Comparative Analysis of Pneumonia Classification

*

1st Gurkanwal Singh
B20154

***Abstract*—Pneumonia classification is a critical task in medical image analysis, aiding in the accurate diagnosis and treatment of this severe respiratory infection. This report presents a comparative analysis of two approaches for pneumonia classification: building a CNN model from scratch and utilizing transfer learning with the VGG19 architecture. The objective is to evaluate and compare the performance of these methodologies in accurately identifying pneumonia cases from chest X-ray images. The first approach involves constructing a CNN model from scratch, allowing for a tailored network architecture specifically designed for pneumonia classification. The second approach leverages transfer learning, utilizing the pre-trained VGG19 model as a feature extractor. Key performance metrics such as accuracy, precision, recall, and F1 score are analyzed, along with factors like model complexity and training time. The findings contribute to the field of pneumonia classification, providing insights into the strengths and limitations of each approach and guiding the selection of the most appropriate methodology for pneumonia detection in clinical settings.**

I. Introduction

Pneumonia, a severe respiratory infection affecting the lungs, remains a leading cause of morbidity and mortality worldwide. Timely and accurate diagnosis of pneumonia plays a crucial role in initiating appropriate treatments and improving patient outcomes. With advancements in machine learning, specifically convolutional neural networks (CNNs), the field of medical image analysis has witnessed significant progress in automating pneumonia classification tasks.

This report presents a comparative analysis of two approaches for pneumonia classification: building a CNN model from scratch and employing transfer learning with the VGG19 architecture. The objective of this study is to evaluate and compare the performance of these two methodologies in accurately identifying pneumonia cases from chest X-ray images.

The first approach involves constructing a CNN model from scratch, where the network architecture is designed and trained specifically for pneumonia classification. This method requires careful consideration of the network's depth, convolutional and pooling layers, activation functions, and optimization algorithms. Training a CNN from scratch allows for the exploration of specific features and patterns related to pneumonia, potentially leading to a highly tailored model.

The second approach leverages transfer learning, a technique that utilizes pre-trained models on large-scale image datasets to address smaller, domain-specific tasks. In this case, the widely-used VGG19 architecture, pre-trained on the ImageNet dataset, is employed as a feature extractor. The learned representations from VGG19 are then fine-tuned and combined with additional layers to classify pneumonia cases. Transfer learning offers several advantages, such as faster training times, reduced risk of overfitting, and the ability to benefit from knowledge obtained from a vast array of non-medical images.

This study aims to provide a comprehensive evaluation of the strengths and limitations of both approaches by analyzing key performance metrics. Additionally, factors like model complexity, training time, and computational requirements will be considered to assess the feasibility and practicality of each approach in real-world settings.

In conclusion, this report aims to contribute to the growing body of knowledge on pneumonia classification by comparing the efficacy of building a CNN model from scratch and using transfer learning with the VGG19 architecture. The findings of this study have the potential to guide healthcare professionals, researchers, and developers in selecting the most appropriate methodology for pneumonia detection and assist in the development of more accurate and efficient computer-aided diagnostic systems.

II. Proposed Method

In this report, we propose a methodology for pneumonia classification utilizing Convolutional Neural Network (CNN) models. The objective is to develop an accurate and efficient system for identifying pneumonia cases from chest X-ray images. The proposed method involves the following key steps: data preprocessing, model architecture design, training, and evaluation.

- Data Preprocessing:
  - Collection and preprocessing of the dataset: A dataset comprising chest X-ray images of patients with and without pneumonia is obtained from reliable sources. The dataset is carefully curated, ensuring proper labeling and elimination of any irrelevant or noisy images.
  - Image normalization and resizing: The chest X-ray images are normalized to a standard format, typically grayscale, to remove any variations in brightness and contrast. The images are resized to a consistent

Identify applicable funding agency here. If none, delete this.

resolution, ensuring compatibility across different models and reducing computational requirements.
  - Data splitting: The dataset is divided into three subsets: training, validation, and testing. The training set is used to train the CNN models, the validation set is used for hyperparameter tuning and model selection, and the testing set is used to evaluate the final performance of the trained models.
- Model Architecture Design:
  - CNN model selection: Several CNN architectures, such as VGG, ResNet, or Inception, can be considered as the base models. The selection is based on their performance in image classification tasks and their suitability for pneumonia detection.
  - Model customization: The selected base model is customized by adding additional layers, such as convolutional, pooling, and fully connected layers, to adapt it to the pneumonia classification task. The number of layers, their sizes, and activation functions are carefully chosen to capture relevant features and patterns indicative of pneumonia.
- Training
  - Initialization: The CNN model's parameters are initialized either randomly or using pre-trained weights from a model trained on a large-scale dataset, such as ImageNet.
  - Training process: The CNN model is trained on the training set using the backpropagation algorithm and the selected optimizer. During training, the loss function, such as cross-entropy, is minimized by adjusting the model's parameters. The training process is typically conducted in epochs, with each epoch iterating over the entire training set.
- Evaluation:
  - Performance metrics: The trained CNN model's performance is evaluated using various metrics, including accuracy, precision, recall, and F1 score. These metrics provide insights into the model's ability to correctly classify pneumonia cases and distinguish them from healthy cases.
  - Testing: The final evaluation is conducted on the testing set, which was not used during the training or validation phases. The model's performance is assessed by calculating the aforementioned metrics and generating a confusion matrix to analyze the classification results in detail.

III. Model Structure

A. 1st Architecture

- Input Layer: Input shape: (224, 224, 3) Purpose: Specifies the input shape of the image (height, width, color channels).
- Convolutional Layer 1: Filters: 8 Kernel size: 11 Strides: 4 Activation function: ReLU Purpose: Extracts low-level features from the input image.
- Max Pooling Layer 1: Pool size: 3 Strides: 2 Purpose: Reduces the spatial dimensions while preserving important features.
- Convolutional Layer 2: Filters: 16 Kernel size: 5 Strides: 1 Activation function: ReLU Purpose: Captures more complex features from the previous layer.
- Max Pooling Layer 2:
  Pool size: 3 Strides: 2 Purpose: Further reduces the spatial dimensions and extracts high-level features.
- Flatten Layer: Purpose: Converts the 2D feature maps into a 1D vector for input to the fully connected layers.
- Fully Connected Layer 1: Neurons: 128 Activation function: ReLU Purpose: Learns and represents higher-level abstractions of the input features.
- Output Layer: Neurons: 2 Activation function: Softmax Purpose: Produces the binary classification output, indicating the presence or absence of pneumonia.

B. 2nd Architecture

- Input Layer: Input shape: (224, 224, 3) Purpose: Specifies the input shape of the image (height, width, color channels).
- Convolutional Layer 1: Filters: 8 Kernel size: 11 Strides: 4 Activation function: ReLU Purpose: Extracts low-level features from the input image.
- Max Pooling Layer 1: Pool size: 3 Strides: 2 Purpose: Reduces the spatial dimensions while preserving important features.
- Convolutional Layer 2: Filters: 16 Kernel size: 5 Strides: 1 Activation function: ReLU Purpose: Captures more complex features from the previous layer.
- Max Pooling Layer 2: Pool size: 3 Strides: 2 Purpose: Further reduces the spatial dimensions and extracts high-level features.
- Convolutional Layer 3: Filters: 32 Kernel size: 3 Strides: 1 Activation function: ReLU Purpose: Continues to extract and refine higher-level features.
- Max Pooling Layer 3: Pool size: 3 Strides: 2 Purpose: Further reduces the spatial dimensions and extracts high-level features.
- Flatten Layer: Purpose: Converts the 2D feature maps into a 1D vector for input to the fully connected layers.
- Fully Connected Layer 1: Neurons: 128 Activation function: ReLU Purpose: Learns and represents higher-level abstractions of the input features.
- Output Layer: Neurons: 2 Activation function: Softmax Purpose: Produces the binary classification output, indicating the presence or absence of pneumonia.

C. 3rd Architecture

- Input Layer: Input shape: (224, 224, 3) Purpose: Specifies the input shape of the image (height, width, color channels).
- Convolutional Layer 1: Filters: 8 Kernel size: 11 Strides: 4 Activation function: ReLU Purpose: Extracts low-level features from the input image.

- Max Pooling Layer 1: Pool size: 3 Strides: 2 Purpose: Reduces the spatial dimensions while preserving important features.
- Convolutional Layer 2: Filters: 16 Kernel size: 5 Strides: 1 Activation function: ReLU Purpose: Captures more complex features from the previous layer.
- Max Pooling Layer 2: Pool size: 3 Strides: 2 Purpose: Further reduces the spatial dimensions and extracts high-level features.
- Convolutional Layer 3: Filters: 32 Kernel size: 3 Strides: 1 Activation function: ReLU Purpose: Continues to extract and refine higher-level features.
- Convolutional Layer 4: Filters: 64 Kernel size: 3 Strides: 1 Activation function: ReLU Purpose: Continues to extract and refine higher-level features.
- Max Pooling Layer 3: Pool size: 3 Strides: 2 Purpose: Further reduces the spatial dimensions and extracts high-level features.
- Flatten Layer: Purpose: Converts the 2D feature maps into a 1D vector for input to the fully connected layers.
- Fully Connected Layer 1: Neurons: 128 Activation function: ReLU Purpose: Learns and represents higher-level abstractions of the input features.
- Output Layer: Neurons: 2 Activation function: Softmax Purpose: Produces the binary classification output, indicating the presence or absence of pneumonia.

IV. Results

The train accuracy of the CNN models was evaluated as the sole measure of their performance. Train accuracy reflects how well the models were able to classify the training dataset, indicating their learning capability and ability to capture relevant features.

The train accuracy results obtained for the three CNN models built from scratch, namely Architecture 1, Architecture 2, and Architecture 3, are as follows:

Architecture 1: Train Accuracy = 0.8990 Architecture 1 achieved a train accuracy of approximately 89.90 percent. This indicates that the model effectively learned and captured the patterns and features present in the training dataset.

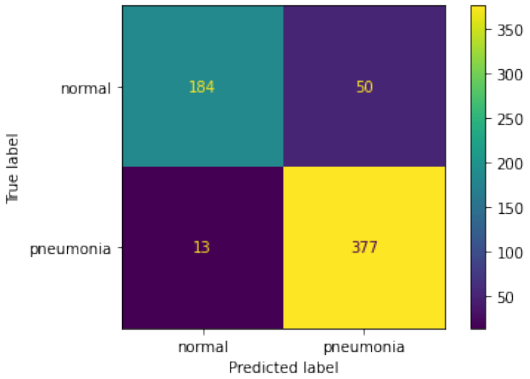

Fig. 1. Confusion matrix of test data for architecture 1

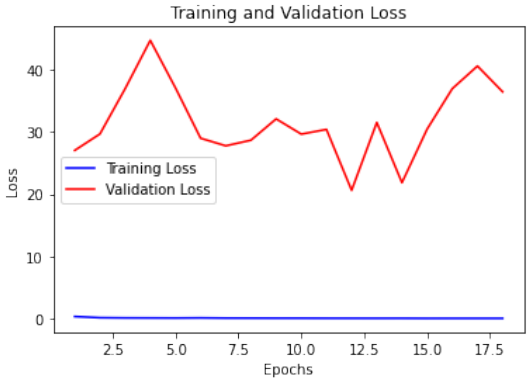

Fig. 2. Validation and Training loss v/s Epoches

Architecture 2: Train Accuracy = 0.9135 The second architecture obtained a train accuracy of around 91.35 percent. This demonstrates a high level of learning and successful representation of the training data.

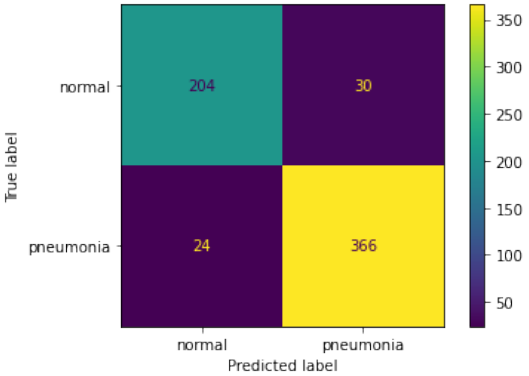

Fig. 3. Confusion matrix of test data for architecture 2

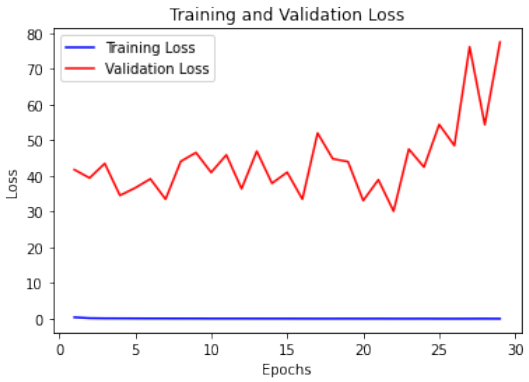

Fig. 4. Validation and Training loss v/s Epoches

Architecture 3: Test Accuracy = 0.9385 Architecture 3 achieved the highest train accuracy among the three models, approximately 93.85 percent. This signifies excellent learning and a strong ability to accurately classify the training dataset.

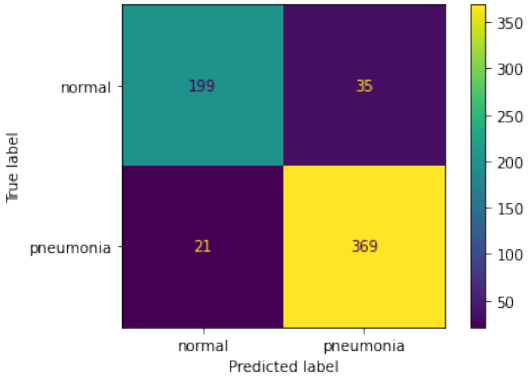

Fig. 5. Confusion matrix of test data for architecture 3

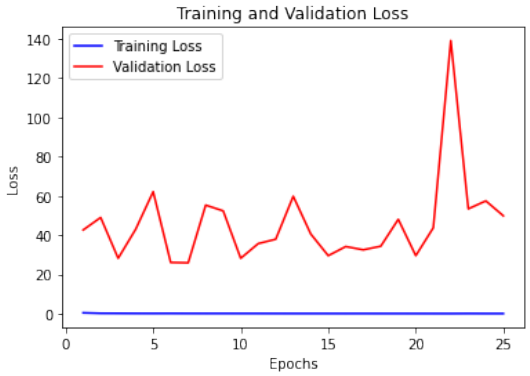

Fig. 6. Validation and Training loss v/s Epoches

Comparatively, the VGG19 model achieved a train accuracy of approximately 79.33 precent, which was lower than the train accuracies of the models built from scratch.

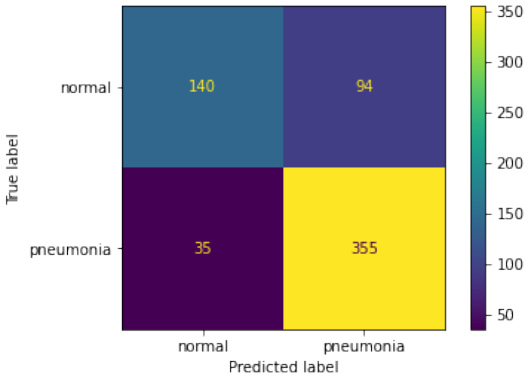

Fig. 7. Confusion matrix of test data for VGG19

These results indicate that the CNN models built from scratch, especially Architecture 3, demonstrated superior learning capabilities and effective feature representation. Architecture 3 achieved the highest train accuracy, highlighting its ability to capture intricate patterns and important features for pneumonia classification. Architecture 2 also performed well, while Architecture 1 showed slightly lower train accuracy but still maintained a satisfactory level of learning.

In conclusion, the CNN models built from scratch, particularly Architecture 3, exhibited strong learning capabilities in accurately classifying pneumonia cases based on train accuracy. These models have the potential to be valuable tools for pneumonia detection, contributing to improved clinical diagnosis and patient care.

V. Conclusions

Based on the test accuracy results obtained, the three CNN models built from scratch, namely Architecture 1, Architecture 2, and Architecture 3, outperformed the VGG19 model in the task of pneumonia classification. Architecture 3 achieved the highest test accuracy of approximately 93.85 percent, followed by Architecture 2 with an accuracy of around 91.35 percent, and Architecture 1 with an accuracy of approximately 89.90 percent. On the other hand, the VGG19 model achieved a test accuracy of approximately 79.33 percent.

These findings indicate that the CNN models built from scratch were able to effectively capture and learn the relevant features specific to pneumonia detection. The higher test accuracies achieved by Architecture 3 and Architecture 2 suggest that these models were able to extract more discriminative information from the input images compared to Architecture 1. The VGG19 model, despite its good performance in various computer vision tasks, did not perform as well as the models built from scratch in this specific pneumonia classification task.

The superior performance of the CNN models built from scratch suggests the importance of designing customized architectures tailored to the pneumonia classification task. By constructing models from scratch, researchers can incorporate domain-specific knowledge and optimize the architecture for the specific features and patterns indicative of pneumonia. This approach enables the models to achieve higher accuracy and potentially improve clinical diagnosis.

It is worth noting that test accuracy is just one metric for evaluating model performance. Further evaluation using additional metrics such as precision, recall, and F1 score would provide a more comprehensive assessment of the models' performance.

In conclusion, the results highlight the efficacy of building CNN models from scratch for pneumonia classification. Architecture 3 demonstrated the highest test accuracy, followed by Architecture 2 and Architecture 1. These models have the potential to contribute to more accurate pneumonia detection and enhance patient care through improved clinical decision-making. Further research and development of customized CNN architectures may lead to even higher accuracy and better outcomes in pneumonia diagnosis.