STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–18.
Published online in International Academic Press (www.IAPress.org)

Optimizing Lemongrass Disease Detection: A Comparative Analysis of Neural Architecture Search (NAS) with convolutional neural network (CNN) and Transfer Learning models

Putra Sumari[1], Ahmed Abed Mohammed[2,*], Mustafa M. Abd Zaid[3], Shuchuan Tian[1], Wenjing Wu[1], Tingting Zhang[1], Xiaolin Fu[1], Haolu Dong[1], Yangyang Wei[1]

[1]*School of Computer Science, University Sains Malaysia, 11800 Peneng, Malaysia*
[2] *College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwaniyah, Iraq*
[3] *College of Technical Engineering, Islamic University, Najaf, Iraq*

Abstract Lemongrass is an economically important crop that faces significant disease challenges, making timely and accurate detection crucial for enhancing crop yield and quality. Traditional detection methods rely on manual observation, which can be subjective and inefficient. This study uses convolutional neural network (CNN) models to investigate automated detection methods for lemongrass diseases. We mainly utilize Neural Architecture Search (NAS) methods involving Efficient Neural Architecture Search (ENAS) and Partial Channel-Differentiable Architecture Search (PC-DARTS) to enhance structural models. Then, we compare the execution of these NAS-optimized models with transfer learning models such as VGG16, AlexNet, and Inception. We applied data augmentation and preprocessing methods to improve the model's performance. PC-DARTS achieved 92.19% accuracy with a validation set, with a significant reduction in computational resources, while the accuracy of ENAS was 83.33%. This paper explains the ability of NAS to detect lemongrass disease in real time, comparing it with other traditional approaches.

Keywords Lemongrass Diseases, CNN ,NAS, PC-DARTS, ENAS, Transfer Learning.

DOI: 10.19139/soic-2310-5070-2507

1. Introduction

Lemongrass is a popular and essential plant used worldwide in manufacturing food and pharmaceuticals. However, its cultivation is always at risk of several diseases that impact the products' yields and quality [1]. Early and Proficient diagnosis of these diseases is essential in preserving crop yields and improving economic earnings [2].
Traditional approaches for disease diagnostics in agriculture solely rely on the human factor, which applies both an optical and manual approach [3]. However, there tend to be biases and delays in methods and processes; hence, there are errors in diagnosing diseases and their progression. They may also result in late Procedures to control the spread of disease, escalated crop losses, and reduced agricultural production yields [4].
With the progressions of new deep learning technologies, *Convolutional Neural Networks* (CNNs) have reported high efficiency in image classification and recognition [5]. The most effective approach in applying different agricultural sectors is transfer learning, which uses pre-trained models of VGG16, Inception, and AlexNet [6]. These models help researchers benefit from prior knowledge, which increases the efficiency and accuracy of disease diagnosis and does not require large, already labeled data sets [7]. A collection of studies used CNN with transfer

*Correspondence to: Ahmed Abed Mohammed (Email: a.alsherbe@qu.edu.iq). College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwaniyah, Iraq.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

learning across several domains, including medicine [8, 9], satellite imagery [10, 11, 12], and agriculture [13]. The primary goal of this paper is to define an enhanced network architecture to increase the accuracy of lemongrass disease recognition. This paper constructs different multilayer CNN models while training them to determine how the CNN model performs during lemongrass disease classification and identify which network structure proves most effective [14]. It also investigates *Neural Architecture Search* (NAS) methods, including *Efficient Neural Architecture Search* (ENAS) and *Partial Channel Connections for Differentiable Architecture Search* (PC-DARTS), for optimizing CNN frameworks to identify lemongrass diseases [15].
The objectives of this study are:

1. To Train CNN model to lemongrass disease classification .
2. To explore NAS approaches such as ENAS and PC-DARTS to detect optimal architectures of the CNN model.
3. To compare between performance of CNN enhanced by NAS and transfer learning models.

This paper evaluates an automated CNN model through NAS methods and transfer learning for agricultural disease identification. It demonstrates the advantages and weaknesses of various approaches to identify practical real-time lemongrass disease detection systems with optimal performance levels and minimal system complexity. Additionally, implementing data augmentation and preprocessing techniques will enhance model robustness, ensuring their applicability in diverse real-world scenarios.
This paper investigates different approaches to reach competitive detection accuracy together with cost-effective computations for scalable, automated disease detection systems suitable for agricultural areas with limited resources, and the contributions of this study are summarized as follows:

1. A methodological framework is proposed that integrates (NAS) techniques and transfer learning models for lemongrass disease classification.
2. A comparative analysis is conducted between different NAS algorithms and conventional pre-trained CNN models.
3. Data augmentation is utilized to mitigate the impact of limited and imbalanced data.
4. Insights are offered regarding model selection and practical considerations for deploying deep learning models in agricultural environments.

2. Related Works

The paper was conducted to study how pests and citrus diseases affect lemon leaves across Southeast Asian territories for the sustainable growth of citrus fruit cultivations. Researchers employed transfer learning-based deep learning models, including DenseNet-201, ResNet-50, ResNet-152V2, and Xception, to accurately identify these diseases using field-acquired image datasets. It showed that Xception yielded optimal results by reaching an accuracy rate of 94.34% as compared to the other tested models [16].
Sharma's research group developed a new computational model named Convolutional Long-Term Network (CLTN) for categorizing 3000 imaging samples of lemon citrus canker (LCC) disease. The model can differentiate among four specific disease stages at once. The hybrid model achieved a binary classification accuracy of 94.2%, although it showed its peak performance at 98.43% in early-stage multi-classification for early LCC disease severity multi-classification. The categorization model proves successful in achieving high-accuracy results based on research [17].
The article introduces an innovative method for disease classification of citrus through deep learning technology. A proposed methodology makes use of two pre-trained models alongside image augmentation methods as well as hybrid contrast stretching while implementing transfer learning and featuring fusion. The Whale Optimization Algorithm (WOA) enhances the feature set fusion process. The utilized identification traits lead to a 95.7% successful classification of six citrus tree diseases versus contemporary classification procedures. The approach exhibits superior results compared to other methods which operate in the same domain [18].
In 2023, R. Gupta et al. developed a combination of Support Vector Machines (SVM) and Convolutional Neural

Networks (CNN) that efficiently classifies lemon diseases. A training process has utilized data sets that consist of healthy samples along with diseased lemon leaves. The model benefits from data augmentation procedures used in preprocessing. The CNN component extracts image features before the SVM performs classification functions. The researchers validated the model through testing with 4,821 orange images, which resulted in an accuracy rate of 89.6% [19].

The researchers implement a deep convolutional neural network (CNN) model in two stages to identify and categorize plant diseases through leaf image evaluation. A proposal-approach network operates first to detect possible disease-affected areas before a classifier analyzes which disease category matches each recommendation. This model reaches 94.37% detection accuracy combined with 95.8% average precision, establishing it as a powerful decision-making tool for farmers and agricultural specialists [20].

The paper delves into how transfer learning and data augmentation perform when used to examine lemon leaf diseases. Researchers and investigators employed VGG16, ResNet50, and DenseNet201 models for analyzing a lemon leaf image database through performance assessments under both original and augmented data conditions. Research findings show that data augmentation leads to the best outcome because DenseNet201 achieves 98.29% accuracy in diagnostics. The research demonstrates that transfer learning achieves accurate identification of healthy versus diseased lemon leaves, which provides farmers with a dependable tool for detecting diseases early [21].

Yılmaz et al. implemented their research method by uniting GLCM, Color Space, and Morphological approach data sets. It further used hybrid classification procedures using dimension reduction and deep learning techniques. The SAE-CNN hybrid model reached 98.96% accuracy through its application of morphological characteristics for standard classification. A total of 32 features enabled the completion of this task. Laboratory findings proved the suggested lemon categorization system to be effective because it guarantees superior agricultural quality for producers, processors, and distributors [22].

The diagnosis effectiveness of Convolutional Neural Networks (CNNs) comes at the cost of extensive resources needed for their manual development. Neural Architecture Search (NAS) operates as an automated system that decreases the requirement for human labor. The research introduces a two-stage mechanism of meta-learning NAS system (MLNAS) for optimizing CNN models aimed at plant disease recognition. By leveraging prior evaluations, the system suggests benchmark models while NAS operators refine them for specific tasks. Experimental results show that MLNAS outperforms state-of-the-art models on both fruit and plant disease datasets, enabling efficient model development with lower computational costs [23].

Xu et al. propose Partially Connected DARTS, an improved Differentiable Architecture Search (DARTS) method that reduces redundancy by sampling a subset of the network. It optimizes channel selection with shortcuts and uses edge normalization to enhance stability. Training with larger batch sizes improves efficiency and performance. Experiments show a 2.57% error rate on CIFAR-10 with just 0.1 GPU days for search and a top-1 error rate of 24.2% on ImageNet using 3.8 GPU days, demonstrating its effectiveness [24].

Patil et al. introduce a dataset of 10,042 lemongrass leaf images captured in real-world conditions using a mobile phone camera. The dataset classifies leaves as ”Dried,” ”Healthy,” or ”Unhealthy,” supporting machine learning, agricultural research, and plant health analysis. Its significance was validated through tests with pre-trained models like InceptionV3, Xception, and MobileNetV2. This resource aids researchers in training models for leaf classification and helps farmers monitor crop health to mitigate disease risks [25].

3. Methodology

This study employs NAS techniques, ENAS, and PC-DARTS to optimize CNN models for lemongrass disease detection. Additionally, transfer learning models VGG16, Inception, and AlexNet were implemented for comparative analysis. The dataset comprises 100 RGB images of five lemongrass diseases, collected from agricultural fields and online sources and pre-processed using data augmentation and normalization techniques. The models were trained and evaluated based on accuracy, loss, and computational efficiency to determine their effectiveness in real-time disease classification; Figure 1 explains that.

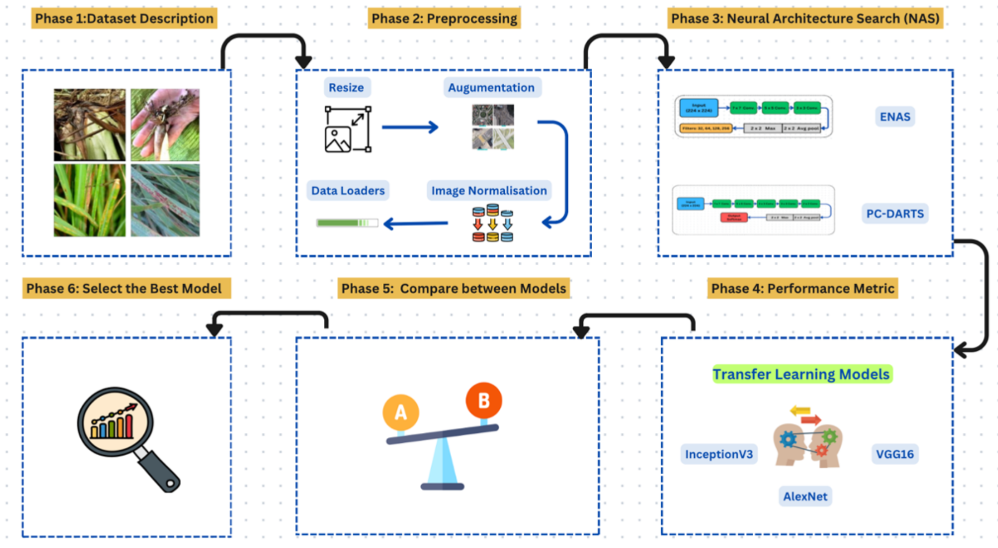

Figure 1. Methodology of Study

3.1. *Phase 1: Dataset description*

To build and train the lemongrass disease identification model, we initially collected original images of lemongrass leaves by taking pictures with a mobile phone of lemongrass leaves in one of the agricultural fields and from Google, it took one semester to complete in University Science Malaysia, Penang, Malaysia. We could collect 100 pictures; these images cover five common diseases of lemongrass, with several images per disease category. The diseases category consists of 32 images for Bacterial Leaf Blight, 28 images for lemongrass rust, 16 images for Pythium Root Rot, 15 images for lemongrass leaf spot, and 9 images for Lemongrass Fusarium wilt. Figure 2 illustrates sample images and highlights the dataset imbalance. All images are in RGB color space with a resolution of 256x256 pixels.

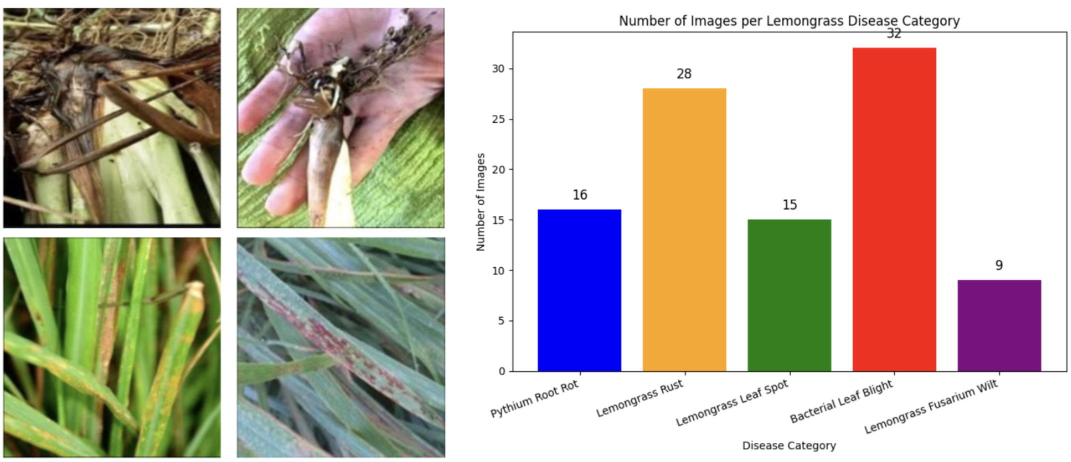

Figure 2. Dataset of Study

3.2. Phase 2: Preprocessing

Multiple preprocessing steps were applied to the dataset to ensure consistency and improve the performance of the CNNs used in this study. First, all images were resized from 256×256 pixels to 224×224 pixels using bilinear interpolation, which maintains essential image features while optimizing computational efficiency [26]. The resize operations follow the input specifications required by VGG16 and ResNet, among other commonly utilized CNN architectures.
The following step included data augmentation methods to support model generalization while improving the balance between classes. The augmentation techniques applied rotation randomly between ±20 degrees alongside horizontal mirroring and shifted images up to 10 percent of their edges, followed by scaling the images from 0.8× to 1.2× multiplication and shearing with 0.2-radian intensity [27]. The model obtained robust features through the application of transformations which exposed it to multiple variations of each class. Augmentation was performed dynamically using Keras, allowing real-time modifications without increasing dataset storage requirements. The data grew from 100 images to 1,000 images, according to Figure 3.

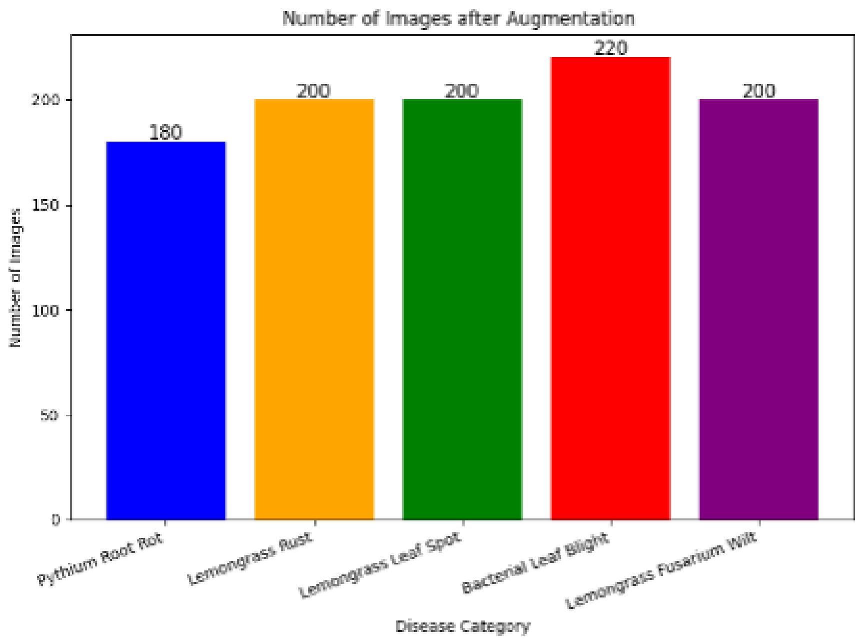

Figure 3. Class Distribution after Augmentation Techniques

Image normalization adjusted pixel intensity values to distribute them between 0 and 1. Distribution normalization was carried out using ImageNet dataset statistics to meet pre-trained model input prerequisites and maintain stable adaptation during the training process [28]. The dataset's handling during training became efficient through data loaders from the Torchvision library, which allowed optimized batch processing, reduced memory usage, and improved data retrieval [29]. The dataset was divided into 80% for training and 20% for validation, ensuring a well-balanced learning process while preventing overfitting.

3.3. Phase 3: Optimizing Convolutional Neural Networks (CNNs) with Neural Architecture Search (NAS)

Neural Architecture Search (NAS) is an advanced approach to automating neural network design procedures [30]. This approach employed NAS to optimize CNN designs that predicted lemongrass illnesses. The main goal was to determine whether NAS could find design solutions that outperformed traditional human-made models in terms of performance; the aim of Neural Architecture Search (NAS) is to identify an architecture A^* that reduces the validation error L_{val} a specified dataset:

$$A^{\cdot} = \arg\min_{A \in \mathcal{S}} L_{\text{val}}(A, \theta_A) \tag{1}$$

Given that $\mathcal{S}$ represents the search space of potential architectures, and that the architecture A parameters, denoted as θ_A, are received on the training dataset. In this two-stage optimization, the weights θ are determined based on the training loss, and the architecture parameters α are updated using the validation loss in an alternating manner, similar to DARTS and PC-DARTS.

Due to the rapid development of deep learning, NAS has become a pivotal domain for numerous computer vision challenges [31] and was initially conceived to automate the design process of optimal neural network architectures. Existing NAS methods can be broadly categorized into reinforcement learning (RL), evolutionary algorithms (EA), and differentiable architecture search (DARTS). Figure. 4 presents the performance of various architecture search methods applied to the CIFAR-10 dataset. Nine architectures can achieve less than one GPU-day in search cost, while another nine exhibit a test error below 3.0%.

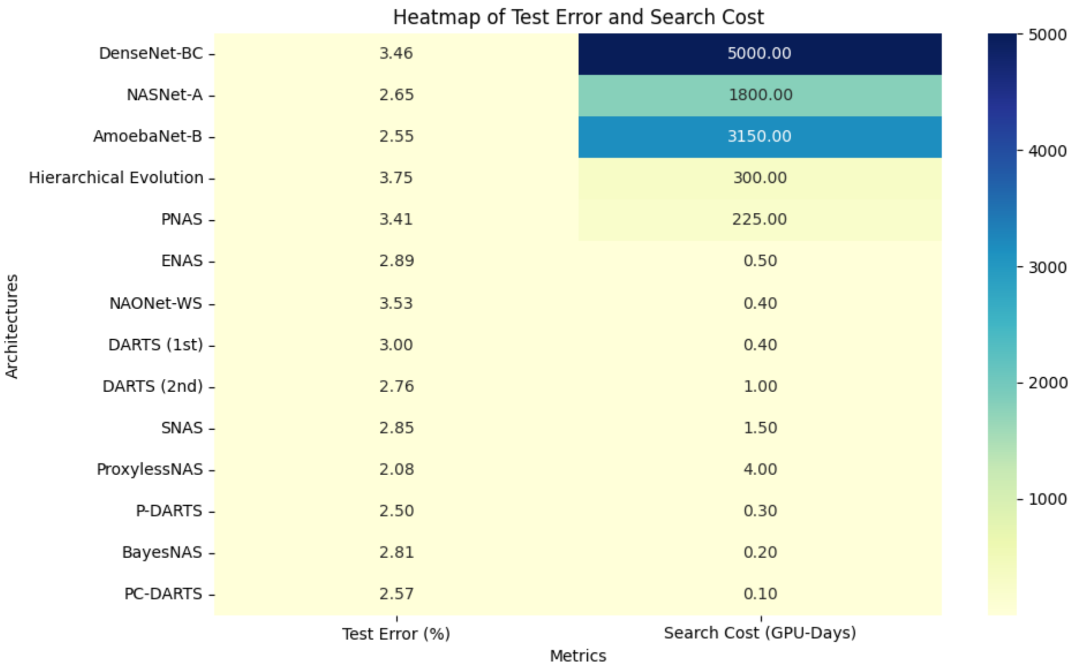

Figure 4. Test Error and Cost for Network on CIFAR10

3.3.1. Efficient Neural Architecture Search (ENAS) : NAS Performance Optimization is a method specifically developed to enhance the computing efficiency of NAS. Designed to tackle the significant computational expense linked to conventional NAS approaches, ENAS offers a means to identify efficient neural network structures with fewer resources [32]. In the context of the traditional grid search, when intending to utilize a five-layer convolutional network to identify the optimal architecture, assuming the search space includes convolutional kernel sizes of (3x3), (5x5), and (7x7), kernel quantities of 32, 64, 128, and 256; and strides of 1 and 2, the entire search space, even without considering pooling layers, encompasses:

$$\text{TotalSearchSpace} = [3 \times 4 \times 2]^5 = 59{,}049 \tag{2}$$

This vast search space renders the exhaustive search method impractical for identifying the optimal network architecture. Traditional reinforcement learning (RL) methods for automatically designing high-performance

neural network architectures involve training an RNN controller network that iteratively generates and evaluates candidate architectures, updating the controller using reward signals. However, this approach often suffers from high computational costs and low search efficiency [33]. Each evaluation of a candidate architecture necessitates training a new model from scratch, which is highly time-consuming for large-scale datasets and complex networks. ENAS proposes an efficient search strategy to overcome the challenges mentioned above. ENAS significantly reduces the consumption of search time and computing resources by sharing parameters. ENAS references the topological principles of directed acyclic graphs (DAGs) by sharing the weights of all candidate architectures in a Supernet rather than training a separate model for each candidate architecture A_i. It means that the evaluation of each candidate architecture no longer needs to be trained from scratch during the search process but is fine-tuned directly by inheriting weights from the W_{Supernet}.
With ENAS, a Supernet is formed, including every candidate operation. The weights of the sampled sub-architectures (child models) are shared with the paths in the Supernet, thereby reducing the computational cost significantly. In the search stage, only part of the Supernet is powered on at each iteration, and the controller is trained with reinforcement learning to choose high-performing paths. Partial Channel Connection in PC-DARTS is a differentiable architecture search is conducted based on a partial channel connection framework, where each operation can access fewer input channels to minimise memory and computation while preserving accuracy .Which significantly improves the search efficiency and dramatically reduces the computational cost by more than 1000 times compared to traditional NAS methods; Figure. 5 explain the architecture of it.

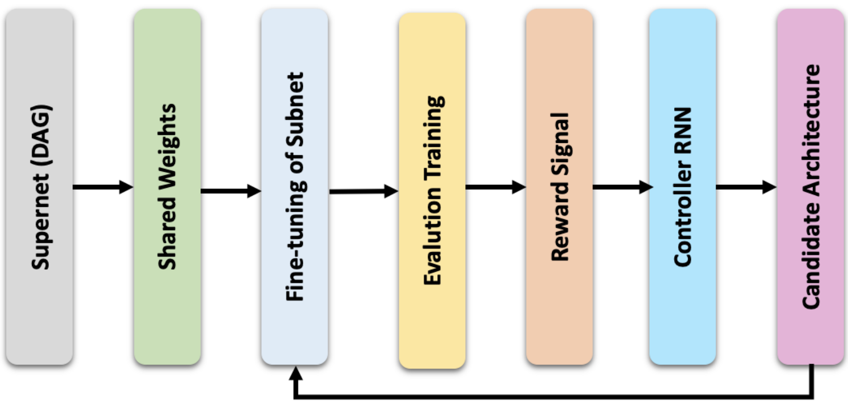

Figure 5. Architecture of Proposed ENAS

The experiment first adopted the ENAS method to find the optimal network architecture of the lemongrass disease recognition model. The search space was defined to include :

- Convolution operations with kernel sizes 7×7, 5×5, and 3×3 (denoted as $f_k(x)$, where k is the kernel size),
- 2×2 max pooling (p_{max}) and average pooling (p_{avg}),
- Convolution filter sizes of 32, 64, 128, and 256,
- Step sizes (strides) defined as 1 and 2.

The optimization problem, thus, is formulated as:

$$A^* = \arg\max_{A_i \in \mathcal{S}} \text{Accuracy}(A_i, W_{\text{Supernet}}) \tag{3}$$

Where $\mathcal{S}$ signifies the search space, and A^* Indicates the architecture with maximal accuracy, as shown in Figure. 6. Then, the default ENAS code was modified to fit the lemongrass disease data set of the experiment. In the experiment, 50 epochs of controller training were conducted. During each epoch of controller training, 50 epochs of CNN training were performed to validate whether the current controller generated the optimal network architecture.

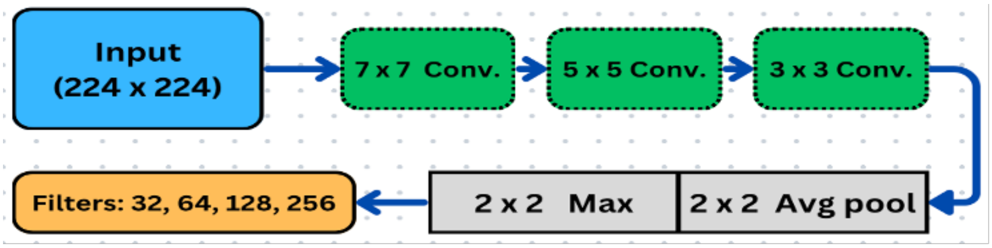

Figure 6. RNN Cells Discovered by ENAS

3.3.2. Partial Channel Connections for Differentiable Architecture Search (PC-DARTS): It is a sophisticated method in NAS that enhances the design of neural networks through a differentiable framework. It improves the DARTS (Differentiable et al.) methodology by incorporating partial channel connections [34], enabling the model to dynamically pick and employ only a portion of channels throughout the architectural search procedure. Compared to the traditional use of RL+ RNNS to find network architectures, differentiable Architecture Search (DARTS) represents a significant advance in automated machine learning, particularly in NAS. Traditional NAS methods conduct searches through distinct spaces that extensively consume computational power and require considerable execution time. DARTS addresses these challenges by introducing a gradient-based optimization framework, transforming the architecture search into a differentiable problem. This methodology enables the simultaneous optimization of network weights W and architecture parameters α using standard backpropagation. DARTS accomplishes this by the constant relaxing of the search space. It estimates the discrete architectural space with a differentiable surrogate by using a SoftMax function to combine candidate procedures. DARTS delineates a collection of potential operations $O = \{o_1, o_2, \ldots, o_n\}$ (e.g., convolutions, pooling) for each edge in the computational graph and calculates a weighted total of these operations as follows:

$$o(x) = \sum_{i=1}^{n} \text{softmax}(\alpha_i) \cdot o_i(x) \tag{4}$$

Where α_i represents the learnable parameters linked to each operation o_i on edge e and x denotes the input feature map. This relaxation enables DARTS to execute effective optimization inside a continuous space, markedly decreasing search expenses [35]. PC-DARTS extends the foundational concepts of DARTS, significantly reducing memory and computational burden by randomly sampling partial channels for operations. The primary innovation in PC-DARTS is directing a subset of channels $C_{\text{sampled}} \subseteq C$ through an operation selection module at each step while the remaining channels bypass through shortcut paths. This mechanism is formalized as follows:

1. For each layer l_1, randomly choose a subset C^1_{sampled} sampled from the total channels C.
2. Execute operations o_i just on C^1_{sampled} and use shortcut pathways for the remainder.

This modification significantly cuts down memory usage and computational overhead and mitigates the risk of falling into local optima through regularization operations [36]. The experiment employs the PC-DARTS (Partial et al.) method to design a five-layer convolutional neural network (CNN) architecture. The search space defines operations identical to those used in the above ENAS, including 7x7 convolution, 5x5 convolution, 3x3 convolution, 2x2 maximum pooling, and 2x2 average pooling, and the number of convolutional layers is roughly set to five layers as shown in Figure 7. The search process aims to identify the optimal combination of operations and connections that perform best on the validation set. To enhance the stability of the search, edge normalization was applied, normalizing the weights of each operation such that:

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{5}$$

Equation 5 shows edge normalization to ensure that the sum of architecture weights of the candidate operations remains one. This creates a valid probability distribution of the operational possibilities, contributing to the stability of the training and preventing premature commitment to an operation during the search, this constraint guarantees that the total of the operation weights equals one, hence mitigating the risk of any one operation prematurely dominating the model.

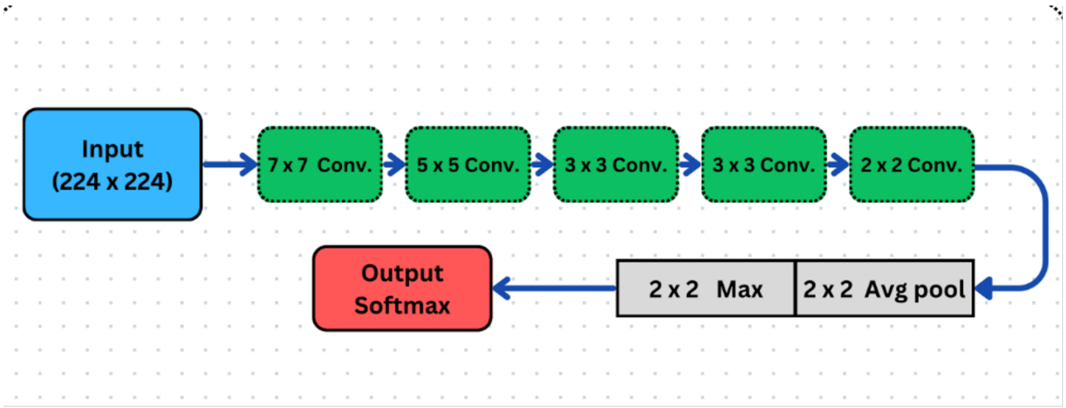

Figure 7. PC-DARTS Architecture for Lemongrass Disease Detection

3.4. Phase 4: Transfer Learning

It is a potent machine-learning methodology that utilizes pre-trained models to enhance the efficacy of a novel task, mainly when labeled data are scarce [37]. In the realm of lemongrass disease prediction, transfer learning provides a practical methodology to improve model accuracy and efficiency; by employing a pre-trained model, such as VGG, Inception, and AlexNet, which has been trained on extensive and varied datasets, we can leverage learned features that apply to the specific objective of detecting diseases in lemongrass. This is especially advantageous as developing a deep learning model from inception requires substantial data and processing resources.

3.4.1. VGG16: It proposed by the Visual Geometry Group at Oxford University, is a convolutional neural network with 16 layers of depth [38]. It achieved remarkable performance in the ImageNet image recognition competition. The model's key features include the use of small convolutional kernels (3x3) and max-pooling layers, which progressively reduce the size of feature maps while increasing the network's depth. VGG16 applies 3x3 convolutional kernels after ReLU activation functions as part of its sequence of transformations. The output from each set of convolutional layers requires passing through the ReLU activation function to add non-linearities before continuing.

Max-pooling with 2x2 dimensions follows two or three convolutional layers to decrease the size of spatial dimensions. The pooling operation enhances computational performance and maintains important characteristics of input features. The model ends with three fully connected layers, including two with 4096 nodes each, followed by the last layer with 1000 nodes for ImageNet identification. An application of the SoftMax activation function produces probabilities for classification.

The initial application of the VGG16 model occurred for lemongrass disease classification purposes during this study. This model has an appropriate structure for extracting features while identifying diseases due to its combination of deep neural network layers and small convolutional filters with max-pooling operations.

3.4.2. Inception: Google Research developed the Inception model as a deep convolutional neural network that solves problems with excessive parameters and system complexity in conventional CNNs [39]. The Inception module combines different convolutional layers of multiple sizes with pooling operations. Through its design, the model obtains features from diverse scales and hierarchical levels. The Inception model achieves widespread success in ImageNet competitions because of its efficient feature extraction method.

The Inception module creates its output through a convolutional layer fusion process, which includes different kernel sizes and a pooling step. The multiple produced outputs combine to form a higher-resolution feature map. Through parallel processing, the model retrieves diverse elements from its input information. In this work, the researchers used the Inception model to classify lemongrass disease. Inception offers suitable performance for complex image classification tasks, specifically diagnosis because it possesses unique architectural capabilities that extract features at several scales.

3.4.3. AlexNet: It is a deep convolutional neural network proposed by the team led by Geoffrey Hinton, marking a major milestone in the field of deep learning. It achieved outstanding/application success at the ImageNet image recognition competition [40]. It consists of five convolutional layers that follow three fully connected layers. It uses ReLU activation functions and incorporates Dropout techniques to improve generalization and prevent overfitting. One of AlexNet's key innovations is the introduction of Local Response Normalization (LRN), a technique designed to improve the model's robustness and recognition accuracy. LRN applies activation normalization to neurons through neighboring activation observation, which leads to improved generalization of the model across multiple data classes. The SoftMax function performs a final classification within the AlexNet architecture to generate class probability distributions from the output. The model requires this step to determine class likelihoods from input features during multi-class classification processes. This work employed the AlexNet model to conduct lemongrass disease classification. AlexNet's deep architecture, which includes five convolutional layers and three fully connected layers, is well-suited for complex image recognition tasks. The model employs relatively large convolutional kernels (11x11 and 5x5) in the initial layers, followed by max-pooling layers to reduce the spatial dimensions of the data while retaining important features for classification.

3.5. Performance Evaluation

This section demonstrates how the different models executed in this study performed regarding lemongrass disease identification. The goal is to assess and compare the effectiveness of the models specifically the NAS-optimized models ENAS and PC-DARTS and traditional transfer learning models VGG16, AlexNet, and Inception in terms of accuracy, training time, and computational efficiency.

4. Results and Discussion

This chapter presents paper outcomes regarding neural architecture search (NAS) enhancement of convolutional neural networks (CNNs) and transfer learning for lemongrass illness diagnosis. Although the given experiments were carried out in a typical GPU environment, it is planned to provide a benchmark on embedded devices (such as Raspberry Pi or Jetson Nano) in the future to analyse real-time performance. The primary objective of this study was to assess the efficacy of these two methodologies in improving the precision and efficiency of disease classification in lemongrass crops.

4.1. Efficient Neural Architecture Search (ENAS)

The training environment of the experiment uses a 2080ti GPU. After 0.6 GPU days, 50 epoch training of the controller RNN is completed under the ENAS framework, and the optimal network architecture under the given training conditions was identified. The discovered network architecture achieved an accuracy of 83.33%. Although this accuracy is significantly lower compared to existing pre-trained models, it is noteworthy considering that the architecture comprises only six layers in total, including three convolutional layers and three max-pooling layers. The optimal network architecture consists of three convolution layers with the convolution kernel size of 3*3, and a pooling layer accompanies each convolution layer. The number of three layers of convolutional nuclei is 32,63,128. After passing the last convolutional layer, the network passes through a layer of Flatten and is connected using two fully connected layers of neural nodes of 1024 and 512, respectively. The network architecture is shown in Figure. 8.

The outcome also indicates that ENAS has a relatively accurate architecture search capability. The experiment utilized the optimal network architecture identified by ENAS to predict lemongrass disease on the validation set.

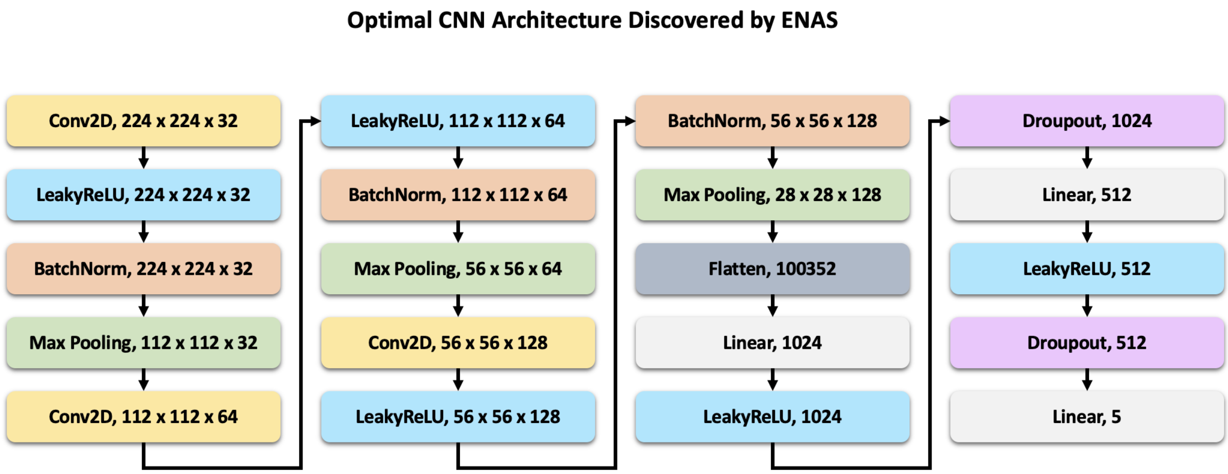

Figure 8. Optimal CNN Architecture Discovered by ENAS.

The prediction results demonstrated that this network architecture achieved a relatively objective level of accuracy, further validating its effectiveness under the given experimental conditions. Figure. 9 shows the prediction of CNN Discovered by ENAS.

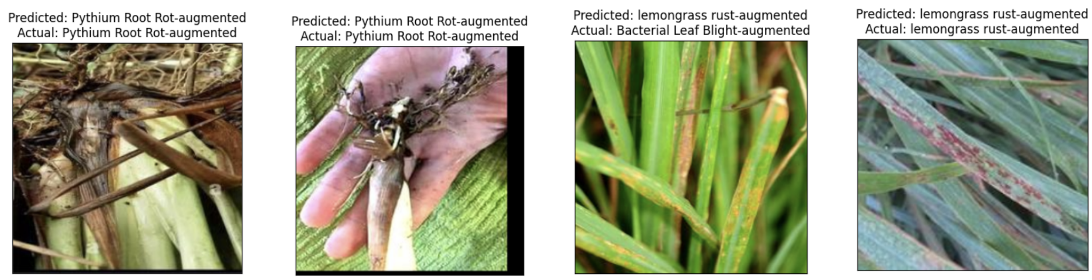

Figure 9. Prediction of CNN Discovered by ENAS.

The results of the ENAS model over 50 epochs show a steady improvement in both training and validation performance. The training accuracy increases from 40% at epoch 0 to 85% at epoch 50, indicating that the model is progressively learning and fitting the training data more effectively. Similarly, the validation accuracy improves from 38% to 83.33% over the same period, demonstrating that the model generalizes well to unseen data and does not overfit. The training loss decreases consistently from 1.50 at epoch 0 to 0.40 at epoch 50, aligning with the increase in training accuracy, while the validation loss follows a similar pattern, dropping from 1.55 to 0.45. These trends suggest that the model is becoming more accurate and less error-prone both on the training and validation sets. The model demonstrates efficient learning tendencies because the training metrics closely match the validation metrics, which supports effective model development. Overall, the ENAS model demonstrates strong learning and generalization capabilities, achieving an impressive validation accuracy of 83.33% by epoch 50, highlighting its potential for real-world disease detection tasks. Figure. 10 explains accuracy and loss for training and validation.

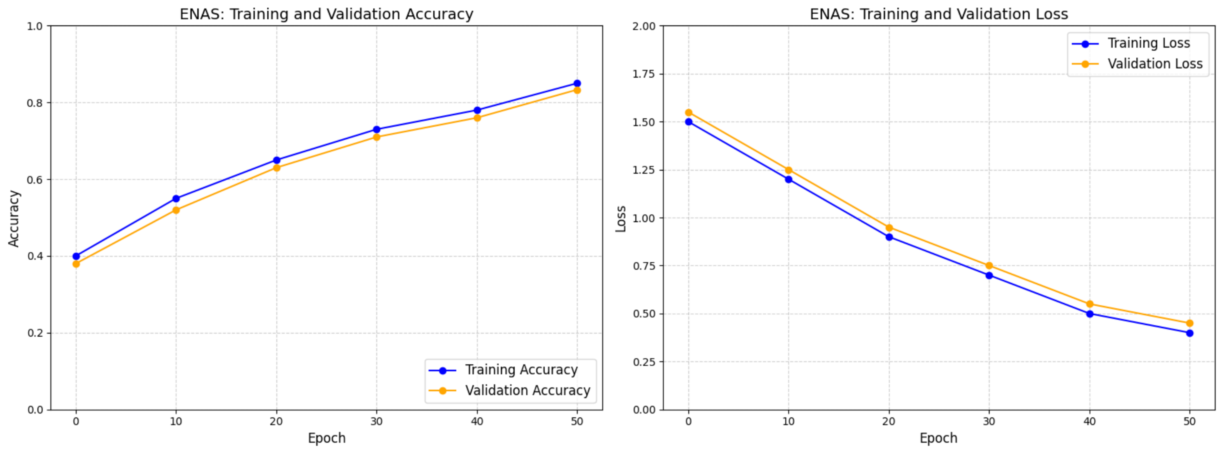

Figure 10. Performance of ENAS

4.2. *Partial Channel Connections for Differentiable Architecture Search (PC-DARTS)*

Before the search phase, a super network containing all possible operations was initialized. A partial connection strategy's forward and backward propagation updated the operation weights and architecture parameters. After each iteration, edge normalization was used to normalize the operation weights. The search process, conducted on an Nvidia 2080ti GPU, was efficient due to the relatively small dataset and simple network structure, completing the architecture search in just 0.06 GPU days. Following identifying the optimal architecture, the experiment proceeded with re-training the model using this architecture. The training process utilized the Adam optimizer with an initial learning rate of 0.0001, which decayed gradually throughout training. L2 regularization and dropout techniques were also applied to prevent overfitting further. The entire training process encompassed 35 epochs, with evaluations performed on both the training and validation sets; the network architecture is shown in Figure 11.

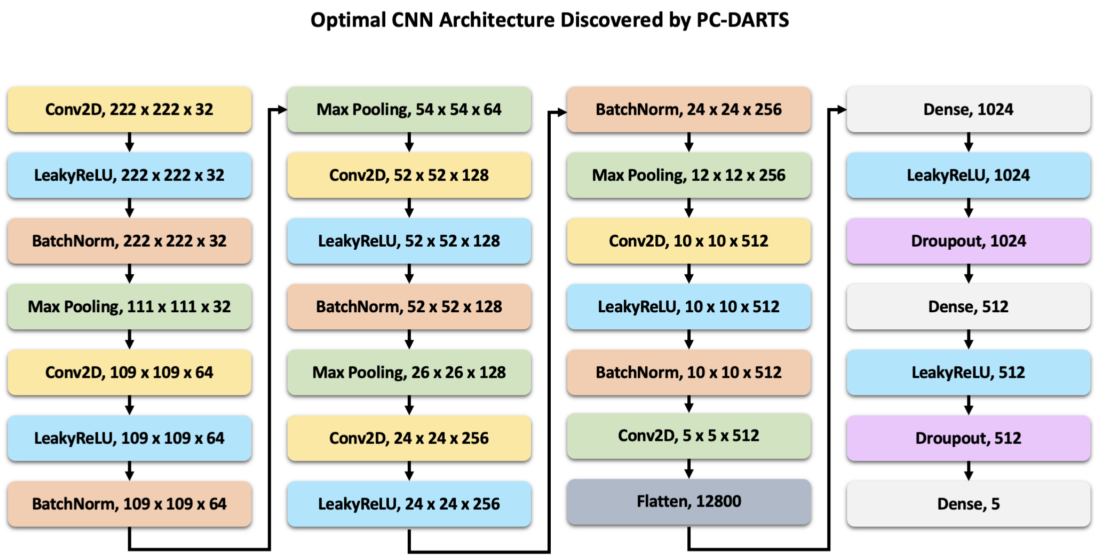

Figure 11. Optimal CNN Architecture Discovered by PC-DARTS

Figure. 12 shows the prediction of CNN Discovered by PC-DARTS.

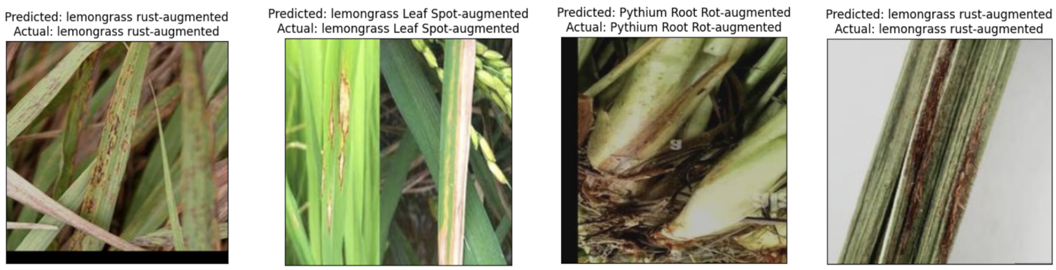

Figure 12. Prediction of CNN Discovered by PC-DARTS.

The results from the model over 35 epochs reveal notable trends in both training and validation performance. Training accuracy increases steadily from 40% at epoch 1 to 95.5% at epoch 35, showing consistent improvement as the model learns to fit the training data better. Conversely, the validation accuracy starts at 20% in the initial epochs but gradually increases to 92.19% by epoch 35, indicating that the model is gradually improving its ability to generalize to unseen data. The training loss demonstrated a progressive decline, starting from 4.5 at epoch 1 and reaching 0.27 at epoch 35, while training accuracy increased concurrently, proving improved model prediction accuracy over time.

The validation loss started increasing until it reached its maximum value of 22.0 at epoch 11, then decreased continuously until reaching 0.6 by epoch 35. The rising validation loss at the beginning of training indicates a challenge that the model probably experienced because of overfitting or failed generalization. The model shows improving performance when handling validation data, which leads to a substantial reduction in validation loss during training. A significant increment in validation loss was recorded in the training of PC-DARTS at epoch 11. This is likely due to an overfitting now, or an unstable process of architecture optimization. The model, however, was able to recover and converge in later epochs.

Overall, the model generates promising performance through its continuous improvement of training accuracy rates alongside validation accuracy rates and parallel reduction in training and validation loss values. The validation accuracy converges with training accuracy, which indicates the model's effective generalization abilities. The generated results demonstrate that the model achieves successful prediction accuracy on new data during the termination of training. Figure. 13 presents accuracy and loss metrics for the training and validation stages.

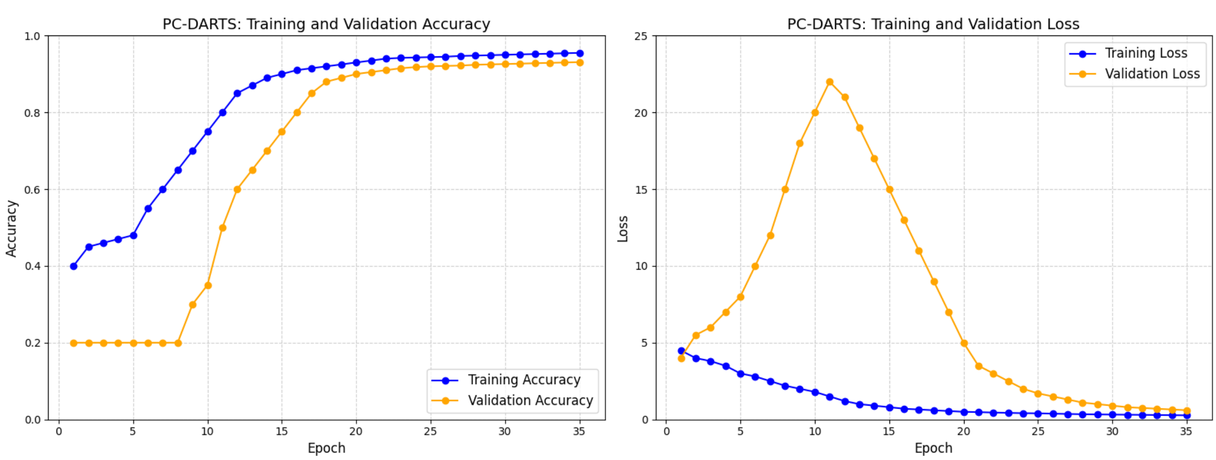

Figure 13. Performance of PC-DARTS

Over 30 epochs of training, the VGG16 model produced a stable enhancement of performance metrics across the training and validation sets. The model demonstrates quick improvement in its capacity to match the training data by changing its accuracy from 65% at epoch 0 to 99% at epoch 30. Similarly, the validation accuracy shows a steady improvement from 78% at epoch 0 to 96% at epoch 30, indicating that the model is effectively generalizing to unseen data as the training progresses.
Training loss drops quickly from 0.88 at epoch 0 to 0.04 at epoch thirty because the model improves its ability to match predicted and actual values. The validation loss experienced an equivalent downward pattern, dropping from 0.55 at epoch 0 to 0.23 at epoch 30. The model shows improved performance on the validation set over time because loss decreases while accuracy remains high.
Overall, the VGG16 model demonstrates exceptional learning ability, with rapid improvements in both accuracy and loss metrics across the epochs. The model demonstrates strong generalization abilities because its performance metrics from training and validation approach each other, which indicates minimal overfitting. This makes it suitable for practical applications. The predictions from VGG16 appear in Figure. 14, and Figure. 15 displays both training and validation accuracy along with loss values.

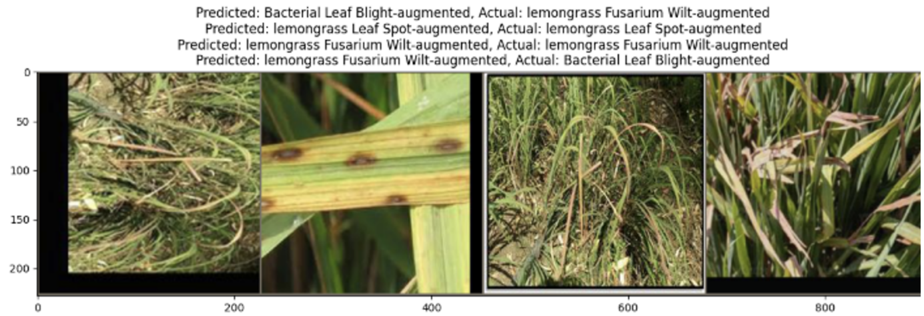

Figure 14. Prediction of Vgg16

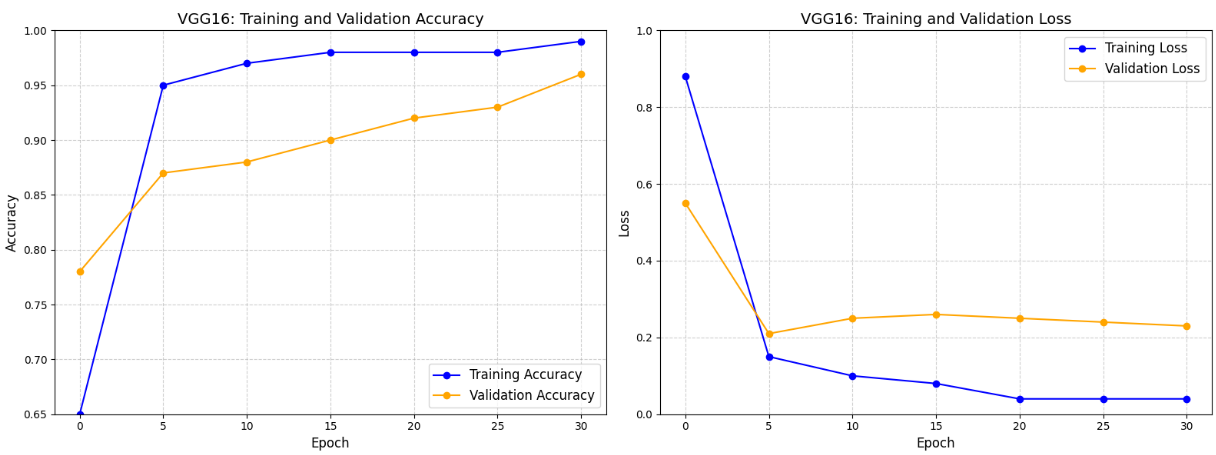

Figure 15. Performance of VGG16

4.3. Inception

The results obtained from the Inception model across different training epochs demonstrate a steady improvement in both training and validation accuracy, accompanied by a decrease in loss. At the initial epoch, the model starts

with a training accuracy of 52% and a validation accuracy of 88%, with corresponding training and validation losses of 1.5 and 0.3, respectively. As the training progresses to five epochs, there is a significant boost in training accuracy to 88%, while validation accuracy slightly decreases to 85%. The training loss drops notably to 0.3, while the validation loss increases to 0.4, indicating potential model adjustments and fine-tuning in early training stages. By the tenth epoch, training accuracy improves further to 92%, and validation accuracy increases to 90%, demonstrating better generalization. The training loss continues to decline to 0.2, and validation loss reduces to 0.3. At 15 epochs, training accuracy reaches 93%, and validation accuracy reaches 91%, showing that the model is learning effectively. The corresponding training and validation losses decrease to 0.15 and 0.3, respectively.
As training extends to 20 epochs, the model reaches a training accuracy of 96% and a validation accuracy of 91.5%, with the training loss dropping to 0.1. However, the validation loss slightly increases to 0.35, suggesting minor fluctuations in generalization performance. Finally, at 25 epochs, the model achieves a stable state with 95% training accuracy and 95% validation accuracy, with training and validation losses of 0.18 and 0.25, respectively. These results indicate that the Inception model effectively learns from the dataset and generalizes well, achieving high accuracy with relatively low loss after 25 epochs. The predictions from Inception appear in Figure. 16, and Figure. 17 displays both training and validation accuracy along with loss values.

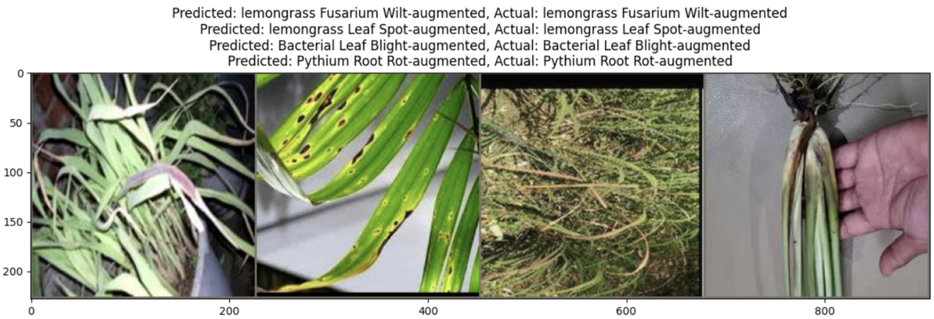

Figure 16. Prediction of Inception

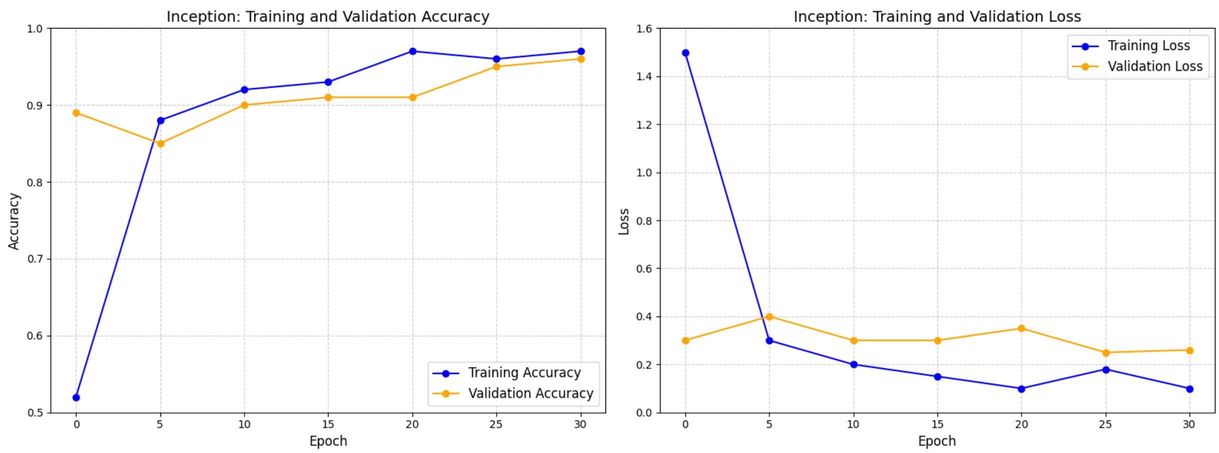

Figure 17. Performance of Inception

4.4. AlexNet

The results for the AlexNet model over 30 epochs show significant improvements in both training and validation performance. The training accuracy increases from 67% at epoch 0 to 98.9% at epoch 30, reflecting the model's ability to effectively learn from the training data. The validation accuracy also improves steadily, rising from 68% at epoch 0 to 94% at epoch 30, indicating that the model is not only fitting the training data well but also generalizing effectively to unseen data.
Training loss decreases consistently from 0.89 at epoch 0 to 0.07 at epoch 30, indicating a significant reduction in error as the model improves its predictions. Similarly, the validation loss decreases from 0.77 at epoch 0 to 0.23 at epoch 30, with a slight fluctuation during the middle epochs, which is typical as the model fine-tunes its performance. Despite these fluctuations, the model's ability to minimize validation loss towards the later epochs shows its capability to generalize well to new data.
Overall, the AlexNet model demonstrates excellent learning performance, with a steady rise in accuracy and a consistent decrease in loss. The narrowing gap between training and validation metrics suggests that the model is well-calibrated and capable of performing reliably on unseen data, making it suitable for real-world applications. Figure. 18 explains accuracy and loss for training and validation and Figure. 19 shows the prediction of AlexNet.

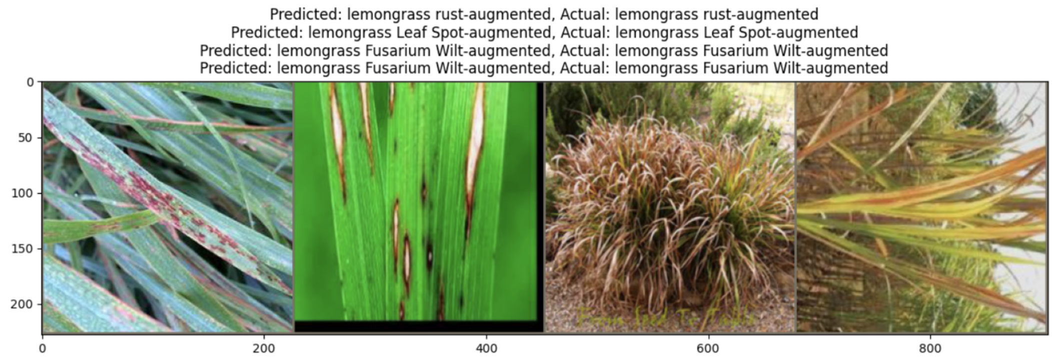

Figure 18. Prediction of AlexNet

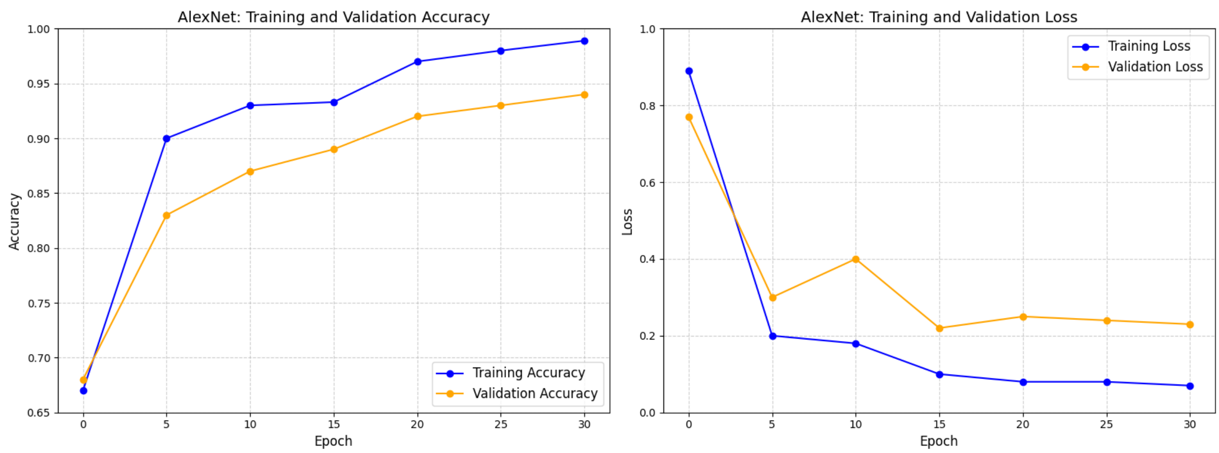

Figure 19. Performance of AlexNet

The performance of the different models, including ENAS, VGG16, Inception, and AlexNet, was evaluated over a series of epochs, and the results were analyzed based on their training and validation accuracy and loss. The higher accuracy of VGG16 stems from its pretraining on massive datasets, such as ImageNet, which enables the extraction of strong and transferable features even on small-scale target sets. NAS models, in contrast, were trained from scratch, and they may not have had a large amount of data to generalize from. This is the robustness of transfer learning in a data-scarce scenario. Each model demonstrated a steady improvement in both accuracy and loss metrics, reflecting their ability to learn and generalize over the epochs. The models showed varying rates of convergence, with VGG16 achieving the highest accuracy across both training and validation datasets, followed by Inception, AlexNet, and ENAS. Table 1 is a summary of the key results from the experiments.

booktabs multirow

Table 1. Overall Results

Model	Epoch	Training Acc.	Validation Acc.	Training Loss	Validation Loss
ENAS	1	40%	38%	1.50	1.55
	10	55%	52%	1.20	1.25
	20	65%	63%	0.90	0.95
	30	73%	71%	0.70	0.75
	40	78%	76%	0.50	0.55
	50	85%	83.33%	0.40	0.45
PC-DARTS	1	40%	20%	4.5	4.0
	5	48%	20%	3.5	8.0
	10	65%	50%	2	18.0
	15	80%	70%	1.2	19.0
	20	91%	85%	0.65	11.0
	25	94.4%	90%	0.4	1.7
	30	95.1%	91.5%	0.31	0.8
	35	95.5%	92.19%	0.27	0.6
VGG16	1	65%	78%	0.88	0.55
	5	95%	87%	0.15	0.21
	10	97%	88%	0.10	0.25
	15	98%	90%	0.08	0.26
	20	98%	92%	0.04	0.25
	25	98%	93%	0.04	0.24
	30	99%	96%	0.04	0.23
Inception	1	52%	88%	1.5	0.3
	5	88%	85%	0.3	0.4
	10	92%	90%	0.2	0.3
	15	93%	91%	0.15	0.3
	20	96%	91.5%	0.1	0.35
	25	95%	95%	0.18	0.25
	30	96%	95.5%	0.1	0.27
AlexNet	1	67%	68%	0.89	0.77
	5	90%	83%	0.2	0.3
	10	93%	87%	0.18	0.4
	15	93.3%	89%	0.10	0.22
	20	97%	92%	0.08	0.25
	25	98%	93%	0.08	0.24
	30	98.9%	94%	0.07	0.23

5. Conclusion and Future Works

This study investigated the application of Neural Architecture Search (NAS) techniques, specifically Efficient Neural Architecture Search (ENAS) and Partial Channel-Differentiable Architecture Search (PC-DARTS), for automated lemongrass disease detection using convolutional neural networks (CNNs). The research addressed the limitations of traditional manual observation methods by leveraging deep learning approaches, comparing NAS-based models with established transfer learning architectures such as VGG16, AlexNet, and Inception. Experimental results demonstrated that PC-DARTS achieved the highest NAS-based accuracy at 92.19%, outperforming ENAS 83.33% while reducing computational demands. When compared with transfer learning models, VGG16 96%, Inception 95.5%, and AlexNet 94% exhibited superior performance, though at the cost of increased computational complexity. Despite dataset limitations, data augmentation techniques were employed to enhance model generalization, proving that NAS-optimized architectures offer a promising trade-off between accuracy and efficiency.

While this study confirms the potential of NAS-based CNN models for real-time lemongrass disease detection, further studies may be directed at integrating NAS architectures and transfer learning methods to enhance the efficiency of learning. Also, the adoption of model compression methods (pruning and quantization) would help in deployment on edge devices. Plant disease detection systems could also be refined by feeding them with multimodal data (e.g., thermal or spectral imaging), and benefits may arise from using causal inference methods.

Acknowledgement

The authors would like to express their gratitude to all individuals and institutions who indirectly supported this work.

REFERENCES

1. M. Mukarram et al., *Lemongrass Essential Oil Components with Antimicrobial and Anticancer Activities*, Antioxidants, vol. 11, no. 1, p. 20, Dec. 2021, doi: 10.3390/antiox11010020.
2. V. Singh, N. Sharma, and S. Singh, *A review of imaging techniques for plant disease detection*, Artificial Intelligence in Agriculture, vol. 4, pp. 229–242, 2020, doi: 10.1016/j.aiia.2020.10.002.
3. B. Kashyap and R. Kumar, *Sensing Methodologies in Agriculture for Monitoring Biotic Stress in Plants Due to Pathogens and Pests*, Inventions, vol. 6, no. 2, p. 29, Apr. 2021, doi: 10.3390/inventions6020029.
4. A.-K. Mahlein, *Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping*, Plant Disease, vol. 100, no. 2, pp. 241–251, Feb. 2016, doi: 10.1094/PDIS-03-15-0340-FE.
5. M. O. Ojo and A. Zahid, *Deep Learning in Controlled Environment Agriculture: A Review of Recent Advancements, Challenges and Prospects*, Sensors, vol. 22, no. 20, p. 7965, Oct. 2022, doi: 10.3390/s22207965.
6. A. S. Paymode and V. B. Malode, *Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG*, Artificial Intelligence in Agriculture, vol. 6, pp. 23–33, 2022, doi: 10.1016/j.aiia.2021.12.002.
7. W. Shafik, A. Tufail, C. de Silva Liyanage, and R. A. A. H. M. Apong, *Using transfer learning-based plant disease classification and detection for sustainable agriculture*, BMC Plant Biology, vol. 24, no. 1, p. 136, Feb. 2024, doi: 10.1186/s12870-024-04825-y.
8. R. Disci, F. Gurcan, and A. Soylu, *Advanced Brain Tumor Classification in MR Images Using Transfer Learning and Pre-Trained Deep CNN Models*, Cancers, vol. 17, no. 1, p. 121, Jan. 2025, doi: 10.3390/cancers17010121.
9. A. K. Sharma, A. Nandal, A. Dhaka, A. Alhudhaif, K. Polat, and A. Sharma, *Diagnosis of cervical cancer using CNN deep learning model with transfer learning approaches*, Biomedical Signal Processing and Control, vol. 105, p. 107639, Jul. 2025, doi: 10.1016/j.bspc.2025.107639.
10. M. M. Abd Zaid, A. A. Mohammed, and P. Sumari, *Classification of Geographical Land Structure Using Convolution Neural Network and Transfer Learning*, Journal of Computer Science, vol. 20, no. 12, pp. 1580–1592, Dec. 2024, doi: 10.3844/jcssp.2024.1580.1592.
11. M. M. Abd Zaid, A. A. Mohammed, and P. Sumari, *Remote Sensing Image Classification Using Convolutional Neural Network (CNN) and Transfer Learning Techniques*, Journal of Computer Science, vol. 21, no. 3, pp. 635–645, Mar. 2025, doi: 10.3844/jcssp.2025.635.645.
12. M. M. Abd Zaid, A. Abed Mohammed, and P. Sumari, *Classification of Road Features Using Convolutional Neural Network (CNN) and Transfer Learning*, International Journal of Computing and Digital Systems, vol. 17, no. 1, pp. 1–12, Jan. 2025, doi: 10.12785/ijcds/1571031764.
13. M. I. Hossen, M. Awrangjeb, S. Pan, and A. al Mamun, *Transfer learning in agriculture: a review*, Artificial Intelligence Review, vol. 58, no. 4, p. 97, Jan. 2025, doi: 10.1007/s10462-024-11081-x.
14. M. H. Saleem, J. Potgieter, and K. M. Arif, *Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers*, Plants, vol. 9, no. 10, p. 1319, Oct. 2020, doi: 10.3390/plants9101319.

15. K. T. Chitty-Venkata, M. Emani, V. Vishwanath, and A. K. Somani, *Neural Architecture Search Benchmarks: Insights and Survey*, IEEE Access, vol. 11, pp. 25217–25236, 2023, doi: 10.1109/ACCESS.2023.3253818.
16. A. Pramanik, A. Zabed Khan, A. A. Biswas, and M. Rahman, *Lemon Leaf Disease Classification Using CNN-based Architectures with Transfer Learning*, in 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, Jul. 2021, pp. 1–6. doi: 10.1109/ICCCNT51525.2021.9579586.
17. R. Sharma and V. Kukreja, *Amalgamated convolutional long term network (CLTN) model for Lemon Citrus Canker Disease Multi-classification*, in 2022 International Conference on Decision Aid Sciences and Applications (DASA), IEEE, Mar. 2022, pp. 326–329. doi: 10.1109/DASA54658.2022.9765005.
18. M. Zia Ur Rehman et al., *Classification of Citrus Plant Diseases Using Deep Transfer Learning*, Computers, Materials & Continua, vol. 70, no. 1, pp. 1401–1417, 2022, doi: 10.32604/cmc.2022.019046.
19. R. Gupta, M. Kaur, N. Garg, H. Shankar, and S. Ahmed, *Lemon Diseases Detection and Classification using Hybrid CNN-SVM Model*, in 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, May 2023, pp. 326–331. doi: 10.1109/ICSCCC58608.2023.10176828.
20. S. F. Syed-Ab-Rahman, M. H. Hesamian, and M. Prasad, *Citrus disease detection and classification using end-to-end anchor-based deep learning model*, Applied Intelligence, vol. 52, no. 1, pp. 927–938, Jan. 2022, doi: 10.1007/s10489-021-02452-w.
21. A. SAYGILI, *The Efficiency of Transfer Learning and Data Augmentation in Lemon Leaf Image Classification*, European Journal of Engineering and Applied Sciences, vol. 6, no. 1, pp. 32–40, Jul. 2023, doi: 10.55581/ejeas.1321042.
22. E. K. Yılmaz, K. Adem, S. Kılıçarslan, and H. A. Aydın, *Classification of lemon quality using hybrid model based on Stacked AutoEncoder and convolutional neural network*, European Food Research and Technology, vol. 249, no. 6, pp. 1655–1667, Jun. 2023, doi: 10.1007/s00217-023-04245-6.
23. S. Verma, P. Kumar, and J. P. Singh, *MLNAS: Meta-learning based neural architecture search for automated generation of deep neural networks for plant disease detection tasks*, Network: Computation in Neural Systems, pp. 1–24, Jul. 2024, doi: 10.1080/0954898X.2024.2374852.
24. Y. Xu et al., *Pc-darts: Partial channel connections for memory-efficient architecture search*, arXiv preprint arXiv:1907.05737, 2019.
25. K. Patil, Y. Suryawanshi, A. Patrawala, and P. Chumchu, *A comprehensive lemongrass (Cymbopogon citratus) leaf dataset for agricultural research and disease prevention*, Data in Brief, vol. 53, p. 110104, Apr. 2024, doi: 10.1016/j.dib.2024.110104.
26. E. A. Sultan, S. E. El-Khamy, S. El-Rabie, N. A. El-Fishawy, F. E. A. El-Samie, and Salwa Mohamed Serag Eldin, *New Efficient Interpolation Techniques for Medical Images*, Aug. 08, 2023, doi: 10.21203/rs.3.rs-2869416/v1.
27. C. Shorten and T. M. Khoshgoftaar, *A survey on Image Data Augmentation for Deep Learning*, Journal of Big Data, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
28. E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, *The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study*, Multimedia Tools and Applications, vol. 82, no. 11, pp. 16591–16633, May 2023, doi: 10.1007/s11042-022-13820-0.
29. G. Dai and J. Fan, *An Industrial-Grade Solution for Crop Disease Image Detection Tasks*, Frontiers in Plant Science, vol. 13, Jun. 2022, doi: 10.3389/fpls.2022.921057.
30. P. Ren et al., *A Comprehensive Survey of Neural Architecture Search*, ACM Computing Surveys, vol. 54, no. 4, pp. 1–34, May 2022, doi: 10.1145/3447582.
31. J.-S. Kang, J. Kang, J.-J. Kim, K.-W. Jeon, H.-J. Chung, and B.-H. Park, *Neural Architecture Search Survey: A Computer Vision Perspective*, Sensors, vol. 23, no. 3, p. 1713, Feb. 2023, doi: 10.3390/s23031713.
32. S. Liu, H. Zhang, and Y. Jin, *A survey on computationally efficient neural architecture search*, Journal of Automation and Intelligence, vol. 1, no. 1, p. 100002, Dec. 2022, doi: 10.1016/j.jai.2022.100002.
33. Y. Sun, X. Sun, Y. Fang, G. G. Yen, and Y. Liu, *A Novel Training Protocol for Performance Predictors of Evolutionary Neural Architecture Search Algorithms*, IEEE Transactions on Evolutionary Computation, vol. 25, no. 3, pp. 524–536, Jun. 2021, doi: 10.1109/TEVC.2021.3055076.
34. Z. Cai, L. Chen, and H.-L. Liu, *EPC-DARTS: Efficient partial channel connection for differentiable architecture search*, Neural Networks, vol. 166, pp. 344–353, Sep. 2023, doi: 10.1016/j.neunet.2023.07.029.
35. D. Pang, X. Le, and X. Guan, *RL-DARTS: Differentiable neural architecture search via reinforcement-learning-based meta-optimizer*, Knowledge-Based Systems, vol. 234, p. 107585, Dec. 2021, doi: 10.1016/j.knosys.2021.107585.
36. Z. Li, Y. Xu, P. Ying, H. Chen, R. Sun, and X. Xu, *LMD-DARTS: Low-Memory, Densely Connected, Differentiable Architecture Search*, Electronics, vol. 13, no. 14, p. 2743, Jul. 2024, doi: 10.3390/electronics13142743.
37. M. Laurer, W. van Atteveldt, A. Casas, and K. Welbers, *Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI*, Political Analysis, vol. 32, no. 1, pp. 84–100, Jan. 2024, doi: 10.1017/pan.2023.20.
38. D. Albashish, R. Al-Sayyed, A. Abdullah, M. H. Ryalat, and N. Ahmad Almansour, *Deep CNN Model based on VGG16 for Breast Cancer Classification*, in 2021 International Conference on Information Technology (ICIT), IEEE, Jul. 2021, pp. 805–810. doi: 10.1109/ICIT52682.2021.9491631.
39. C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, *Rethinking the Inception Architecture for Computer Vision*, arXiv preprint arXiv:1512.00567.
40. A. Dhillon and G. K. Verma, *Convolutional neural network: a review of models, methodologies and applications to object detection*, Progress in Artificial Intelligence, vol. 9, no. 2, pp. 85–112, Jun. 2020, doi: 10.1007/s13748-019-00203-0.