Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33520

A Study on a Multi-Classification Model for Chest X-Ray Images Based on Convolutional Neural Networks and Transfer Learning

Jiwen Zhang

Department of Computer Science and Technology, Nanjing University, Nanjing, China
ztomzhangjiwen@gmail.com

Abstract. Currently, regarding the diagnosis of pulmonary diseases, especially the Emphysema, Pneumonia, Tuberculosis, it is mainly up to the physicians' judgment. To be specific, they distinguish different types of pulmonary diseases by observing subtle differences in image features on X-ray images. This mainstream diagnostic method is very time consuming and laborious and maybe leads to delayed diagnosis and misdiagnosis. Therefore, it is essential to develop a productive, automated and intelligent method to assist physicians to identify disease category. This study uses CNNs and transfer learning to train a multi classification model which meets the above requirements in Edge Impulse platform. About 14,000 chest X-ray images were used as the training set. This model performed well in independent test set and it reached the level of preliminary application although it still had some limitations in the classification of certain categories. This study provides some guidance for the future development of this field of technology.

Keywords: CNN, Edge Impulse, Chest X-ray

1. Introduction

Nowadays, pulmonary diseases have endangered human health dramatically [1]. The pulmonary diseases such as Emphysema, Tuberculosis and Pneumonia are very common. The main method to diagnose them is to identify the subtle differences in image features on X-ray images. However, this method which heavily relies on the professional knowledge of physicians is very time consuming and laborious. Furthermore, a large number of diagnostic requirements may cause many problems such as misdiagnosis and delayed diagnosis. Therefore, in order to improve diagnostic efficiency and avoid the above problems, it is very urgent and essential to develop an automated, rapid and accurate computer tool to analyse the chest X-ray images.

In recent years, deep learning has become the critical and important technology in the field of artificial intelligence. It can construct a deep neural network architecture and learn complex features from massive data automatically, which can improve the efficiency and accuracy in the field of identification and classification dramatically. In particular, Convolutional neural network (CNNs) [2], which is the representative deep learning structure in the field of visual information processing, has unique technology of local connection, weight sharing and spatial subsampling. It can decrease

DOI: 10.54254/2755-2721/2026.GU33520

the computational complexity of processing high dimensional data like image information significantly and performs well in the image classification and other related tasks. In medical image analysis, it has significant progress and shows good applicability. It has efficient performance with high accuracy in the classification tasks of pulmonary diseases [3]. In the previous studies, Artificial Intelligence (AI) models have been applied to identify the diseases such as pneumonia and tuberculosis successfully [4]. However, the existing research still has some limitations. For example, most studies focus on classifying a limited number of specific diseases, while research on classifying chest images of six common pulmonary disease status including COVID-19, emphysema, bacterial pneumonia, viral pneumonia, tuberculosis, and normal within the same framework simultaneously remains relatively few.

This study aims to fill in the gaps mentioned above by developing a high-precision lightweight multi category classification model which is used to identify the six common pulmonary disease status mentioned above in order to improve the artificial intelligence assisted pulmonary disease diagnosis. To this end, a dataset containing chest X-ray images of the six pulmonary diseases [5] mentioned above was collected. This study employed Edge Impulse [6] machine learning development platform to develop and train a deep learning model for six-category classification (COVID-19, Emphysema, Pneumonia-Bacterial, Pneumonia-Viral, Tuberculosis, Normal) of chest X-ray images. The performance evaluation indicated that the trained model decreased computing resource requirements dramatically while maintaining high accuracy. It promoted development of low-cost and portable artificial intelligence assisted diagnostic tool.

2. Method

2.1. Dataset preparation

The dataset used in this study was sourced from the publicly available chest X-ray image dataset [5] on Kaggle [7], a global online platform for data science and machine learning. To be specific, the training set comprises 14,551 chest X-ray images, with the distribution as follows: 2,671 images for the Normal class, 2,400 for Pneumonia-Bacterial, 2,413 for Pneumonia-Viral, 2,417 for COVID-19, 2,600 for Tuberculosis, and 2,050 for Emphysema. The test set comprises 1737 chest X-ray images, with the distribution as follows: 300 images for the Normal class, 300 for Pneumonia-Bacterial, 300 for Pneumonia-Viral, 300 for COVID-19, 287 for Tuberculosis, and 250 for Emphysema. The original images all have grayscale color depth and their size are 224×224 pixels. In this study, however, in order to meet the input specification requirements of pre-trained model selected on Edge Impulse platform, the size of images were resized to 160×160 pixels and the image format were converted to RGB. In training phase, the default data augmentation methods were applied.

2.2. Edge impulse-based CNN model

2.2.1. Introduction of CNN

CNN's main idea is to learn hierarchical Abstract Representation automatically from low level visual features to advanced semantic concepts by using a mechanism of local connection, weight sharing and hierarchical feature extraction [2,8-10]. To be specific, it adopts a learnable convolution kernel which slide over the input data to extract local features. Through weight sharing, it can decrease the number of model parameters dramatically while ensuring the translation invariance. Furthermore, it can expand receptive field gradually and enhance the robustness of the learned features by applying

DOI: 10.54254/2755-2721/2026.GU33520

pooling operation to perform spatial downsampling. A typical CNN architecture usually has a feature extractor which consists of stacked convolutional layers, activation layer, and pooling layer. Then it is connected to one or more fully connected layers acting as a classifier. This design achieves a end to end mapping from original input to final classification decision.

2.2.2. Introduction of transfer learning

Transfer learning refers to transferring knowledge and model parameters learned from one task (the source domain) to another related but different task, then making appropriate adjustments to promote the learning of the new task. In the image classification tasks, the commonly adopted form is using the deep CNN pre-trained on a large-scale universal image dataset as a feature extractor or fine-tuning it to adapt the new specific classification tasks. This method is mainly based on a key principle: different tasks share bottom level feature representation. For example, the basic visual characteristics learned from other images can be transferred to medical image analysis. Transfer learning can decrease the data requirements of target domain, shorten training time and enhance convergence stability dramatically.

2.2.3. Introduction of edge impulse

Edge Impulse aims to simplify the whole workflow of building, training and deploying artificial intelligence through a fully integrated, cloud based toolchain. This platform can abstract and automate the traditional and complex machine learning workflow. Through automatic signal processing (e.g., MFCC for audio, preprocessing for images) and model optimization techniques, it can generate highly optimized reasoning models for the specific edge hardware. A typical Edge Impulse project usually follows a basic workflow, which consists of "Data Acquisition," "Impulse Design (i.e., combining processing blocks and learning blocks)," "Model Training," "Testing & Verification," and "Deployment."

2.2.4. This experiment

In this study, the preprocessed dataset (including the training and test sets) was uploaded to the Edge Impulse project. The width and height of all images were set to 160 pixels and the resizing mode was set to "fit shortest axis."

The chosen processing block was "Image" and learning block was "Transfer Learning (Images)." The processing block "Image" has a standard image preprocessing process which is specially designed for processing image data. The learning block "Transfer Learning (Images)" uses transfer learning technology, this technology utilizes general visual features learned from large scale universal image dataset by pre-trained deep neural network (such as the MobileNet series). This method can decrease the training cycles and computing resources of new specific tasks dramatically while achieving good training results.

In the processing block parameter settings, the image color depth was set to RGB. The resulting feature distribution is shown in Figure 1.

DOI: 10.54254/2755-2721/2026.GU33520

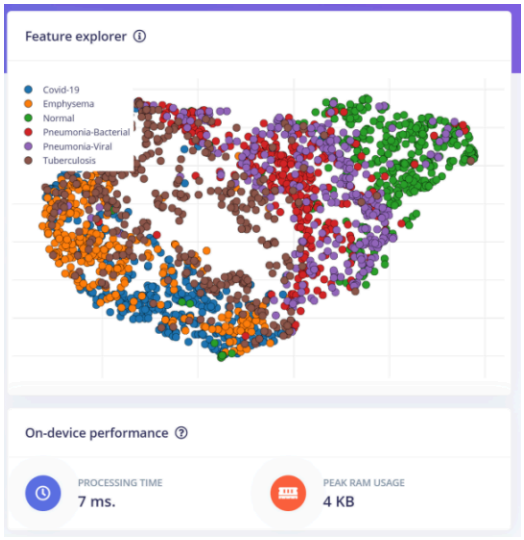

Figure 1. Feature explorer of dataset (picture credit: original)

In the learning block parameter settings shown in Figure 2, the selected pre-trained model was MobileNet V2 160×160 0.35, the number of training epochs was set to 15, the learning rate was set to 0.0005, and data augmentation was enabled.

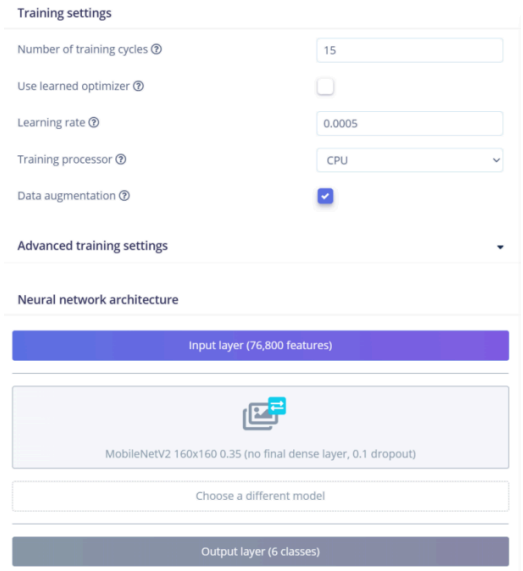

Figure 2. Training parameter settings (picture credit: original)

3. Results and discussion

3.1. Provide the results

In this study, the final training results of the model are presented as follows, including performance data for both the Quantized (int8) and Unoptimized (float32) versions. The related results are provided in Table 1, Figure 3, Figure 4, Figure 5 and Figure 6.

DOI: 10.54254/2755-2721/2026.GU33520

Table 1. Training results

Model version	Accuracy	Loss	Inferencing time	Peak RAM usage	Flash usage
Quantized (int8)	76.2%	1.34	2947ms	416.1K	713.0K
Unoptimized (float32)	86.9%	0.33	4279ms	1.1M	2.3M

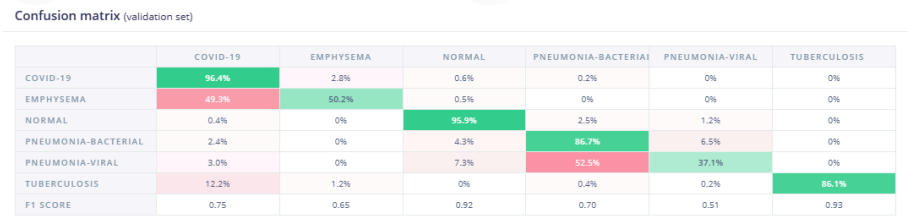

Figure 3. Confusion matrix(validation set) of quantized (int8) version (picture credit: original)

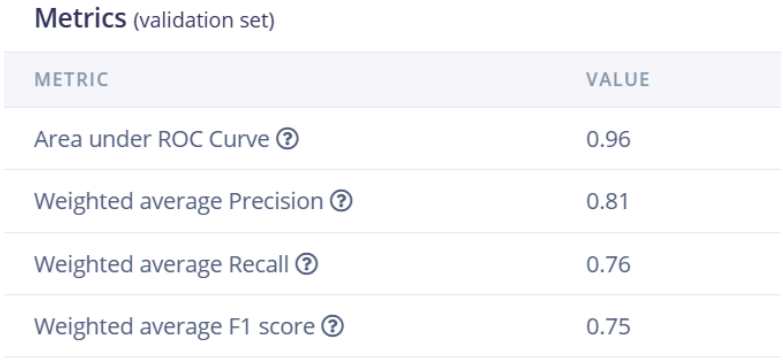

Figure 4. Metrics(validation set) of quantized (int8) version (picture credit: original)

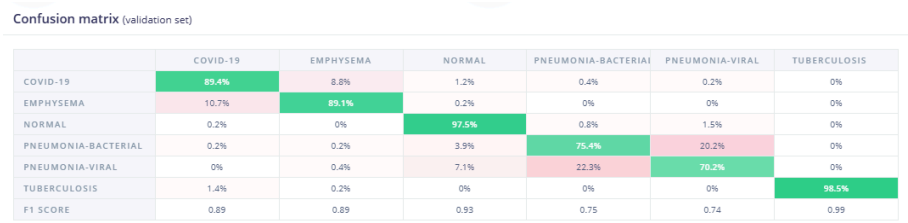

Figure 5. Confusion matrix(validation set) of unoptimized (float32) version (picture credit: original)

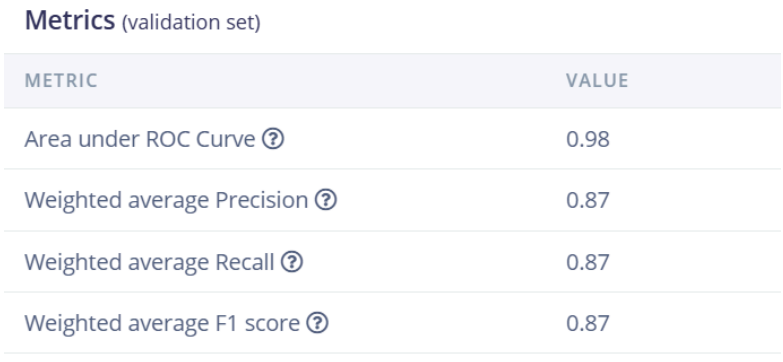

Figure 6. Metrics(validation set) of unoptimized (float32) version (picture credit: original)

DOI: 10.54254/2755-2721/2026.GU33520

The final model testing results of the model are presented as follows, including performance data for both the Quantized (int8) and Unoptimized (float32) versions. The related results are provided in Table 2, Figure 7, Figure 8, Figure 9 and Figure 10.

Table 2. Model testing results

Model version	Accuracy
Quantized (int8)	72.34%
Unoptimized (float32)	83.14%

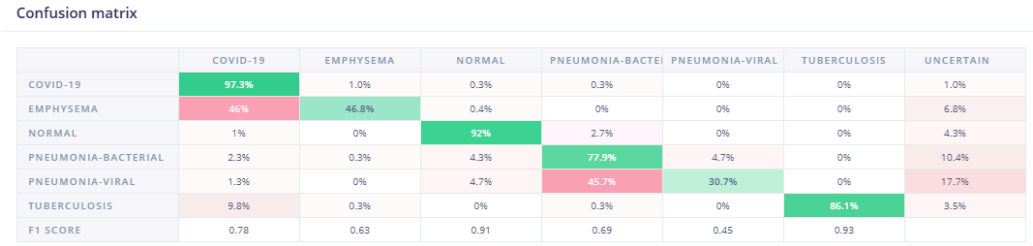

Figure 7. Confusion matrix of quantized (int8) version (picture credit: original)

Figure 8. Metrics for transfer learning of quantized (int8) version (picture credit: original)

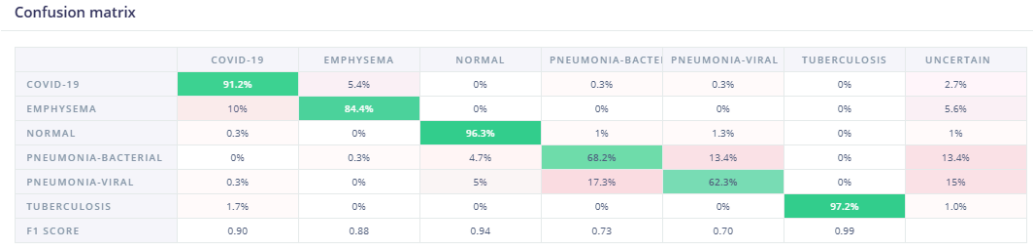

Figure 9. Confusion matrix of unoptimized (float32) version (picture credit: original)

DOI: 10.54254/2755-2721/2026.GU33520

Metrics for Transfer learning

METRIC	VALUE
Area under ROC Curve	0.98
Weighted average Precision	0.86
Weighted average Recall	0.87
Weighted average F1 score	0.86

Figure 10. Metrics for Transfer learning of unoptimized (float32) version (picture credit: original)

3.2. Describe the results

The results for the Quantized (int8) version show a classification accuracy of 76.2% and a loss value of 1.34 on the validation set, and an accuracy of 72.34% on the independent test set. Key metrics indicate that the model achieved an AUC of 0.96 and a weighted average F1 score of 0.74. It indicates that the model demonstrates strong recognition capability for "COVID-19" and "Tuberculosis" (with accuracies reaching 97.3% and 86.1%, respectively), but shows difficulty in distinguishing "Emphysema" and "Viral Pneumonia." Specifically, 45.7% of the samples belonging to the latter are misclassified as "Bacterial Pneumonia." After optimization with the EON™ compiler for RAM efficiency, the model achieved an average inference time of 2,947 ms, a peak RAM usage of 416.1 KB, and a flash memory usage of 713.0 KB in a simulated edge device environment.

According to the results of unoptimized (float32) model version, in the validation set, it achieved a significantly higher classification accuracy of 86.9% and lower loss value of 0.33. In the independent testing set, it achieved an accuracy of 83.14%, the AUC was 0.98, the weighted average F1 score was 0.86, they were all better than the quantized version. According to the confusion matrix, the recognition accuracy of all categories all improved. To be specific, the recognition rate of "Normal" category reached 97.5%, the recognition rate of "Tuberculosis" reached 98.5%. However, there is still a certain degree of mutual confusion between "Bacterial Pneumonia" and "Viral Pneumonia." Under the same optimization settings, the average inference time per prediction of this model version increased to 4279 milliseconds, the peak RAM usage was 1.1 MB, the Flash usage was 2.3 MB.

3.3. Analyze and discuss the results

The Unoptimized (float32) model retains the complete floating-point precision so it can achieve higher classification accuracy comparing to the Quantized (int8) version. However, the cost of this advantage is the significantly increased inference time and resource consumption. Therefore, in the test set, the former achieved the classification accuracy of 83.14%, which was clearly superior to the latter (72.34%). In contrast, the Quantized (int8) version has higher deployment efficiency: its inference time (2947 ms) is approximately 31% faster than the figure for the Unoptimized version (4279 ms). Furthermore, its peak RAM usage (416.1K) and Flash usage (713.0K) are only about the 38% and 31% of the latter respectively.

DOI: 10.54254/2755-2721/2026.GU33520

This model performed well and robustly in identifying COVID-19, Normal, and Tuberculosis. In the tests of both model versions, the recall rates of COVID-19 were more than 91%, the recall rates of Tuberculosis were more than 86%, the recall rates of Normal class were more than 92%. These shows that this model can learn the highly discriminative imaging features of these categories effectively. However, it still has some limitations in distinguishing between emphysema (EMPHYSEMA), bacterial pneumonia (PNEUMONIA-B), and viral pneumonia (PNEUMONIA-V). In the Quantized (int8) version, the classification accuracy of emphysema is only 46.8% and there is confusion between the two types of pneumonia. It reflects the true challenge in the clinical diagnosis. It may be due to the highly overlapping of the image features between the emphysema and normal variations and the significant similarity of image features between bacterial and viral pneumonia.

The current results are restricted by the platform limitations and the choice of model. Because of the constraints of the Edge Impulse free version, in this study, the training epochs was set to 15 and the selected pre-trained model was the lightweight MobileNetV2 0.35, which has less parameters and shallow calculation depth. Although these can improve the convenience of deployment, they restrict the model's ability of learning complex and subtle features. It may be the main reason of the poor distinction between pneumonia subtypes with similar characteristics. The deeper and more powerful pre-trained model may improve the performance, but they were beyond the resource constraints of this experiment.

4. Conclusion

To meet the urgent requirements of automated, efficient and intelligent tools for assisting diagnosis in the field of pulmonary disease diagnosis, aiming at the six pulmonary conditions mentioned above, this study developed an efficient and lightweight multi-class classification model. According to the test results, although this model had some limitations in distinguishing between certain categories, it performed well in the classification tasks of the six pulmonary diseases. Overall, this model has reached the preliminary capability of assisting medical professionals to diagnose pulmonary diseases. However, it still has room for improvement. Through following the method in this study, adopting deeper, more powerful pre-trained model, stricter training parameters, and higher-quality datasets can further improve the model's performance.

References

[1] Oh, J., Kim, S., Yim, Y., Kim, M. S., Hay, S. I., Il Shin, J., & Yon, D. K. (2026). Global, regional, and national burden of chronic respiratory diseases and impact of the COVID-19 pandemic, 1990–2023: a Global Burden of Disease study. Nature medicine, 1-27.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[3] Wang, J., Wang, S., & Zhang, Y. (2025). Deep learning on medical image analysis. CAAI Transactions on Intelligence Technology, 10(1), 1-35.

[4] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[5] https: //www.kaggle.com/datasets/mohamedasak/chest-x-ray-6-classes-dataset

[6] Hymel, S., Banbury, C., Situnayake, D., Elium, A., Ward, C., Kelcey, M., ... & Reddi, V. J. (2022). Edge impulse: An mlops platform for tiny machine learning. arXiv preprint arXiv: 2212.03332.

[7] Twyman, M., Murić, G., & Zheng, W. (2023). Positioning in a collaboration network and performance in competitions: a case study of Kaggle. Journal of Computer-Mediated Communication, 28(4), zmad024.

[8] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

DOI: 10.54254/2755-2721/2026.GU33520

[9] Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23), 495.
[10] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.