Utilizing Transfer Learning for Deep Learning Based Image Classification

Mengying Shu [a,*]

[a]Computer Engineering, lowa State University, Ames, Iowa, 50011, USA; tinamyshu@gmail.com

*Corresponding Author: tinamyshu@gmail.com

DOI: https://doi.org/10.30212/JITI.202503.004

Submitted: Dec. 12, 2024 Accepted: Feb. 16, 2025

ABSTRACT

Since the inception of the ImageNet Challenge in 2010, numerous innovative deep CNNs (D-CNNs) have been developed. These models typically require extensive datasets for training, making their application to smaller datasets infrequent due to overfitting risks. This paper introduces an adapted deep neural network model designed to effectively fit a small dataset. The main contribution of this work lies in demonstrating the effective use of transfer learning and fine-tuning techniques to adapt pre-trained deep learning models, such as VGG16, VGG19, Inception V3, and Inception-ResNetV2, for small-scale image classification tasks. By incorporating data augmentation, dropout regularization, and selective layer freezing, the study achieves high classification accuracy while minimizing overfitting. This approach provides a practical solution for leveraging advanced deep learning architectures in scenarios with limited labeled data.

Keywords: Image Classification, Deep Learning, Computer Vision

1. Overview

Deep learning has profoundly influenced well-structured perception and classification tasks, often requiring vast datasets to effectively train models. The success of DNNs is largely attributed to the large and diverse training datasets. Over recent years, leading models have surpassed human abilities in object recognition, showcasing deep learning's maturity and widespread application. However, achieving high performance with small datasets remains challenging. Previous research has primarily focused on developing deep learning models for large-scale datasets, such as ImageNet, with less emphasis on addressing the unique challenges posed by small datasets. Existing studies often highlight the risks of overfitting and poor generalization in small data scenarios but provide limited practical solutions for effectively adapting advanced deep learning architectures to such contexts. For instance, while transfer learning has been explored, many approaches lack detailed insights into optimizing pre-trained models for small datasets or mitigating the challenges of domain-specific features and limited sample diversity.

This research aims to address these gaps by focusing on the effective application of transfer learning and fine-tuning techniques to small datasets. By leveraging pre-trained deep learning models, such as VGG16, VGG19, Inception V3, and Inception-ResNetV2, this study investigates how to adapt these architectures for small-scale image classification tasks. The study emphasizes strategies such as selective layer freezing, data augmentation, and dropout regularization to mitigate overfitting while maximizing model generalization. The focus is to provide a practical framework for utilizing advanced deep learning models in small data contexts, which are increasingly common in real-world applications where labeled data is limited.

1.1 Introduction

CNNs have become pivotal model in deep learning, primarily used for image analysis. Originating in the 1980s [1], CNNs gained momentum with the advent of GPUs [2] in the early 2000s. Dan, 2021 introduced CNN trained on GPUs, achieving notable test error rates on datasets like MNIST, NORB, and CIFAR10 [3]. In 2012, Krizhevsky's model triumphed in the ImageNet Challenge [4, 5], utilizing 1.2 million high-resolution images and an architecture featuring eight weight layers, including 5 Conv and 3 FC layers, alongside the ReLU nonlinearity [6]. Following AlexNet's success, ZFNet was introduced in 2013, enhancing AlexNet's hyperparameters for improved performance [7]. GoogleNet, the 2014 ImageNet Challenge winner, featured a 22-layer network, significantly reducing parameters from AlexNet's 60 million to 4 million. ResNet brings a skip connections, allowing input passage between layers without modification, leading to a 152-layer network that won the ImageNet Challenge [8]. These architectures highlight the importance of depth in achieving superior CNN performance [9].

While GoogleNet, VGG, and ResNet can be trained on large ImageNet datasets, such massive datasets are often unavailable. Acquiring large datasets can be costly for specific tasks [10]. For smaller datasets, employing technology that works with deep networks is efficient, achieving high performance through transfer learning. Transfer learning involves applying pre-trained model knowledge to downstream tasks without starting from the beginning. Transfer learning is a method that transfers knowledge from one completed task to a new one. Fine-tuning, as a subset of transfer learning, typically focuses on retraining certain layers of a pre-trained model to adapt to the characteristics of the new dataset. In this work, transfer learning is implemented by leveraging deep learning models pre-trained on the large-scale ImageNet dataset, such as VGG16, VGG19, Inception V3, and Inception-ResNet V2. Fine-tuning involves unfreezing and retraining specific layers in these models to better capture features of the smaller target dataset. CNNs like ResNet, GoogleNet, and VGG, can be pre-trained on large ImageNet classification, with later layers capturing more complex structures. The final layer classifies images into categories. Thus, transfer learning can be applied to small datasets, speeding up optimization by retaining relevant lower-layer features and mitigating overfitting through data augmentation & dropout techniques.

2. Introduction to CNN and ML

This section provides an introduction to machine learning, transfer learning, and various CNN models.

2.1 Background of ML

In recent years, ML, DL, and AI have been extensively covered in numerous articles. Originating in the 1950s, artificial intelligence emerged from the computer science, exploring whether digital circuits could emulate human thinking. AI encompasses ML and DL, among other disciplines. ML focuses on whether computers could autonomously solve tasks. In classical programming, rules and data yield answers, while in machine learning, data and answers generate rules, which are then applied to downstream data for new tasks. Essentially, a ML system learns by being exposed to task-related examples to derive rules that conduct automation. Deep learning, a machine learning subset, uses consecutive layers to represent increasingly complex features, driven by experimental findings rather than theory.

2.2 CNNs

CNN models comprise convolutional, pooling, ReLU, fully connected, and final loss calculation layers. These components work together to extract hierarchical features from input data. The convolutional layer applies filters (kernels) to input images to detect patterns like edges, textures, and more complex structures as layers deepen. Pooling layers, such as max pooling or average pooling, reduce the spatial dimensions of feature maps, preserving the most critical information while minimizing computational load. The ReLU (Rectified Linear Unit) activation function introduces non-linearity, allowing the model to capture complex feature relationships across layers. Together, these components enable CNNs to progressively extract and combine low-level features (e.g., edges) into high-level abstractions (e.g., shapes or objects), forming the basis for effective feature learning. The fully connected layer connects all previous layer activations, flattening the output to represent high-level data features. The loss layer quantifies the difference the prediction and ground-truth target.

2.3 CNN with Deeper Architectures

VGGNet [11] secured second place in the ImageNet Challenge, following GoogLeNet and leading in image localization. VGG19 is slightly superior to VGG16 but demands more memory. VGG16 comprises convolutional & pooling & FC layers, totaling 16 layers in five blocks, while each block has a max pooling layer. VGG19, similar to VGG16, includes 19 layers, adding another CONV layers in the extra blocks. These deep CNNs enable VGG19 & VGG16 to excel in vision tasks.

GoogLeNet, the 2014 ILSVRC winner, features a 22-layer network with an inception module [12]. It underscores the significance of depth while using 12 times fewer parameters than AlexNet, achieving near-human performance. GoogLeNet's key innovation is the parallel combination of 1x1, 3x3, and 5x5 convolutional filters and 1x1 convolutional kernels to reduce computation. Subsequent research led to Inception V2, which normalizes feature map responses by their mean and standard deviation. Inception V3 further refines performance by employing 3x3 and 1x1 filters instead of 5x5 and 7x7 filters, showcasing the power of deeper models.

Inspired by ResNet's performance [8], Google developed InceptionResNet, incorporating residual connections into the inception module's convolutional output [13]. This model, with approximately 467 layers, accelerates training and enhances accuracy, illustrating the strength of deep layers. While these models excel on large datasets like ImageNet, their architectural features also make them particularly effective for small datasets when combined with transfer learning techniques[14]. For instance, ResNet's use of skip connections mitigates the vanishing gradient problem in deep networks, allowing efficient training even with limited data. Similarly, Inception-based architectures optimize computational efficiency through their inception modules, which capture multi-scale features by combining convolutions of varying kernel sizes. These characteristics make models like ResNet and Inception well-suited for small datasets, as they can extract complex features with fewer parameters and reduced overfitting risk when appropriately fine-tuned.

3. Related Work

The methodology outlined in this paper intersects with multiple fields, including image categorization, knowledge transfer, and advanced neural network techniques, which are discussed briefly below.

3.1 Vision Classification

Choosing an appropriate method for categorizing images and efficiently leveraging diverse data characteristics [15] is critical for tackling image categorization problems. Similar to this approach, utilizes neural-network-driven feature extraction for tasks like facial recognition and detecting flaws in wooden materials, incorporating grouping of basic-level features. In contrast to [16], my methodology involves leveraging pre-trained models on substantial datasets and fine-tuning them for use on smaller-scale datasets, yielding significant performance advancements.

3.2 Transfer Learning

Developing a model entirely from scratch is uncommon due to the lack of extensive datasets. Instead, researchers often adopt the practice of pre-training models on large datasets such as ImageNet and subsequently extracting features for other applications. Knowledge transfer is introduced as a strategy to improve efficiency and conserve time. However, transfer learning is not without limitations. When the source and target domains differ significantly, negative transfer can occur, where knowledge from the source task adversely impacts performance on the target task. To mitigate this, careful selection of source models and task-specific fine-tuning strategies are essential. As demonstrated in [17], transferring learned knowledge enhances outcomes while minimizing the need for new data collection. The study in [17] addresses key questions: "what knowledge should be transferred," "how to achieve effective transfer," and "when transfer is appropriate." "What to transfer" involves determining specific elements to transfer across domains, aligning with the considerations for "how to transfer." "When to transfer" evaluates the suitable circumstances to apply knowledge transfer. These principles guide my approach to incorporating transfer learning into my work. Moreover, [18] discusses the obstacles linked to transfer learning. As described in [18], the goal is to improve competencies in the target task by leveraging insights from a source domain. A critical challenge lies in fostering a positive transfer while avoiding undesirable transfer, which may occur when the tasks are loosely related. Therefore, it is important to identify and exclude irrelevant information during the training process, ensuring transfer learning performs comparably or better than its absence. [18] provides a framework for addressing these challenges and avoiding negative transfer.

Transfer learning supports the learning process in a new task by utilizing knowledge derived from a previously solved, related problem. In computer vision, case studies like [19] and [20] highlight significantly enhanced classification accuracy for objects[21]. Previous research has demonstrated that models such as ResNet and Inception consistently outperform shallower architectures on small datasets due to their unique design choices. For example, ResNet's residual connections enable the network to maintain feature quality across its depth[22], a critical advantage when adapting pre-trained models to small datasets. Meanwhile, Inception's multi-branch design efficiently handles feature extraction at multiple scales, making it versatile for datasets with limited samples but diverse feature representations[23-25]. These architectural strengths underscore the rationale for selecting these models in this study, as they effectively balance model complexity and generalization performance.

In alignment with my research, studies by [26], investigate the application of deep neural networks for image categorization tasks using large-scale data sources. Our approach adapts pre-trained models for use with smaller datasets, mitigating the risk of significant overfitting. Transfer learning has been applied in medical imaging tasks, similar to the methods employed in my work. Other investigations suggest utilizing representations extracted from large-scale image datasets with CNN architectures, employing regularization techniques like dropout and fine-tuning [27]. However, these studies predominantly focus on transfer learning for individual identification in extensive datasets. In this report, I specifically focus on transfer learning techniques tailored for smaller datasets.

3.3 Deep Learning

The growing focus on neural networks with multiple layers has led to significant advancements in deep learning. Researchers have explored both guided approaches and unguided methodologies [28, 29].

Advanced neural networks facilitate the development of models capable of extracting intricate data features [30]. Notable architectures, such as AlexNet, VGGNet, GoogleNet[31], ResNet, ResNeXt [32], RCNN (Region-Based CNN) [33], and YOLO (You Only Look Once) [34], showcase diverse applications of deep learning. This paper emphasizes the use of VGG, GoogleNet, and ResNet.

While these models demonstrate strong capabilities, their performance with limited datasets remains an area of exploration. This challenge is especially apparent when adapting these architectures to constrained environments. The VGG-16 model, a highly structured neural network, has been customized for tasks like classifying images in compact dataset[35]. This work aims to highlight how deep convolutional models can effectively handle small datasets, including CIFAR-10, through tailored and practical changes. For example, [34] applies Batch Normalization and robust dropout techniques to improve network performance. Our approach includes enhanced preprocessing steps, such as data augmentation, to improve model robustness.

4. Methods

A deep convolutional network (CNN) is employed to analyze a small-scale Kaggle dataset. The following sections outline the model's structure and implementation details. This study involves classifying images of dogs and cats from a dataset of 6,000 samples, distributed as follows: 3,000 for training, 2,000 for validation, and 1,000 for testing. The Kaggle dog vs. cat dataset was chosen for its representativeness of small, real-world datasets, which are commonly encountered in practical applications. This dataset has several characteristics that align with the paper's central arguments about the challenges of small datasets: it contains a limited number of labeled images, exhibits significant variability in image quality (e.g., blurry or poorly scaled images), and includes instances of occlusion or cluttered backgrounds. These properties make it an ideal candidate for testing the robustness of transfer learning techniques in small data scenarios. Moreover, its binary classification task allows for focused evaluation of how well pre-trained models generalize to new domains with minimal data. Despite using less than 20% of the dataset for training, the model achieves a commendable accuracy of 96%.

4.1 Model Structure

The models selected for this study were originally trained on ImageNet, a comprehensive dataset containing various classes of animals and objects. These pre-trained models are adapted to suit image classification tasks in smaller datasets. Deep architectures, such as VGG16, VGG19, Inception V3, and Inception-ResNetV2, were specifically chosen for their relevance to compact-scale problems.

4.2 DNNs

Initially, a baseline model was developed to evaluate performance on the smaller dataset. This serves as a reference point for comparing pre-trained networks. For deep architectures like VGG16, VGG19, Inception V3, and Inception-ResNetV2, weights trained on ImageNet were imported and utilized. Layer configurations and final features were analyzed using model summaries. To adapt the pre-trained models, lower convolutional layers were frozen to preserve the general low-level features learned from the ImageNet dataset, such as edges and textures. A new dense classifier was added on top of the feature maps, along with dropout layers for regularization. Additionally, fine-tuning was applied to the final convolutional blocks (e.g., block5 of VGG16 and the mixed9 layer of Inception V3) to optimize the models for the target dataset.

A binary classifier was subsequently integrated into these models for classification tasks. The training and validation accuracies were documented to assess performance.

To further enhance results, feature extraction techniques were applied to each model. Recognizing the risk of negative transfer, lower convolutional layers of the pre-trained models were frozen to retain generalized features, while higher layers were fine-tuned to adapt to the specific characteristics of the target dataset. The preprocessing pipeline included data augmentation strategies, such as flipping and cropping, to expand the dataset size artificially and improve generalization. This approach addresses the limitations posed by input size constraints in small datasets.

5. Results

The training process was performed on a small dataset consisting of 3,000 images for training, 2,000 for validation, and 1,000 for testing, amounting to 6,000 total images. This dataset size presents challenges for a deep convolutional neural network (CNN).

5.1 Training Results

Experiments were conducted on Google Colab using a virtual GPU. Initially, the data was retrieved from Kaggle, and folders were created to separate training, validation, and testing sets. Within each folder, the images of cats and dogs were further organized. The dataset was uploaded to Google Drive for execution. As the images were in JPEG format, preprocessing involved converting them into RGB pixel matrices using the Keras ImageDataGenerator utility in TensorFlow. Five models were evaluated: a baseline model, VGG16, VGG19, Inception V3, and Inception-ResNetV2. For the latter models, techniques like data augmentation, feature extraction, and fine-tuning were incorporated. The performance metrics used were accuracy and loss, enabling an assessment of model effectiveness and overfitting.

The baseline model, comprising a 5-layer CNN, was trained without using dropout or data augmentation. This simple setup served as a benchmark to compare against deeper models. As shown in Figure 5.1, the baseline model delivered only 73% accuracy on test data, with significant overfitting evident from the gap between training and validation performance metrics.

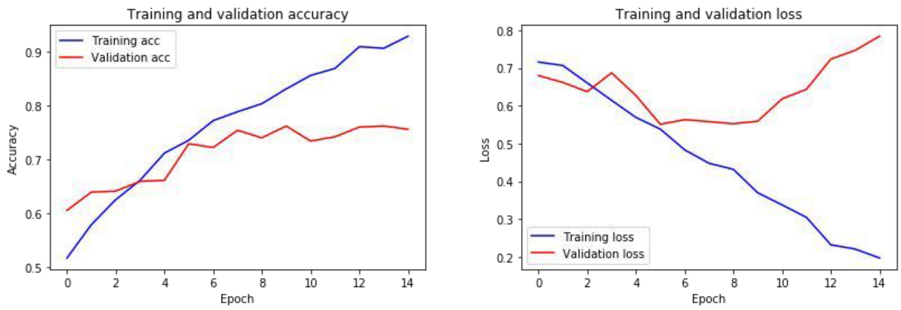

Figure 1. Baselines' Training Results.

I initialized pre-trained weights for the deeper models and added a dense classifier with dropout at the top of the final feature maps. The feature map dimensions were (4, 4, 512) for VGG16 and VGG19, and (3, 3, 2048) or (3, 3, 1536) for Inception V3 and Inception-ResNetV2, respectively. These feature maps served as input to the applied layers for training. When running these models on the smaller dataset, all outperformed the baseline, though significant overfitting persisted, emphasizing the need for data augmentation.

While using feature extraction, overfitting remained a challenge, even as accuracy improved. To mitigate this, data augmentation techniques were applied, such as random flips and rotations. In addition to accuracy, precision, recall, and F1-score were analyzed to better understand the model's behavior. For example, while InceptionResNetV2 achieved an F1-score of 95.4% on the test set, its recall of 96.3% suggests a strong ability to identify true positives, though its precision of 94.5%

indicates occasional misclassification of false positives. Misclassification analysis revealed common issues, such as images where dogs or cats were partially occluded or featured in non-standard poses. These insights underscore the challenges of small datasets and the importance of advanced techniques like fine-tuning and dropout for improving model robustness.

Additionally, the convolutional layers of the pretrained models were frozen to maintain previously learned weights. Each model ran for approximately 20 minutes on a virtual GPU in Google Colab. This approach improved both accuracy and generalization. For VGG16 and VGG19, validation accuracy reached 90%, though some degree of overfitting persisted. Inception V3 and Inception-ResNetV2 achieved validation accuracy around 95%, though with a slight increase in loss, indicating room for further optimization.

To enhance the performance further, I fine-tuned the models. For VGG16 and VGG19, the final convolutional and pooling layers were unfrozen, enabling the adjustment of feature extraction weights. Similarly, dropout and fully connected layers were fine-tuned. For Inception V3, the mixed9 layer and subsequent layers were optimized, while for Inception-ResNetV2, block 10 mixed was selected for fine-tuning. The training time averaged around 15 minutes per model on Google Colab. After fine-tuning, the models displayed significant improvements, and overfitting was further reduced. For VGG16 and VGG19, validation accuracy improved, and loss decreased, though some variations in loss indicate potential for future refinement. Inception V3 and Inception-ResNetV2 achieved the highest accuracy levels, though validation loss was slightly higher, suggesting a need for additional adjustments in later research..

5.2 Results

Figures 1 through 13 illustrate the experimental results. When applying feature extraction with pretrained deep models to this small dataset, these models achieved significantly better accuracy than the baseline while requiring less computational effort. This indicates that deep architectures are advantageous for tasks requiring precision under limited resource conditions.

By employing data augmentation and dropout with deep networks, these models achieved superior performance. In my analysis, the depth of these architectures was key to achieving high accuracy, while techniques like augmentation and dropout successfully minimized overfitting. This demonstrates that even small datasets can benefit from the strengths of advanced deep learning architectures.

5.3 Training on baseline

As shown in Figure 1, the baseline model is straightforward to train, delivering results relatively quickly. However, its performance, achieving only 75% accuracy, significantly lags behind the deeper models' performance, highlighting the limitations of shallow architectures in complex tasks.

5.4 VGG16 and VGG19 and VGG16 Training Results

Figures 2 and 3 demonstrate noticeable performance improvements relative to the baseline. Despite this, overfitting persists, as training accuracy exceeds validation accuracy, and training loss is lower than validation loss. The test accuracy for both VGG16 and VGG19 reaches 83%.

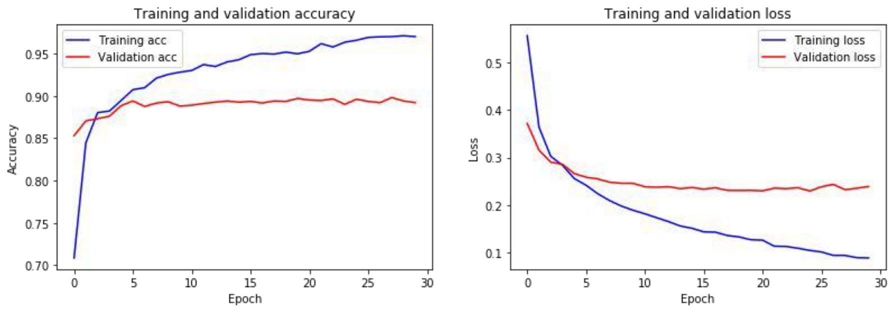

Figure 2. VGG16’s Training Result

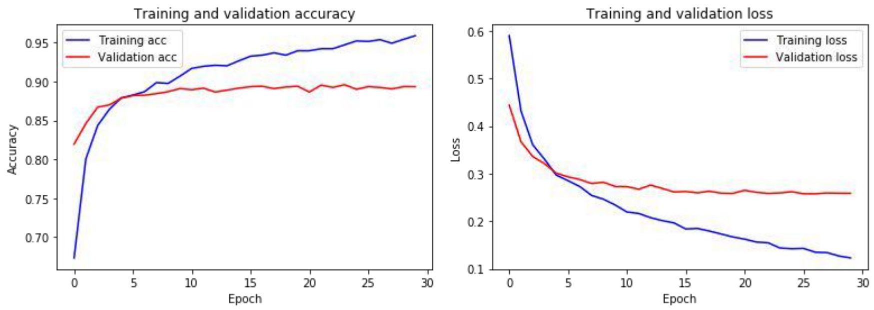

Figure 3. VGG19’s Training Result.

5.5 With data augmentation, VGG16 and VGG19 and VGG16 Training Results

After applying data augmentation techniques, test accuracies improve to approximately 85.2% for VGG19 and 88.3% for VGG16. This approach significantly reduces overfitting, as highlighted in Figures 4 and 5. While training accuracy is slightly less than validation accuracy, training loss marginally exceeds validation loss, suggesting that data augmentation minimizes overfitting. Additional fine-tuning, however, may further enhance accuracy levels.

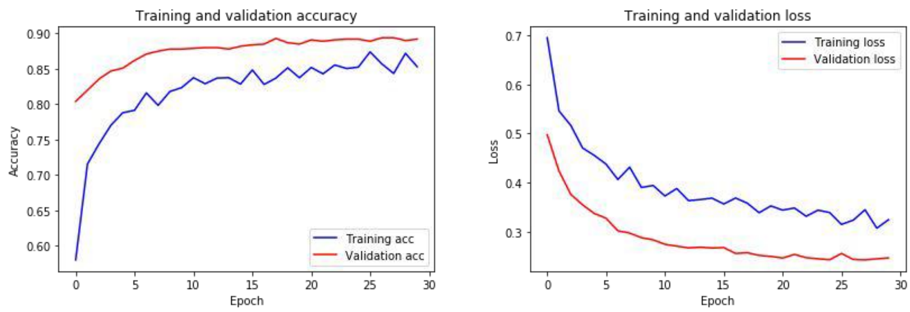

Figure 4. With Data Augmentation, VGG16’s Training.

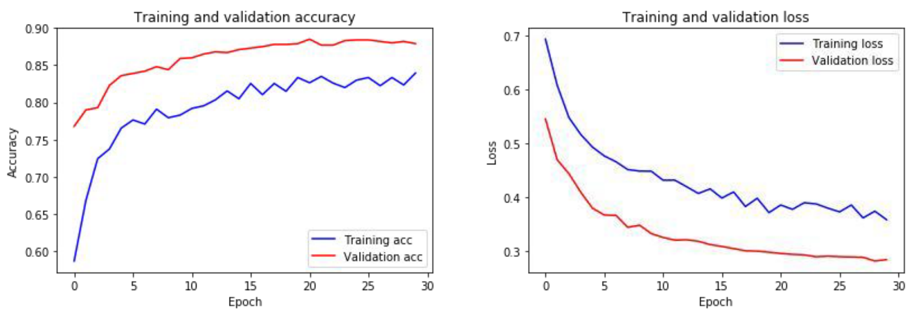

Figure 5. With Data Augmentation, VGG19's Training.

5.6 With fine-tuning, Training on VGG19 and VGG16

Fine-tuning the models leads to test accuracies of about 88.1% for VGG16 and 85.9% for VGG19. Figures 4 and 5 reveal that data augmentation coupled with fine-tuning helps mitigate overfitting. Training accuracy remains lower than validation accuracy, and training loss is slightly greater than validation loss, showing further reduction in overfitting. Fine-tuning is critical to achieving more precise results.

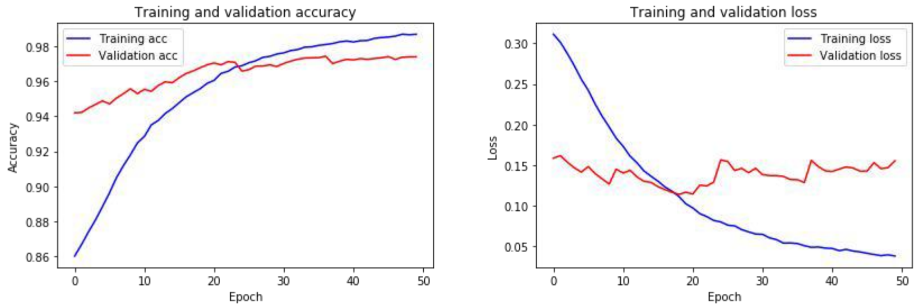

Figure 6. VGG16 Fine-tuning Result.

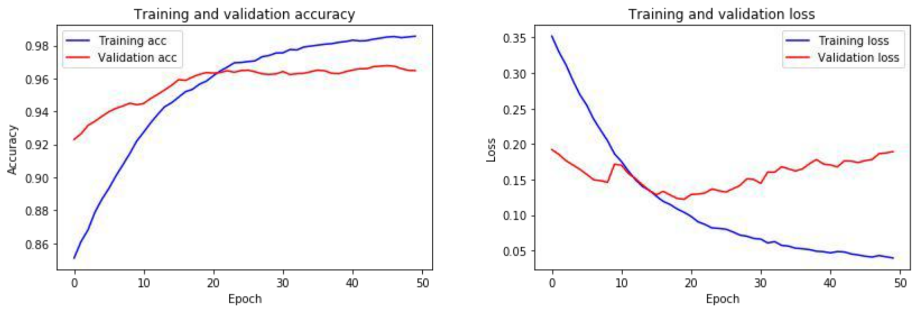

Figure 7. VGG19 Fine-tuning Result.

5.7 InceptionResNet V2 Inception V3 Training Results

Figures 8 and 11 illustrate that training accuracy is considerably higher than validation accuracy, while training loss is much lower than validation loss. Overfitting remains an issue, yet Inception V3 and InceptionResNetV2 achieve test accuracies of 93% and 95%, respectively, showcasing their superior performance compared to the baseline.

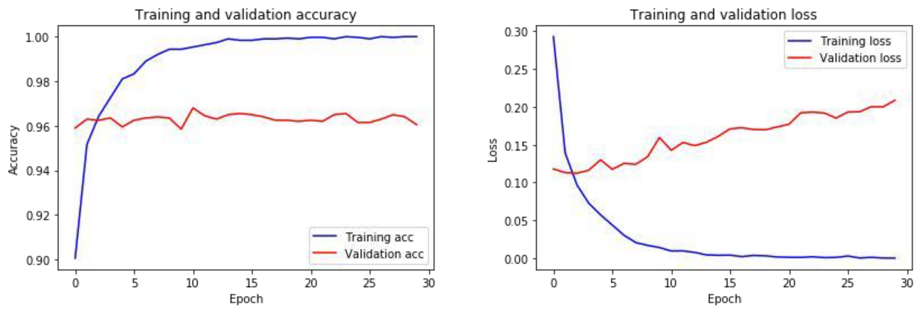

Figure 8. Inception V3 Training Result.

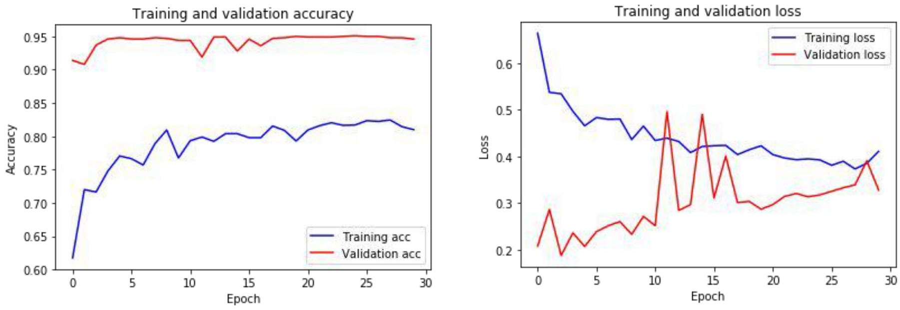

Figure 9. With Data Augmentation,Inception V3's Training Result.

5.8. With Data Augmentation, Inception V3 and InceptionResNet V2 Training Results

Figures 9 and 12 indicate that training accuracy falls short of validation accuracy, and training loss is greater than validation loss. Data augmentation, similar to its application in VGG16 and VGG19, enhances model performance. Validation accuracy surpasses training accuracy, yet the loss values remain elevated. Improvement strategies are required to further optimize performance. Inception V3 and InceptionResNetV2 achieve test accuracies of 94.7% and 95.6%, respectively.

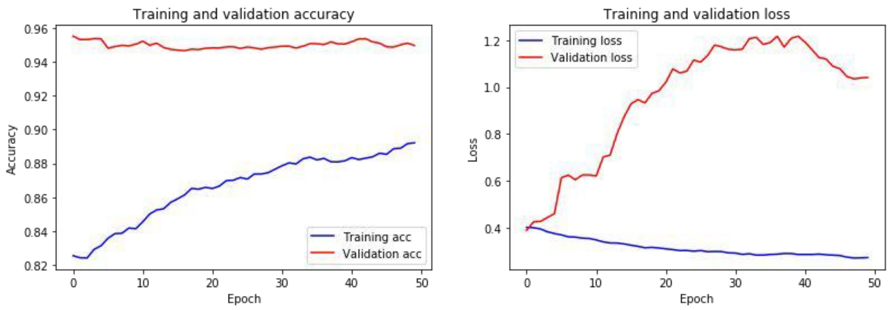

Figure 10. Inception V3 Fine-tuning Result.

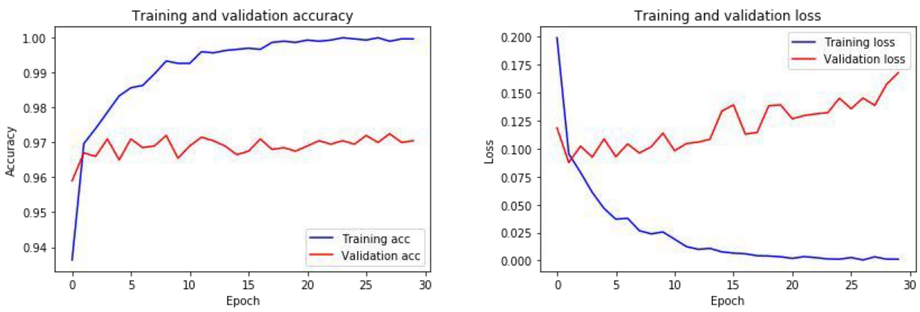

Figure 11. InceptionResNet V2 Training Result.

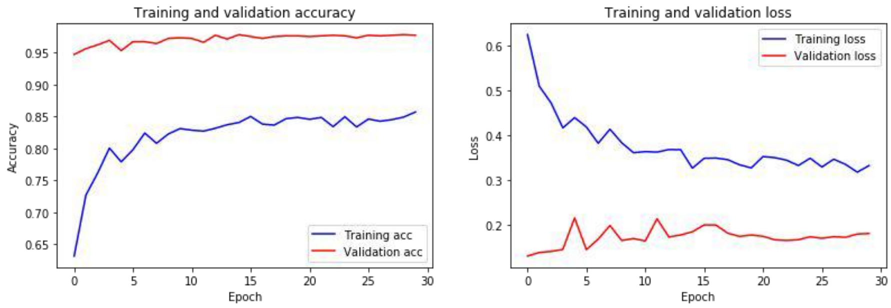

Figure 12. Training on InceptionResNet V2 with Data Augmentation.

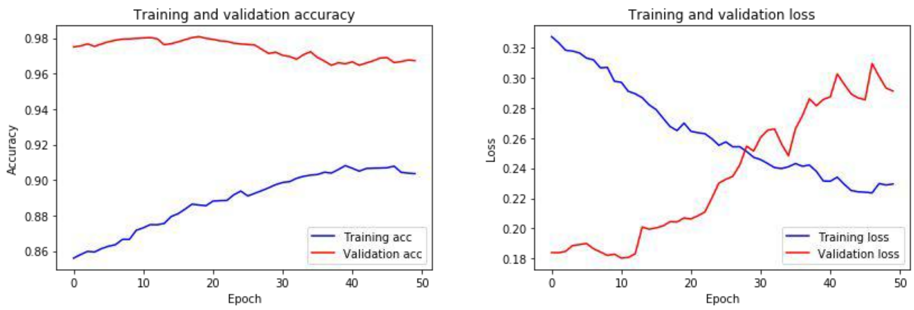

Figure 13. InceptionResNet V2 Fine-tuning Result.

5.9 With Fine-tuning, InceptionResNet V2 and Inception V3's Training Results

Figures 10 and 13 demonstrate that training accuracy falls below validation accuracy. In Figure 10, training loss is notably lower than validation loss, yet overfitting persists. In Figure 13, while the training loss initially exceeds the validation loss, it eventually drops below it. Fine-tuning improves outcomes significantly, with InceptionResNetV2 achieving 96% test accuracy and Inception V3 reaching 95%. Nonetheless, validation loss increases slightly, warranting additional investigation in subsequent research.

5.10 Conclusion

The results presented in the figures summarize the accuracy and loss observed during training and validation phases. A noticeable gap between training and validation accuracy often signals overfitting, while the reverse suggests underfitting. Comparing the loss values provides insight into model performance. When training loss surpasses validation loss, underfitting is evident, whereas lower training loss indicates overfitting. Figures 1, 2, 3, 8, and 11 reveal significant overfitting due to the dataset's limited size. By contrast, Figures 4 and 5 illustrate notable reductions in overfitting when enhancements are applied. Figures 9 and 12 reveal minimal underfitting, with both models performing well overall. Fine-tuning improves performance further, as observed in Figures 6 and 7, though slight underfitting persists in Figures 10 and 13. Addressing excessive weight updates remains a priority for future work.

5.11. Analysis

Figure 14. Test Image Analysis.

The highest test accuracy achieved is 96%, with approximately 40 images incorrectly labeled. Figure 14 highlights challenges such as blurry and scaled-up images, animals facing away or blending with their surroundings, closed eyes, and obstructive fences. Additionally, the presence of humans in some images complicates identification. Overall, noise in the images makes distinguishing cats and dogs difficult.

6. Conclusion

This study demonstrates that advanced deep learning models such as VGG16, VGG19, Inception V3, and InceptionResNet V2 can be effectively utilized on small datasets consisting of 6,000 images—3,000 for training, 2,000 for validation, and 1,000 for testing—while minimizing overfitting. The research focuses on applying transfer learning and fine-tuning techniques to adapt pre-trained models for small-scale image classification tasks. Strategies such as selective layer freezing, data augmentation, and dropout regularization were employed to optimize model performance and address challenges like overfitting.

In the future, this framework could be extended by exploring additional pre-trained models or applying the proposed techniques to more diverse datasets and multi-class classification tasks. Furthermore, incorporating advanced optimization strategies could improve model generalization and scalability in real-world applications, such as medical imaging and remote sensing. Transfer learning, in particular, enables effective use of pre-trained CNNs for tasks with limited data by reusing generalized features from large-scale datasets. This capability is especially evident in tasks such as small-scale object classification and medical image analysis, where the availability of labeled data is constrained. By adapting pre-trained models, this study demonstrates that even small datasets can achieve high accuracy with carefully designed transfer learning strategies.

The use of transfer learning allowed us to leverage pre-trained models on large-scale datasets, which provided robust low-level feature representations. Fine-tuning specific layers in the pre-trained models further improved the adaptability and accuracy on the small target dataset, demonstrating the effectiveness of combining both strategies. As shown by the results, deep learning models can adapt well to small-scale datasets with careful selection and appropriate modifications. Essential techniques include augmenting the dataset with transformations, employing dropout layers, and performing fine-tuning. Choosing suitable deep architectures for smaller datasets ensures better accuracy and performance. Although overfitting was substantially mitigated during the experiments, certain models faced underfitting issues, which were possibly linked to excessive dropout settings. Overall, the study confirms that deep neural networks can be customized effectively for small datasets by employing appropriate adjustments.

6.1 Future work

Future investigations could involve testing additional models and exploring various configuration adjustments (see Table 1). For example, analyzing the causes behind increased validation loss during fine-tuning and identifying factors contributing to underfitting. Moreover, examining how models such as YOLO (You Only Look Once) and RCNN (Region-Based CNN) perform on smaller datasets would be a valuable direction. Further optimizing models to enhance compatibility with constrained datasets and performing comparative analyses could yield significant insights for future research.

Table1. Results Collection

	Training Accuracy	Validation accuracy	Testing accuracy
Baseline	86%	75%	73.8%
VGG16	95%	89%	83%
VGG16 with Data Augmenation	83%	89%	88.1%
VGG16 with FineTuning	97%	96%	95%
VGG19	95%	89%	83%
VGG19 with Data Augmenation	83%	88%	85.2%
VGG19 with FineTuning	98%	96%	95.3%
Inception V3	99%	96%	93%
Inception V3 with Data Augmenation	80%	95%	94.7%
Inception V3 with FineTuning	88%	95%	95%
InceptionResNet V2	99%	97%	95%
InceptionResNet V2 with Data Augmenation	82%	95%	95.6%
InceptionResNet V2 with FineTuning	90%	97%	96%

References

[1] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 1980, 36(4), 193-202.

[2] McClanahan, C. History and evolution of gpu architecture. A Survey Paper, 2010, 9, 1-7.

[3] Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M. and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification.

[4] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 248-255.

[5] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115, 211-252.

[6] Krizhevsky, A., Sutskever, I. and Hinton, G.E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012, 25.

[7] Zeiler, M.D. Visualizing and Understanding Convolutional Networks.

[8] Shafiq, M. and Gu, Z. Deep residual learning for image recognition: A survey. Applied Sciences, 2022, 12(18), 8972.

[9] Luo, M., Du, B., Zhang, W., Song, T., Li, K., Zhu, H., Birkin, M. and Wen, H. Fleet rebalancing for expanding shared e-Mobility systems: A multi-agent deep reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(4), 3868-3881.

[10] Pang, H., Zhou, L., Dong, Y., Chen, P., Gu, D., Lyu, T. and Zhang, H. Electronic Health Records-Based Data-Driven Diabetes Knowledge Unveiling and Risk Prognosis, 2024

[11] Bhatti, U.A., Tang, H., Wu, G., Marjan, S. and Hussain, A. Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. International Journal of Intelligent Systems, 2023, 2023(1), 8342104.

[12] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G. and Jégou, H. Going deeper with image transformers. 32–42.

[13] Neshat, M., Ahmed, M., Askari, H., Thilakaratne, M. and Mirjalili, S. Hybrid Inception Architecture with Residual Connection: Fine-tuned Inception-ResNet Deep Learning Model for Lung Inflammation Diagnosis from Chest Radiographs. Procedia Computer Science, 2024, 235, 1841–1850.

[14] Lyu, T., Gu, D., Chen, P., Jiang, Y., Zhang, Z., Pang, H., Zhou, L. and Dong, Y. Optimized CNNs for Rapid 3D Point Cloud Object Recognition, 2024.

[15] Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D. and Van De Weijer, J. Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(5), 5513–5533.

[16] Sarkar, S., Ejaz, N., Maiti, J. and Pramanik, A. An integrated approach using growing self-organizing map-based genetic K-means clustering and tolerance rough set in occupational risk analysis. Neural Computing and Applications, 2022, 34(12), 9661–9687.

[17] Ribani, R. and Marengoni, M. A survey of transfer learning for convolutional neural networks. 47–57.

[18] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. A comprehensive survey on transfer learning. Proceedings of the IEEE, 2020, 109(1), 43–76.

[19] Oquab, M., Bottou, L., Laptev, I. and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. 1717–1724.

[20] Wang, S., Chen, X., Wang, Y., Long, M. and Wang, J. Progressive adversarial networks for fine-grained domain adaptation. 9213–9222.

[21] Taherkhani, A., Cosma, G. and McGinnity, T.M. AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. Neurocomputing, 2020, 404, 351–366.

[22] Shirke, V., Walika, R. and Tambade, L. Drop: a simple way to prevent neural network by overfitting. International Journal of Research in Engineering Science and Management, 2018, 1, 2581–5782.

[23] Wang, J. and Perez, L. The effectiveness of data augmentation in image classification using deep learning. Convolutional Neural Networks for Visual Recognition, 2017, 11(2017), 1–8.

[24] Wei, X.-S., Song, Y.-Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J. and Belongie, S. Fine-grained image analysis with deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(12), 8927–8948.

[25] Wong, S.C., Gatt, A., Stamatescu, V. and McDonnell, M.D. Understanding data augmentation for classification: when to warp?. 1–6.

[26] Huang, Z., Zhu, X., Ding, M. and Zhang, X. Medical image classification using a light-weighted hybrid neural network based on PCANet and DenseNet. IEEE Access, 2020, 8, 24697–24712.

[27] Chen, H., Wang, Y., Shi, Y., Yan, K., Geng, M., Tian, Y. and Xiang, T. Deep transfer learning for person re-identification. 1–5.

[28] Rolls, E.T. Learning invariant object and spatial view representations in the brain using slow unsupervised learning. Frontiers in Computational Neuroscience, 2021, 15, 686239.

[29] Shahabi, H., Rahimzad, M., Tavakkoli Piralilou, S., Ghorbanzadeh, O., Homayouni, S., Blaschke, T., Lim, S. and Ghamisi, P. Unsupervised deep learning for landslide detection from multispectral Sentinel-2 imagery. Remote Sensing, 2021, 13(22), 4698.

[30] Mathew, A., Amudha, P. and Sivakumari, S. Deep learning techniques: an overview. In: Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020, 2021, 599–608.

[31] McNeely-White, D., Beveridge, J.R. and Draper, B.A. Inception and ResNet features are (almost) equivalent. Cognitive Systems Research, 2020, 59, 312–318.

[32] Zhang, G., Yang, Z., Gong, L., Jiang, S., Wang, L. and Zhang, H. Classification of lung nodules based on CT images using squeeze-and-excitation network and aggregated residual transformations. La Radiologia Medica, 2020, 125, 374–383.

[33] Xie, X., Cheng, G., Wang, J., Yao, X. and Han, J. Oriented R-CNN for object detection. 3520–3529.

[34] Han, X., Chang, J. and Wang, K. You only look once: unified, real-time object detection. Procedia Computer Science, 2021, 183(1), 61–72.

[35] Xi, X., Zhang, C., Jia, W. and Jiang, R. Enhancing human pose estimation in sports training: Integrating spatiotemporal transformer for improved accuracy and real-time performance. Alexandria Engineering Journal, 2024, 109, 144–156.