# COVID-19 Chest X-ray Image Classification Using a Spiking Neural Network

Md Ashraf Hossain Ifty
*Department of Electrical and Computer Engineering*
*University of Alabama in Huntsville*
mi1497@uah.edu

***Abstract*—Spiking Neural Networks (SNNs) are biologically inspired computational models that offer a promising alternative to conventional deep learning methods, particularly for applications requiring energy efficiency. This study replicates the model architecture presented in *Spiking Neural Network Classification of X-ray Chest Images* (2025), implementing it using the PyTorchSpiking framework as a substitute for KerasSpiking, which presented compatibility challenges. The aim is to classify COVID-19 cases from chest X-ray images using a streamlined convolutional SNN and to evaluate its energy consumption. The replicated model achieved a classification accuracy of approximately 60% and an estimated inference energy cost of 0.00054 kWh, as measured by the CodeCarbon library. Although the performance falls short of the 95% accuracy reported in the original implementation, the results affirm the practicality of reproducing spike-based models on alternative software platforms and underscore the energy-efficient potential of SNNs for medical image classification tasks.**

## ACKNOWLEDGMENTS

The original work was authored by Marco Gatti et al. and published in Knowledge-Based Systems. This replication was conducted for academic purposes.

## I. INTRODUCTION

Spiking Neural Networks (SNNs) represent the third generation of artificial neural networks, inspired by the event-driven nature of biological neurons. Unlike traditional models that rely on continuous activation values, SNNs communicate through discrete electrical spikes triggered when a neuron's membrane potential exceeds a threshold [1], [2]. This biologically plausible behavior allows for asynchronous and sparse computation, enabling energy-efficient information processing.

In recent years, neural network architectures have grown increasingly powerful, but also more computationally intensive. Standard deep learning models often require substantial memory and processing power, making them less suitable for deployment in real-time or resource-constrained environments. In contrast, SNNs operate in a spike-efficient manner, where only active neurons consume energy, making them particularly promising for applications requiring low-power inference [3], [4].

A key feature of SNNs is their ability to encode temporal dynamics and adapt via mechanisms such as spike-timing-dependent plasticity (STDP), which modifies synaptic strength based on the timing of input and output spikes [5]. This makes SNNs not only efficient but also highly expressive in modeling time-dependent behaviors, a capability relevant to many tasks in neuroscience and machine learning.

Despite their advantages, SNNs pose unique training challenges due to their non-differentiable spike functions. While progress has been made through surrogate gradient methods and frameworks like PyTorchSpiking and Nengo [4], [6], further work is needed to match the performance of conventional deep networks in large-scale tasks.

This study investigates the use of SNNs for automated diagnosis of COVID-19 from chest X-ray images. Motivated by the energy constraints of mobile healthcare solutions and wearable devices, we propose a lightweight convolutional SNN model that emphasizes architectural simplicity while maintaining energy efficiency. Unlike more complex CNN or transformer-based models, the proposed approach focuses on real-world feasibility and deployability on neuromorphic platforms such as Intel's Loihi.

The remainder of this paper is structured as follows. Section II reviews related research on deep learning and SNNs in medical image analysis. Section III describes the methodology and network architecture. Section IV presents the experimental results and energy analysis. The key findings and implications of this study are summarized in Section V, while Section VI outlines the current limitations and proposes directions for future research.

## II. LITERATURE REVIEW

Many researchers have explored machine learning approaches, especially artificial neural networks (ANNs), for classifying medical images. These methods have been widely applied to detect COVID-19 from X-rays and CT scans. While deep learning models often perform well, they usually require large amounts of computational power and energy.

For example, the model in [7] combines CT images with patient data using a 3D convolutional neural network to predict severe COVID-19 outcomes. Another study [8] introduced a diagnostic tool designed to run efficiently on multi-core processors and GPUs. In [9], the authors developed a CNN architecture tailored to match the characteristics of the dataset, while [10] used a generative adversarial network (GAN) to improve chest X-ray classification. A broader comparison of deep learning approaches was done in [11], with some models reaching very high accuracy- 96% in [12] and 97% in [13].

Recent work has also included more advanced models, such as transformer-based networks and CNNs with attention mechanisms. In [14], a transformer model was used for video object segmentation by combining image and sound information. Similarly, the model in [15] used gated channels to enhance visual feature extraction. However, these advanced models usually require a lot of processing power, making them less suitable for devices with limited resources, such as mobile health systems.

On the other hand, some studies have explored Spiking Neural Networks (SNNs) for diagnosing COVID-19. For instance, [16] proposed an SNN model that achieved 80.7% accuracy in detecting pneumonia from chest X-rays. Another study [17] used SNNs to classify CT scan images, achieving an impressive F1 score of 0.99 - though training took longer than with traditional models. A more recent paper [18] introduced a deep convolutional SNN that used an optimization algorithm to reduce overfitting, reaching accuracies of 96% for healthy lungs, 99% for COVID-19, and 97% for viral pneumonia.

Although deep learning models often provide slightly better accuracy, SNNs offer a major advantage in terms of energy efficiency. Unlike regular neural networks that are always active, SNNs only process information when spikes occur, saving power. This makes them ideal for low-power and real-time systems, as emphasized in the original study [19].

This paper builds on the original work by attempting to replicate the proposed SNN architecture using PyTorchSpiking in place of KerasSpiking. Although the original implementation was based on the Keras and Nengo ecosystem, this replication used PyTorchSpiking due to compatibility challenges encountered during setup. Specifically, the KerasSpiking library had unresolved dependency conflicts with recent versions of TensorFlow, and its integration with modern Python environments was limited by outdated packages and restricted GPU support. These issues hindered reproducibility and experimentation. As a result, PyTorchSpiking was adopted as a viable alternative that integrates seamlessly with the PyTorch ecosystem, allowing for greater flexibility in training workflows. While the replicated model achieved a lower classification accuracy (approximately 60% compared to the original 95%), it successfully preserved the energy-efficient spiking behavior. This demonstrates the potential of implementing SNNs on alternative platforms and underscores their viability for medical image classification in resource-constrained environments.

## III. Methodology

This section describes the approach used to develop and evaluate a Spiking Neural Network (SNN) aimed at classifying chest X-ray images to detect COVID-19. The goal was to design an architecture that balances simplicity, computational efficiency, and accuracy, making it suitable for low-power, real-time diagnostic applications on portable or edge computing platforms.

The model architecture is based on a convolutional SNN that processes grayscale X-ray images and outputs binary class predictions (COVID or non-COVID). The network is composed of convolutional and fully connected layers, each utilizing spiking neurons that activate only when their membrane potential crosses a specific threshold. This behavior mimics biological neurons and supports energy-efficient, event-driven computation.

The change in a neuron's membrane potential $V(t)$ is modeled as:

$$V(t) = V(t-1) + I(t) - \theta, \tag{1}$$

where $I(t)$ represents the synaptic input at time $t$, and $\theta$ is the firing threshold. A spike is generated when $V(t) \geq \theta$, after which the potential is reset.

Learning in the network is enabled through surrogate gradient methods, which allow spiking neural networks to be trained using conventional backpropagation despite the non-differentiability of spike events. During the forward pass, neurons generate binary spikes based on membrane potential thresholds, while the backward pass uses a continuous approximation of the spiking function's gradient to propagate error. This surrogate approach facilitates efficient optimization using standard gradient-based methods such as the Adam optimizer, and is well-supported by the PyTorchSpiking framework. These techniques enable effective learning without relying on biologically-inspired rules like spike-timing dependent plasticity (STDP).

A typical surrogate gradient approximates the derivative of the spike activation function $S(V)$ using a piecewise linear function:

$$\frac{dS(V)}{dV} \approx \begin{cases} \frac{1}{\gamma} & \text{if } |V-\theta| \leq \frac{\gamma}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $V$ is the membrane potential, $\theta$ is the firing threshold, and $\gamma$ controls the width of the linear region. This approximation enables gradient flow during training despite the binary nature of spikes.

The convolutional layers extract spatial features from the input images, while the fully connected layers perform classification based on these features. Due to the sparse and asynchronous activation of spiking neurons, the model naturally reduces redundant computation, making it well-suited for neuromorphic hardware such as Intel's Loihi.

### A. Dataset and Preprocessing

The model was trained and evaluated using the Extensive and Augmented COVID-19 X-ray and CT Chest Images Dataset [20], a public collection compiled from various open-access sources. It includes a total of 8,734 chest X-ray images, with 3,728 labeled as COVID-19 positive and 5,006 as non-COVID.

Figure 1 shows nine representative samples from the dataset.

To prepare the data, all images were resized to 256×256 pixels, converted to grayscale, and normalized to the [0, 1] range by dividing pixel values by 255. These preprocessing steps reduced input complexity and improved training stability.

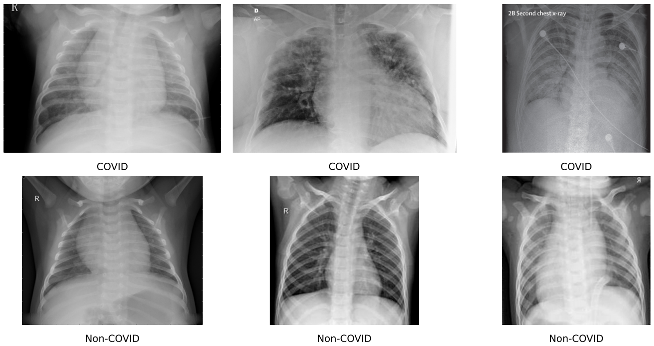


Fig. 1. Chest X-ray samples from the dataset, with COVID-19 positive cases on the top row and non-COVID cases on the bottom.

The dataset was split into training (70%, 6,108 images), validation (14%, 1,226 images), and test (16%, 1,400 images) sets. Each subset preserved the original class distribution to ensure balanced learning. The Adam optimizer and a categorical cross-entropy loss function were used to train the model over 20 epochs with a batch size of 32. Model performance was tracked on the validation set after each epoch to detect overfitting. Final evaluation was performed using the test set, which remained unseen during training.

While this dataset provided a useful starting point, it reflects a single benchmark. Future work will involve training on multiple datasets with varied characteristics, including resolution, imaging noise, and scanner types. Additional tasks such as lung region segmentation may also be incorporated to focus learning on clinically relevant areas, as suggested in [21].

### B. Spiking Activation Mechanism

Training spiking networks presents unique challenges due to the non-differentiable nature of spike events. Unlike conventional networks that use continuous activations, SNNs operate with discrete spikes, making gradient-based learning difficult. To address this, several surrogate gradient methods have been developed.

One such method is SpikeProp [22], which approximates the neuron's membrane potential as a continuous function over time, enabling the use of backpropagation. Its extension, Multi-SpikeProp [4], further models interactions through multiple synapses and has been applied in areas such as seizure detection.

In this work, the PyTorchSpiking library was used to implement spiking behavior within a conventional deep learning framework. Specifically, a spiking version of the ReLU activation was applied, allowing the network to simulate temporal spike activity while remaining compatible with standard gradient descent optimization. These spiking ReLU layers provide spike-frequency outputs suitable for learning and can be trained using backpropagation with the Adam optimizer.

Although the spike timings are not modeled precisely as in biological neurons, the approximation maintains temporal dynamics while enabling efficient training. This method also simplifies experimentation and makes the system easier to deploy on general-purpose hardware, while preserving the energy-aware properties of SNNs.

### C. Convolutional SNN Architecture

The architecture of the proposed model comprises three convolutional layers followed by three fully connected layers, all structured for efficiency and reduced computational overhead. Each convolutional layer employs a 3×3 kernel with increasing filter counts of 8, 64, and 128 to extract progressively abstract spatial features from the input. Importantly, each of these convolutional layers is followed by a spiking ReLU activation, which introduces temporal spiking behavior and supports sparse, event-driven computation. A max-pooling operation with a 4×4 kernel and stride of 2 is applied after the second convolutional layer to reduce spatial dimensionality.

The output of the final convolutional layer is flattened and passed through three fully connected layers with 128, 64, and 8 neurons, respectively. These layers also incorporate spiking ReLU activations, promoting sparsity and enabling effective learning via surrogate gradient methods. The output of the third fully connected layer undergoes temporal average pooling to collapse the time dimension. Finally, a linear output layer with two neurons - corresponding to the COVID-19 and non-COVID classes - is applied, followed by a softmax activation to generate class probabilities.

Although components like max-pooling and softmax are not inherently spike-based, they are preserved for their practical benefits. In this context, they act as functional approximations that integrate well with the temporal dynamics introduced by spiking layers, particularly when combined with temporal pooling.

The overall network was trained using the following cross-entropy loss function:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} y_{i,j}\log(\hat{y}_{i,j}) \tag{3}$$

Here, $n$ is the number of samples, $k$ is the number of output classes, $y_{i,j}$ is the true label, and $\hat{y}_{i,j}$ is the predicted probability. This loss function penalizes incorrect predictions and is suitable for multi-class classification tasks.

Throughout training, performance was tracked using both training and validation sets to monitor for convergence and generalization. The model's architecture prioritizes efficiency and simplicity, making it a strong candidate for real-world deployment in portable, energy-constrained medical imaging systems.

## IV. Results and Discussion

This section presents the experimental evaluation of the proposed Spiking Neural Network (SNN) model, implemented using the PyTorchSpiking framework. The model was tested on a dedicated set of 1,400 chest X-ray images, equally divided between COVID-19 and non-COVID cases. The achieved classification accuracy was 59.15%, which falls considerably short of the 95% accuracy reported in the original reference model that utilized KerasSpiking and the Nengo framework. Despite this performance gap, the results demonstrate that spiking neural dynamics can be effectively approximated using PyTorch, making it a viable platform for prototyping SNN-based architectures.

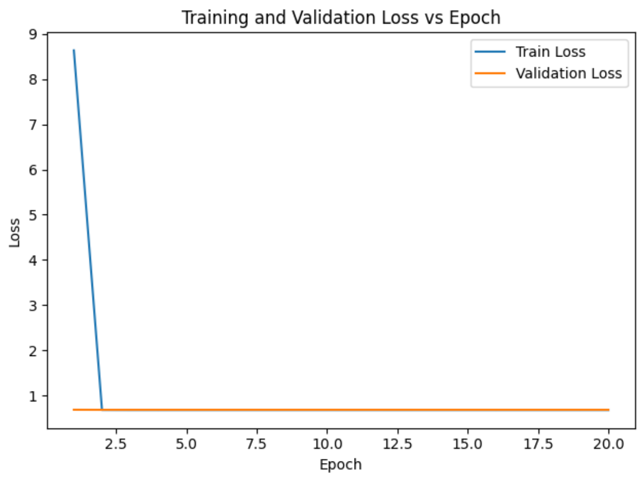


Fig. 2. Training and validation loss over 20 epochs

The model's confusion matrix indicated that all predictions favored a single class, revealing a strong bias and a failure to distinguish between the two categories. This led to an ROC AUC score of 0.5000, which is equivalent to random guessing and highlights the network's inability to learn meaningful decision boundaries. As shown in Figure 2, the training loss rapidly decreased within the first two epochs but then plateaued, while the validation loss remained flat and relatively high throughout. These trends suggest that the model did not generalize well to unseen data and may have suffered from underfitting. Potential contributing factors include the use of a single temporal step for spike encoding, the limited complexity of the model, and challenges in optimizing spiking neuron behavior using surrogate gradients.

To assess the energy efficiency of the replicated implementation, the CodeCarbon library was employed to estimate total energy consumption during inference. When evaluating the full test set, the estimated energy usage was approximately 0.00054 kWh. While this figure exceeds the optimized energy measurement of 0.035 J per inference reported in the original study using Intel's Loihi neuromorphic hardware, it nonetheless reflects the energy-conscious nature of spiking neural networks (SNNs), even when simulated on general-purpose CPUs.

The original study leveraged KerasSpiking's built-in energy profiling functionality to obtain detailed layer-wise energy estimates across different hardware platforms, including CPU, GPU, and Loihi. This allowed for precise per-inference energy values and hardware-specific insights, enabling a direct assessment of the model's efficiency under neuromorphic execution. In contrast, such dedicated energy analysis tools are not available in PyTorchSpiking. Due to this limitation, the present study used CodeCarbon, a system-level tool that estimates energy usage based on CPU utilization and memory consumption during execution.

While CodeCarbon provides a useful approximation of total energy consumption, it does not offer per-layer granularity or hardware-specific breakdowns, and it lacks the ability to measure individual inference cost directly. As a result, the energy values obtained are coarser and less directly comparable to those reported in the original work. Nonetheless, they serve as a practical and accessible proxy to highlight the low-energy characteristics of SNNs, even without specialized neuromorphic support.

Several factors constrained the scope of this study. A significant portion of development time was spent addressing compatibility issues between frameworks, which restricted the opportunity to perform deeper experimentation. Consequently, no parameter tuning, ablation studies, or benchmarking against simpler baseline models (such as standard CNNs or logistic regression) were conducted. Additionally, the model was trained and evaluated only once, and no confidence intervals or statistical variability across multiple runs were computed.

These limitations define clear directions for future research. Enhancements such as incorporating more temporal encoding steps, applying systematic hyperparameter tuning, and repeating experiments across multiple seeds will help evaluate model robustness. Further, deploying the network on neuromorphic hardware platforms will provide a more accurate measure of its energy-saving potential in real-world environments.

## V. Conclusion

This study replicated a previously proposed spiking neural network (SNN) architecture for COVID-19 chest X-ray classification using the PyTorchSpiking framework, in place of KerasSpiking. Although the replicated model did not achieve the high classification accuracy reported in the original study, it maintained the energy-efficient computational principles central to SNNs. The model achieved an accuracy of approximately 60% on the test set and demonstrated low energy consumption during inference, highlighting the viability of SNNs in resource-constrained environments.

However, several challenges limited the scope of experimentation. The prolonged training time typical of SNNs, especially on general-purpose CPUs, restricted the ability to perform extensive hyperparameter tuning or repeated runs for statistical validation. Additionally, adapting the model to PyTorchSpiking required adjustments to the architecture and training pipeline due to the absence of certain features available in KerasSpiking and Nengo.

Despite these limitations, the results affirm the feasibility of using alternative deep learning platforms to simulate spiking behavior and further emphasize the importance of lightweight architectures for deployment on low-power medical diagnostic systems.

## VI. Future Work

There are several directions to improve this work in the future. First, the current model uses only one time step to simulate spikes. Future versions will explore multi-step spike encoding, which can help the network better understand the timing and structure in the input data. Second, different combinations of model settings - such as learning rates, batch sizes, and neuron thresholds - will be tested through systematic hyperparameter tuning. This should improve the model's performance and training stability.

Another important step is to compare the SNN with simpler models like basic CNNs or standard feedforward networks. These comparisons will help understand how much accuracy and energy efficiency the SNN offers compared to more traditional models. Also, future experiments will include multiple training runs with different random seeds to check for consistency and calculate average accuracy and variation.

Lastly, the ultimate goal is to run the trained model on real neuromorphic hardware like Intel's Loihi or BrainScaleS. These platforms are specifically designed for SNNs and can provide precise measurements of energy usage per inference. Running the model on such hardware would give a clearer picture of how well it performs in real-world scenarios and confirm its usefulness for energy-saving applications in healthcare.

Overall, these future efforts aim to improve the model's accuracy, reliability, and practical use, while also offering a more complete evaluation of how SNNs perform in medical image analysis tasks.

## VII. Data and Code Availability

All code used in this experiment is available at: GitHub Repository.

## References


[1] T. Clawson *et al.*, "Spiking neural network (snn) control of a flapping insect-scale robot," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 3381–3388.

[2] E. D. Bois-Reymond, *Studies on Animal Electricity*. Berlin: G. Reimer, 1848, (in German).

[3] W. Maass, "Lower bounds for the computational power of networks of spiking neurons," *Neural Computation*, vol. 8, no. 1, pp. 1–40, 1996.

[4] S. Ghosh-Dastidar and H. Adeli, "A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection," *Neural Networks*, vol. 22, no. 10, pp. 1419–1431, 2009.

[5] M. Gatti, A. Leporati, and C. Zandron, "On spiking neural membrane systems with neuron and synapse creation," *International Journal of Neural Systems*, vol. 32, no. 8, 2022.

[6] W. Gerstner, "Time structure of the activity in neural network models," *Physical Review E*, vol. 51, pp. 738–758, 1995.

[7] L. Meng *et al.*, "A deep learning prognosis model help alert for covid-19 patients at high-risk of death: A multi-center study," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3576–3584, 2020.

[8] G. Goel *et al.*, "Computecovid19+: accelerating covid-19 diagnosis and monitoring via high-performance deep learning on ct images," in *50th International Conference on Parallel Processing*, 2021, pp. 1–11.

[9] V. Dhaka *et al.*, "A deep learning model for mass screening of covid-19," *International Journal of Imaging Systems and Technology*, vol. 31, no. 2, pp. 483–498, 2021.

[10] S. Albahli, "Efficient gan-based chest radiographs (cxr) augmentation to diagnose coronavirus disease pneumonia," *International Journal of Medical Sciences*, vol. 17, no. 10, p. 1439, 2020.

[11] M. Nakamura *et al.*, "Comprehensive study of coronavirus disease 2019 (covid-19) classification based on deep convolutional neural networks," in *SHS Web of Conferences*, vol. 102. EDP Sciences, 2021, p. 04007.

[12] A. Chaddad, L. Hassan, and C. Desrosiers, "Deep radiomic analysis for predicting coronavirus disease 2019 in computerized tomography and x-ray images," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 3–11, 2022.

[13] S. Khan *et al.*, "Covid-19 detection in chest x-ray images using a new channel boosted cnn," *Diagnostics*, vol. 12, no. 2, 2022.

[14] B. Cheng *et al.*, "Scalable video object segmentation with identification mechanism," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[15] Y. Chen *et al.*, "Gated channel transformation for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 794–11 803.

[16] T. Fukuchi *et al.*, "Efficient pneumonia detection method and implementation in chest x-ray images based on a neuromorphic spiking neural network," in *Computational Collective Intelligence*, N. Nguyen, Y. Manolopoulos, R. Chbeir, A. Kozierkiewicz, and B. Trawiński, Eds. Cham: Springer International Publishing, 2022, pp. 311–321.

[17] A. Garain, A. Basu, and F. Giampaolo, "Detection of covid-19 from ct scan images: A spiking neural network-based approach," *Neural Computing and Applications*, vol. 33, pp. 12 591–12 604, 2021.

[18] R. Rajagopal *et al.*, "Deep convolutional spiking neural network optimized with arithmetic optimization algorithm for lung disease detection using chest x-ray images," *Biomedical Signal Processing and Control*, vol. 79, p. 104197, 2023.

[19] M. Gatti, J. A. Barbato, and C. Zandron, "Spiking neural network classification of x-ray chest images," *Knowledge-Based Systems*, vol. 314, p. 113194, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705125002412

[20] W. El-Shafai and E. Fathi, "Extensive and augmented covid-19 x-ray and ct chest images dataset," http://dx.doi.org/10.17632/8h65ywd2jr.2, 2020.

[21] K. Li *et al.*, "Catr: Combinatorial-dependence audio-queried transformer for audio-visual video segmentation," in *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*. ACM, 2023.

[22] S. Bohté, J. Kok, and H. L. Poutré, "Spikeprop: backpropagation for networks of spiking neurons," in *The European Symposium on Artificial Neural Networks*, 2000.