using depthwise separable convolutions, which factorize a standard convolution into two simpler operations: depthwise convolution and pointwise convolution. This improves computational efficiency and performance.

Key Components:

1. **Depthwise Separable Convolution:** Splits convolution into depthwise and pointwise operations.
2. **Residual Connections:** Used to ensure efficient gradient flow.
3. **Fully Convolutional Design:** Eliminates fully connected layers for better parameter efficiency.

Step-by-Step Details:

1. **Input Layer:** Accepts an image or feature map.
2. **Depthwise Convolution:** Applies a single filter per input channel.
3. **Pointwise Convolution:** Combines the outputs of depthwise convolutions.
4. **Residual Blocks:** Combine depthwise separable convolutions with shortcut connections.
5. **Global Average Pooling:** Reduces spatial dimensions before the final output layer.

Graph:

Training and Validation Accuracy:

We used this Graph to track the accuracy of the model on both the training and validation datasets over epochs.

AUC (Area Under the Curve):

AUC is plotted to evaluate the model's ability to distinguish between classes. It provides a measure of the model's performance across all classification thresholds.

ROC Curve (Receiver Operating Characteristic):

The ROC curve is plotted to visualize the trade-off between true positive rate and false positive rate. It helps assess the model's classification performance at different thresholds.

Pros of using DenseNet on PlantVillage Dataset:

High Performance on Image Classification Tasks:

Xception, based on depthwise separable convolutions, is highly efficient at extracting spatial features, making it well-suited for leaf disease classification.

Strong Generalization:

Xception's architecture is designed to capture fine-grained details, which can help in distinguishing subtle differences between diseases.

Pretrained Weights Availability:

Pretrained weights (e.g., from ImageNet) can be leveraged, allowing for transfer learning to improve performance even with limited training data.

Cons of using DenseNet on PlanetVillage Dataset:

Overfitting:

Xception's complexity can lead to overfitting, especially with the PlantVillage dataset's synthetic backgrounds.

Requires High-Quality Input:

The model performs best with clean, high-resolution images, which may not reflect noisy, real-world agricultural conditions.

Dependency on Preprocessing:

Effective training with Xception often requires well-preprocessed data (e.g., resizing, normalization), and suboptimal preprocessing can degrade performance.

Comparative Summary:

Criteria	*ResNet*	*DenseNet*	*Xception*
Architecture	Deep residual learning with skip connections	Dense connections between layers, each layer receives input from all previous layers	Depthwise separable convolutions, efficient
Performance	Excellent for deep networks, high accuracy	High accuracy, especially in complex tasks due to dense connections	High accuracy, especially with transfer learning
Efficiency	Computationally expensive, but efficient for deep models	Efficient with fewer parameters compared to traditional CNNs	Computationally efficient with fewer parameters
Overfitting	Lower risk due to residual connections	Lower risk due to dense connections	Moderate risk, mitigated with data augmentation
Generalization	Strong generalization, especially with deeper models	Excellent generalization, particularly in complex tasks	Strong generalization with fine-tuning
Plant Disease Classification	Excellent for complex patterns and deep datasets	Great for complex disease classification, especially with fine details	Excellent for leaf disease classification with fine details

Conclusion:

ResNet, Xception, and DenseNet are key advancements in deep learning, each tackling distinct challenges. ResNet introduced residual connections for training deeper networks, Xception enhanced efficiency with depthwise separable convolutions, and DenseNet promoted feature reuse through dense connectivity. Collectively, these architectures have significantly advanced computer vision and continue to inspire new research.