Development of a Convolutional Neural Network Model for COVID-19 Detection from Chest X-ray Images

Truong Tan Phong - 20126053[1], Luu Minh Nhat - 20126047[1], Tran Tat Tri -20126030[1]

[1]Department of Computer Vision, Ho Chi Minh City University of Science.

Abstract

The COVID-19 pandemic has created an urgent need for rapid and accurate diagnostic methods to manage and contain the spread of the virus. Traditional diagnostic techniques, such as RT-PCR, though effective, are often time-consuming and resource-intensive. This study aims to develop a convolutional neural network (CNN) model to detect COVID-19 from chest X-ray images, offering a faster alternative for diagnosing the disease. We utilized a dataset comprising chest X-ray images categorized into three classes: Normal, Viral Pneumonia, and COVID-19. The proposed CNN model was trained and validated on this dataset, achieving an accuracy of 95.65% in distinguishing between these classes. The results demonstrate the model's potential as a reliable supplementary diagnostic tool, particularly in resource-limited settings where rapid decision-making is critical. This study underscores the value of integrating deep learning techniques into medical diagnostics, providing a foundation for further research and development in AI-driven healthcare solutions.

Keywords: COVID-19, chest radiograph, convolutional neural network, deep learning, medical imaging, diagnostic tool

1 Introduction

The COVID-19 pandemic has profoundly affected global health, economies, and daily life, underscoring the need for rapid and accurate diagnostic tools. Current standard diagnostic methods, such as reverse transcription polymerase chain reaction (RT-PCR), although accurate, can be time-consuming, resource-intensive, and not

universally accessible, especially in resource-limited settings. In contrast, chest radiographs, which are more readily available and administered more quickly, present a viable alternative for immediate clinical decision-making.

Recent advances in deep learning (DL), particularly convolutional neural networks (CNNs), have shown remarkable potential for the analysis of medical images, including chest radiographs. CNNs can identify complex patterns that might be difficult for the human eye to discern, thus improving diagnostic accuracy and efficiency. For COVID-19, early and accurate detection via chest X-ray images could be instrumental in mitigating the outbreak by facilitating prompt treatment and isolation of infected individuals.

This research aims to develop a CNN-based model capable of classifying chest radiograph images into three categories: Normal, Viral Pneumonia, and COVID-19. Unlike previous studies, which focus primarily on binary classification (COVID-19 vs. Non-COVID-19), our study seeks to address the challenge of differentiating between viral pneumonia and COVID-19 - a task complicated by their visual similarities on X-ray images. By achieving this, the model could serve as a supplementary diagnostic tool, particularly valuable in scenarios where RT-PCR testing is limited or delayed.

In summary, the development of a CNN model for COVID-19 detection from chest radiographs represents a promising approach to enhancing diagnostic capabilities. Our work contributes to ongoing efforts to strengthen healthcare systems in the fight against this global health crisis, offering a scalable and rapid diagnostic solution.

2 Method

2.1 Data Preprocessing

The dataset used in this study consists of chest X-ray images categorized into three classes: Normal, Viral Pneumonia, and COVID-19. Prior to training, the images were resized to 224x224 pixels to maintain uniformity in input dimensions for the CNN. All images were normalized to a range of pixel values of 0 to 1, using the overall mean and standard deviation of the data set, which were calculated by averaging the pixel values in all images.

2.2 Model Architecture

The proposed CNN architecture is designed to effectively extract hierarchical features from chest X-ray images for the purpose of distinguishing between Normal, Viral Pneumonia, and COVID-19 cases. The architecture consists of four convolutional layers, each followed by a batch normalization layer and a max pooling layer, as depicted in Figure 1.

2.2.1 Convolutional Layers

The convolutional layers are the core components of the Convolutional Neural Network (CNN) architecture. This network consists of four convolutional layers, with the number of filters progressively increasing across these layers.

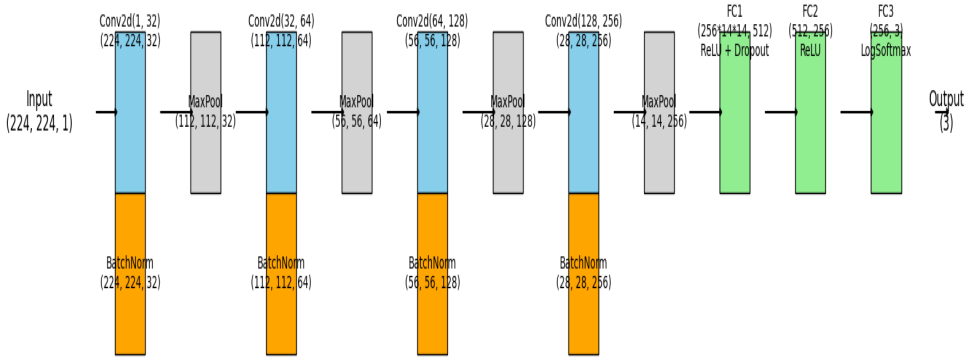

Fig. 1 Architecture of the Convolutional Neural Network (CNN) consisting of 4 convolutional layers (Conv2d) with an increasing number of filters from 32 to 256. Each convolutional layer is followed by a batch normalization (BatchNorm) layer and a max pooling (MaxPool) layer. After passing through the convolutional and pooling layers, the output is fed into three fully connected (FC) layers with 512, 256, and 3 neurons, respectively. The first of these FC layers includes dropout to reduce overfitting. The final layer uses the softmax activation function to classify the output into three classes

- **First Layer**: Uses 32 filters with a kernel size of 3x3.
- **Subsequent Layers**: The number of filters increases to 64, 128, and finally 256.

This progressive increase in the number of filters allows the network to learn more complex and abstract features as the input data passes through the layers. In the initial layers, the network captures simple features like edges and textures. The deeper layers then learn more complex patterns such as shapes and structures, which is essential for classifying chest X-ray.

Each convolutional operation is followed by a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model, enabling it to learn intricate patterns within the data.

2.2.2 Batch Normalization Layers

Following each convolutional layer, a batch normalization layer is applied to stabilize the learning process and accelerate convergence. Batch normalization normalizes the output of the convolutional layers by adjusting and scaling the activations, which helps reduce the internal covariate shift, leading to faster training and potentially better generalization.

2.2.3 Max Pooling Layers

Max-pooling layers are incorporated after each batch normalization layer to downsample the feature maps, reducing their spatial dimensions while retaining the most important features.

2.2.4 Fully Connected Layers

After the convolutional and pooling layers, the output feature maps are flattened into a one-dimensional vector and passed through three fully connected layers with 512, 256, and 3 neurons, respectively.

- **First Fully Connected Layer**: Includes a dropout layer with a rate of 0.5 to reduce overfitting. Dropout randomly sets a fraction of input units to zero during training, encouraging the network to learn more robust features and improving generalization on unseen data.

2.2.5 Softmax Output Layer

The final fully connected layer outputs a vector of three values corresponding to the three classes (Normal, Viral Pneumonia, and COVID-19). These values are passed through a softmax activation function, which converts them into a probability distribution over the classes. The class with the highest probability is selected as the model's prediction.

Justification of the Architecture

This architecture was chosen based on the need to balance model complexity with performance. The progressive increase in the number of filters allows the model to capture features from simple to complex, enabling it to distinguish between visually similar conditions, such as Viral Pneumonia and COVID-19.

The combination of batch normalization and dropout layers ensures that the model is stable during training and less prone to overfitting. Additionally, the softmax activation function in the final layer is particularly suited for multi-class classification, providing a clear and interpretable prediction in the form of class probabilities.

This carefully designed architecture strikes a balance between depth and computational efficiency, making it well-suited for real-time applications in clinical settings where quick and accurate diagnoses are essential.

The CNN model was trained using the categorical cross-entropy loss function, a standard choice for multi-class classification tasks. The Adam optimizer was employed with an initial learning rate of 0.001. The model was trained for 30 epochs with a batch size of 32. Early stopping was implemented to prevent overfitting, with the training halting if the validation loss did not improve for 5 consecutive epochs.

2.3 Loss Function

In our model, we employ the Cross-Entropy Loss function, a standard choice for multi-class classification tasks. The Cross-Entropy Loss function is particularly effective because it measures the dissimilarity between the true labels and the predicted probabilities, directly influencing the optimization of the model during training. It is mathematically defined as:

$$\text{Loss} = -\sum_{c=1}^{C} y_c \log(\hat{p}_c)$$

where:

- C is the number of classes (in this case, 3: COVID-19, Normal, Pneumonia).
- y_c is a binary indicator (0 or 1) that denotes whether class label c is the correct classification for the input.
- $\hat{p}_c$ is the predicted probability of the input belonging to class c.

The Cross-Entropy Loss is particularly powerful due to its ability to penalize the model heavily when it is confident in an incorrect prediction, as represented by a high value of $\hat{p}_c$ when $y_c = 0$ .This characteristic encourages the model to adjust its weights more significantly in response to errors, thereby speeding up convergence to a more accurate solution.

Considering that the task involves distinguishing between three classes: COVID-19, normal, and pneumonia, cross-entropy loss is particularly appropriate. The model not only needs to classify an input correctly but also to provide a reliable estimate of its confidence in that classification. In medical image classification tasks like this one, confidence is crucial, as it directly affects the reliability of the diagnosis. Cross-Entropy Loss helps ensure that the model's confidence is well-calibrated, which is vital in clinical settings where false positives or negatives can have serious consequences.

2.4 L2 regularization

The L2 regularization, also known as Ridge regularization, is a technique in machine learning used to reduce the complexity of a model by imposing a penalty on the sum of the squared values of the coefficients in the model. L2 regularization is often applied in classification problems, especially with complex models like convolutional neural networks (CNNs) in multi-class classification tasks, to help prevent overfitting and enhance the model's generalization ability.

If θ is the vector of parameters in the model, the objective function in a multi-class classification problem with L2 regularization is expressed as follows:

$$L(\theta) = \text{Loss}(\theta) + \frac{\lambda}{2} \sum_{i=1}^{n} \theta_i^2$$

where:

- $\text{Loss}(\theta)$ is the original loss function.
- λ is the regularization parameter, with $\lambda \geq 0$.

- $\sum_{i=1}^{n} \theta_i^2$ is the sum of the squared coefficients.

2.5 Softmax Function

The Softmax function is applied to the output layer of our neural network to convert raw logits into probabilities. This is crucial in multi-class classification, as it ensures that the outputs can be interpreted as probabilities that sum to 1. The Softmax function is defined as:

$$\hat{p}_c = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}}$$

where:

- z_c is the raw score (logit) for class c.
- $\hat{p}_c$ is the resulting probability for class c.
- The denominator sums the exponentiated logits across all classes, normalizing the outputs.

This normalization allows us to interpret the model's output as the probability distribution over the classes, making it easier to assess the model's confidence in its predictions.

3 Experiments

3.1 Dataset

The chest X-ray image dataset plays a crucial role in medical research. Specifically, this dataset is essential for diagnosing COVID-19 and pneumonia, two respiratory conditions that significantly impact global health. The primary purpose of this dataset is to support the development of diagnostic tools capable of accurately distinguishing between COVID-19, viral pneumonia, and normal lung conditions through the analysis of chest X-ray images.

The dataset comprises a large number of chest X-ray images meticulously categorized into three distinct classes: COVID-19, Normal, and Viral Pneumonia. In total, there are 6432 images, with 80% allocated for training and 20% for testing. This balanced distribution ensures that machine learning models trained on this dataset can achieve high accuracy and generalizability.

Each image in the dataset varies in dimensions, reflecting the diversity of real-world medical imaging. The images primarily follow the posteroanterior (PA) view, which is standard practice in chest radiography.

3.2 Metric

In our experiments, we used several key metrics to evaluate the performance of our model:

- **Accuracy**: This metric measures the proportion of correctly classified instances out of the total number of instances. It provides a general assessment of the model's

Fig. 2 Chest X-ray images representing three main classes in the dataset: COVID-19, Normal, and Pneumonia. The images illustrate the distinct lung conditions associated with each class.

overall performance. Accuracy is calculated using the formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **Precision**: Precision is the ratio of true positive predictions to the sum of true positive and false positive predictions for each class. It indicates the model's ability to make accurate predictions for a specific class. Precision is given by:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall**: Recall, also known as Sensitivity, measures the proportion of actual positives that were correctly identified by the model. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score**: The F1 Score is the harmonic mean of Precision and Recall, providing a single metric that balances both aspects. It is especially useful when dealing with imbalanced datasets. The F1 Score is calculated as:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix**: The confusion matrix is a comprehensive tool for visualizing the performance of the classification model. It displays the counts of true positives, false positives, true negatives, and false negatives for each class, allowing for a detailed analysis of classification performance.

3.3 Experimental Setup

In this section, we describe the experimental setup used to evaluate the proposed method. This includes details on the hardware and software configurations, model architecture, training procedure, and hyperparameters.

- **Hardware and Software**: The experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 3050 GPU, an Intel Core i7-11370H CPU, and 16GB of RAM. The software environment included the following:
  - **Operating System**: Windows 11.
  - **Development Environment**: Jupyter Notebook.
  - **Programming Language**: Python version 3.10.11.
- **Model Architecture**: The model used in our experiments is based on a custom neural network. The architecture consists of four convolutional layers, pooling layers, fully connected layers.
- **Training Procedure**: The model was trained using Adam optimizer with a learning rate of 0.001. Additionally, early stopping was employed after 5 epochs to prevent overfitting.
- **Hyperparameters**: The following hyperparameters were tuned for the experiments:
  - **Learning Rate**: 0.001
  - **Batch Size**: 32
  - **Number of Epochs**: 30
  - **Regularization Techniques**: Dropout (0.5), Weight Decay (0.0001)

3.4 Experimental Results

In this section, we present the results obtained from our experiments and analyze their implications. The results include performance metrics for the proposed method and comparisons with baseline models or existing approaches.

Table 1 presents the performance metrics for the proposed model, highlighting its effectiveness across different classes. The model achieves high precision and recall for the covid-19 and pneumonia classes, resulting in F1 scores of 98.71% and 96.74%, respectively. This indicates that the model is highly reliable in identifying these two conditions. The normal class, while still showing strong performance with an F1 score of 91.77%, has slightly lower precision and recall, suggesting some challenges in distinguishing normal cases from pneumonia. In general, the model achieves an impressive accuracy of 95.65%, demonstrating its robustness to classify these conditions accurately.

Table 1 Performance Metrics for the Proposed Model

Class	Precision (%)	Recall (%)	F1-Score (%)
COVID-19	98.29	99.14	98.71
NORMAL	88.79	94.95	91.77
PNEUMONIA	98.08	95.44	96.74
Overall Accuracy	95.65%		

The confusion matrix, presented in Figure 3, demonstrates the model's strong overall performance, particularly in accurately predicting covid19 and pneumonia cases. There is a slight tendency for the model to misclassify normal cases as pneumonia.

However, the low misclassification rates for covid19 and pneumonia suggest that the model is highly effective in distinguishing between these diseases.

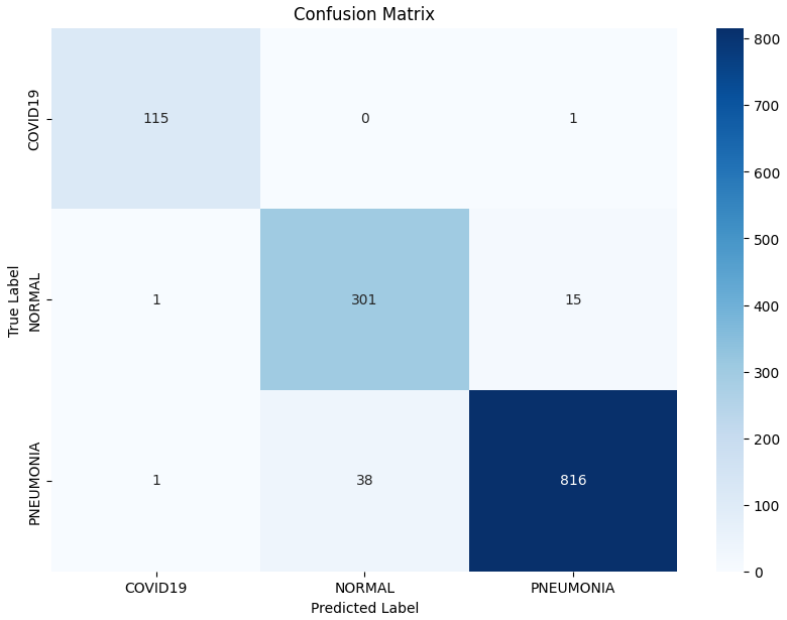

Fig. 3 Confusion matrix showing the classification performance for COVID-19, NORMAL, and PNEUMONIA cases.

Comparison with Baseline Models:

- We compared our proposed method with the ResNet model. The results are summarized in Table 2 3 4 5, where we observe that our method outperforms the baseline model across several key metrics, including accuracy, precision, recall, and F1-Score.
- For instance, our proposed method achieved a higher overall accuracy of 95.65% compared to 93.17% achieved by the ResNet model. Moreover, in the COVID-19 class, our method significantly improved the F1-Score to 0.9871 from 0.9050, demonstrating enhanced reliability in identifying COVID-19 cases.
- Additionally, the proposed model also showed improved performance in distinguishing NORMAL and PNEUMONIA classes, with higher F1-Scores of 0.9177 and 0.9674, respectively, compared to the ResNet model's 0.8889 and 0.9514.

Table 2 COVID-19 Performance Metrics

Model	Precision	Recall	F1-Score
ResNet	0.9524	0.8621	0.9050
Proposed Method	0.9829	0.9914	0.9871

Table 3 NORMAL Performance Metrics

Model	Precision	Recall	F1-Score
ResNet	0.8701	0.9085	0.8889
Proposed Method	0.8879	0.9495	0.9177

Table 4 PNEUMONIA Performance Metrics

Model	Precision	Recall	F1-Score
ResNet	0.9531	0.9497	0.9514
Proposed Method	0.9808	0.9544	0.9674

Table 5 Overall Accuracy Comparison

Model	Overall Accuracy
ResNet	93.17%
Proposed Method	95.65%

3.5 How to Run Code

3.5.1 Required Libraries

The following libraries are required to run the code. These can be installed by running the `pip install -r requirements.txt` command:

- `torch` (PyTorch)
- `seaborn`
- `matplotlib`
- `scikit-learn`
- `torchvision`
- `numpy`

3.5.2 Downloading the Dataset

- If the dataset is not included in the repository, download it from Kaggle Dataset
- Place the dataset in the `data/` directory, or modify the path in the code accordingly.

3.5.3 Running the Notebook

- The code is provided in a Jupyter Notebook (`.ipynb` file). To run the notebook, ensure that you have Jupyter Notebook installed. You can install it using the following command:

```
pip install notebook
```

- To start Jupyter Notebook, navigate to the directory containing your notebook file and run:

```
jupyter notebook
```

- Once Jupyter Notebook is running, open the notebook file and execute the cells sequentially.

4 Conclusion

In this study, we developed and evaluated a Convolutional Neural Network (CNN) model for the detection of COVID-19 from chest X-ray images. The model demonstrated a high degree of precision, achieving an overall precision of 95. 65% in three classes: COVID-19, Normal, and Viral Pneumonia. The results highlight the model's potential as an effective diagnostic tool, particularly in scenarios where rapid decision-making is critical, and traditional methods such as RT-PCR may be impractical due to time or resource constraints.

The successful differentiation between COVID-19 and other conditions like Viral Pneumonia underscores the effectiveness of our approach in handling complex visual similarities, which are challenging even for experienced radiologists. The application of image augmentation techniques and regularization methods like dropout and L2 regularization played a crucial role in mitigating overfitting, thereby improving the generalizability of the model.

While the results are promising, further validation on larger and more diverse datasets is necessary to ensure the robustness and reliability of the model in real-world clinical settings. Additionally, integrating this CNN-based approach with other diagnostic modalities could enhance its effectiveness, paving the way for more comprehensive AI-driven healthcare solutions.

In conclusion, the developed CNN model represents a significant step towards leveraging deep learning for rapid and accurate COVID-19 diagnosis. Future work will focus on optimizing the model's performance, exploring its integration into clinical workflows, and expanding its application to other respiratory diseases, thereby contributing to the broader field of AI in medical diagnostics.

References

[1] Linda Wang and Alexander Wong, *COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images*, 2020, arXiv: 2003.09871 [eess.IV].