



# Detection of COVID-19 from CT scan images: A spiking neural network-based approach

Avishek Garain<sup>1</sup> · Arpan Basu<sup>1</sup> · Fabio Giampaolo<sup>2</sup> · Juan D. Velasquez<sup>3</sup> · Ram Sarkar<sup>1</sup>

Received: 11 December 2020 / Accepted: 8 March 2021  
© The Author(s) 2021

## Abstract

The outbreak of a global pandemic called coronavirus has created unprecedented circumstances resulting into a large number of deaths and risk of community spreading throughout the world. Desperate times have called for desperate measures to detect the disease at an early stage via various medically proven methods like chest computed tomography (CT) scan, chest X-Ray, etc., in order to prevent the virus from spreading across the community. Developing deep learning models for analysing these kinds of radiological images is a well-known methodology in the domain of computer based medical image analysis. However, doing the same by mimicking the biological models and leveraging the newly developed neuromorphic computing chips might be more economical. These chips have been shown to be more powerful and are more efficient than conventional central and graphics processing units. Additionally, these chips facilitate the implementation of spiking neural networks (SNNs) in real-world scenarios. To this end, in this work, we have tried to simulate the SNNs using various deep learning libraries. We have applied them for the classification of chest CT scan images into COVID and non-COVID classes. Our approach has achieved very high F1 score of 0.99 for the potential-based model and outperforms many state-of-the-art models. The working code associated with our present work can be found here.

**Keywords** COVID-19 · CT scan · Deep learning · Medical image · Spiking neural network

## 1 Introduction

Coronavirus disease 2019 (COVID-19) is an infectious respiratory disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It is responsible for an ongoing global pandemic. It has caused a large number of deaths worldwide, with the count of affected people increasing day by day. It has been detrimental to both the economy and the society. Since no vaccine has been developed, preventive measures such as social distancing, quarantining, use of face-masks, etc., have been recommended. The management of the disease involves prompt identification of infected individuals via wide-scale testing and isolation of the infected population. The optimal mode of testing would be by real-time reverse transcription polymerase chain reaction (rRT-PCR). Chest X-Rays and CT images have also been used, though they are not recommended. However, in many cases, X-Ray and CT images can be beneficial to radiologists in an initial screening process. In this work, an approach for detecting COVID-19 in CT images is discussed.

---

✉ Fabio Giampaolo  
fabio.giampaolo@unina.it

Avishek Garain  
avishekgarain@gmail.com

Arpan Basu  
arpan0123@gmail.com

Juan D. Velasquez  
jvelasqu@dii.uchile.cl

Ram Sarkar  
ram.sarkar@jadavpuruniversity.in

<sup>1</sup> Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

<sup>2</sup> Department of Mathematics and Applications “R. Caccioppoli”, University of Naples Federico II, Naples, NA, Italy

<sup>3</sup> Department of Industrial Engineering, Faculty of Physical and Mathematical Sciences, Instituto Sistemas Complejos de Ingeniería (ISCI), University of Chile, Santiago, Chile

With the advent of high computing power and deep learning algorithms for leveraging the same, deep convolutional neural networks (DCNNs) have brought a drastic upliftment in the performances achieved in the domain of computer vision. In many cases, such models perform way better than human vision in various object recognition tasks. Even in the presence of models with high levels of working efficiency and accuracy, the search for computational algorithms which are inspired by the functioning of brains is continuously growing and has been a centre of attraction for researchers from all over the world. Considering this domain of research, a plethora of architectures and computational models based on spiking neural networks (SNNs) have been proposed. Though DCNNs outperform SNNs in terms of recognition accuracy, the obvious question that arises is that, if performance is not the key then what is the reason for this new trend of increasing interest towards neurobiologically modelled SNNs. Previously the problem used to be the limitation in computational abilities; however, in the present time and immediate future, the need of the hour is environment friendly, power-efficient algorithms and computational devices. The human brain has been evolving since a millions of years. The resultant optimization from this evolution has made this fact possible that the human brain consumes approximately 20W power. This is equivalent to the power consumption of an average laptop. Understanding this efficiency of brain is beyond our current capabilities, but using computation based on spikes has already helped researchers for designing neuromorphic energy-efficient microchips [3, 4]. The era of Internet of Things (IoT) calls for evolution of on-device artificial intelligence (AI)-based models. Biologically inspired learning mechanisms such as spike-timing-dependent plasticity (STDP) [1, 7] can be unimaginably friendly to hardware and might be the perfect match for online on-chip training [31]. Apart from these reasons, the natural ability of SNNs of handling spatio-temporal patterns has inspired the researchers to try various methods for applying SNNs for various visual tasks. Using structured neural networks in a hierarchical manner is one of the most common approaches, but configuration of other hyperparameters like the count of layers, neuron models and information encoding needs a lot of experimentation. In the context of brain-inspired algorithms, STDP-based SNNs are the most biologically plausible. By making use of STDP, the network can successfully extract those visual features which are the most frequently occurring. However, after feature extraction, for the purpose of decision making, external classifiers such as support vector machines (SVMs) and radial basis functions (RBFs) or supervised variants of STDP, are usually required.

Chest CT scans have been a well-proven functionality for providing assistance to the detection of COVID-19. Methods have been proposed for classifying these CT scan images to diagnose patients for COVID-19. However, to the best of our knowledge, application of SNNs for the same has not been considered by any researchers. In this work, we have used a 3-layer DCSNN (deep convolutional spiking neural network) with a structure adopted from [15] for the binary classification of COVID-19 from CT scan images. First, the input image is convolved and processed with Gabor filters at various scales and orientations. Then, by means of an intensity-to-latency encoding [6], a wave of spikes is generated and propagated to the next layer. After propagating through a series of convolutions and pooling layers with neurons which are capable of firing at most once, the spike wave finally reaches the penultimate layer. From this layer, the features are extracted in order to be fed to an external classifier for the purpose of final decision-making for class assignment. For every image that is fed to the network, the neurons present in the final layer with either the earliest spike time or maximum potential contributes to the decision of the network, respectively.

The rest of the paper has been organized as follows: Sect. 2 provides a literature survey about the works done on this topic. Section 3 describes the dataset on which the proposed framework has been evaluated. The methodology followed in the present work is described in Sect. 4. This is followed by the results and concluding remarks in Sects. 5 and 6, respectively.

## 2 Literature survey

Deep learning has been successfully applied in many areas of medical imaging like diabetic retinopathy, histological analysis, cardiac imaging, tumour detection, etc. Consequently, deep learning-based approaches have also been applied to detect COVID-19 using radiological images of chest X-Rays, CT scans, etc.

In the work by [26], the authors have proposed an adaptive feature selection guided deep forest (AFS-DF) for the purpose of COVID-19 detection using the chest CT images. They have first extracted location-specific features from the images and have then applied a deep forest model in order to capture the high-level representation of these features with such small-scale data. They have also proposed a feature selection method for the deep forest model for reducing the redundancy of features. The feature selection method has been adaptively incorporated with the COVID-19 classification model. The authors have evaluated their model on the COVID-19 CT scan dataset with 1495 patients of COVID-19 and 1027 patients of community acquired pneumonia (CAP). Their method has

achieved 91.79% accuracy, 93.05% sensitivity, 89.95% specificity and 96.35% AUC, respectively.

Singh et al. [25], in their work, have used a convolutional neural network (CNN) to classify the COVID-19 patients as infected or not. A noticeable point of this work is that the initial parameters of the CNN have been tuned using multi-objective differential evolution (MODE). The authors have performed extensive experiments by considering their approach and the state-of-the-art machine learning techniques on the chest CT images. The authors have reported that their proposed model can classify the chest CT images at a good accuracy rate of over 90%.

The availability of a large and varied sample of data (X-Rays, CT scans, etc.) is important for the generalizability and predictive power of deep learning based models. However, in reality, there is often a lack of suitable data in some domains. The work by [30] aims to tackle this problem by producing synthetic data of normal and COVID-19-affected chest X-Rays using a generative model. They have developed a modified auxiliary classifier generative adversarial network (ACGAN) which they term as CovidGAN for the generation of synthetic images. The authors have observed that the inclusion of the synthetic data in a VGG16 classifier leads to a marked improvement in the performance metrics. The accuracy and F1 score increase to 95% and 0.94 from 85% and 0.84, respectively.

Several works also apply transfer learning and ensembling approach to improve the performance of the models as opposed to training a model from scratch.

The authors in the work by [14] have applied transfer learning using several models which were previously trained on the ImageNet dataset. They have noted that the DenseNet201 model performs the best, as compared to VGG16, ResNet152V2 and InceptionResNetV2. They have used the SARS-CoV-2 CT scan dataset<sup>1</sup> from Kaggle for evaluating their approach. The dataset consists of 2492 CT scans out of which 1262 are COVID-19 positive and the remaining are COVID-19 negative. The authors have reported training, validation and testing accuracies of 99.82%, 97.40% and 96.25%, respectively.

Similarly, the work by [22] combines ensembling with iterative pruning to detect pneumonia-related and COVID-19-related abnormalities from chest X-Rays. The authors use a combination of various datasets as is mentioned in their work. One notable stage in their pipeline is the modality-specific training. The models are pretrained on a pneumonia-related chest X-Ray dataset to learn task-specific feature representations. The reasoning is that since COVID-19-related data (chest X-Rays) is limited, the pretraining can help the models to generalize better. The

authors report their accuracy and AUC as 99.01% and 0.9972, respectively.

In the work by [19], the authors propose a novel 3D convolutional network with an online attention module for detecting COVID-19 in chest CT images. They perform their training and validation on a multi-centre CT data from eight hospitals comprising of 2186 CT scans from 1588 patients. For the testing stage, a similar dataset of 2796 CT scans from 2057 patients was used. A custom 3D ResNet34 architecture with an attention module is proposed by the authors. Two models are trained, one with uniform sampling from the training data and the other with size-balanced sampling due to class imbalance in the data. The predictions from the two models are then combined using an ensemble learning strategy. The authors report the AUC, accuracy, sensitivity, specificity and F1 score as 0.944, 87.5%, 86.9%, 90.1% and 82.0%, respectively.

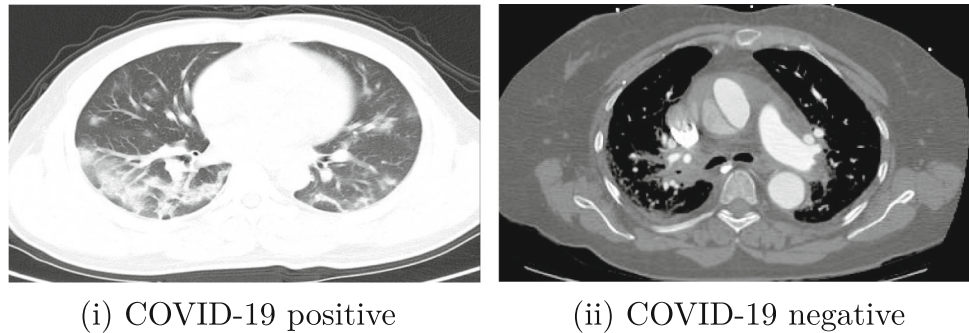
Goel et al. [8] propose an optimized convolutional neural network (OptCoNet) for the automatic diagnosis of Covid-19 from chest X-Rays. The proposed architecture is composed of feature extraction and classification components like a CNN. However, the hyperparameters of the CNN have been optimized using the grey wolf optimizer (GWO) algorithm. The data comprised of chest X-rays of normal and pneumonia-affected patients collected from publicly available repositories. There were 2700 images in all, of which 900 were Covid-19 images. The authors reported that the optimized CNN model outperforms the state-of-the-art models. The reported accuracy, sensitivity, specificity, precision and F1-score values are 97.78%, 97.75%, 96.25%, 92.88% and 95.25%, respectively.

The idea of mimicking brain cells is not a new domain. In fact neural networks themselves are a representation of neurons present in a human brain. But with the increasing energy demands and depleting energy resources followed by the advent of neuromorphic computing various new works have started evolving in the domain.

In the paper by [13], the authors have proposed a simple approach of spike response model (SRM) neuron with high computational efficiency. For representing data, they have used frequency spike coding based on receptive fields and encoded the images by the network. The method they have used for processing the images is equivalent to the mechanism followed by the primary layers in visual cortex. The output of the network has then been used for extracting primary features for refined classification. The authors have reported that the model has successfully learnt and classified greyscale images with added noise or partially ambiguous image samples at a 20x higher speed at an equivalent classification ratio as compared to a classic SRM neuron membrane model. Their solution is a combination of network topology, spike encoding and neuron membrane model.

<sup>1</sup> <https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset>.

**Fig. 1** Sample images from COVID-19 CT Scan dataset belonging to corresponding classes



## 2.1 Research gap

Deep learning-based approaches have become very popular in the recent years. It has been applied in several domains of image processing, pattern recognition and computer vision to achieve state-of-the-art results. In particular, the fields of medical image processing have seen many advances recently in various tasks that include image classification, image segmentation, image retrieval, computer-aided diagnosis, etc. Some works relating to the domain of COVID-19 have already been mentioned above. On the other hand, the applications of SNNs are a relatively new topic in research. As theoretical research is still ongoing, very few application-based works have been published. To the best of our knowledge, no SNN-based works have been proposed till date in the field of COVID-19 detection from chest CT scans or CXRs. Therefore, our work might be the first such work.

## 3 Dataset used

The COVID-CT dataset<sup>2</sup> contains 349 CT scan images of 216 patients (multiple images for same patient taken at different times) diagnosed positive for COVID-19 and 397 CT scan images that are diagnosed negative for COVID-19. The dataset is open-sourced to the public, to foster the research works of CT specific testing of COVID-19. From 760 medRxiv and bioRxiv preprints about COVID-19, the creators of the dataset extracted reported CT images and manually selected those containing clinical findings of COVID-19 by reading the captions of the selected images. The credibility and effectiveness of the dataset have been confirmed by a highly qualified senior radiologist who has intensively diagnosed and treated many COVID-19 patients. The personal data of the patients are anonymized to protect their privacy [28]; there is a metadata file that allows access to the data through descriptors such as

patient ID, patient information, DOI and image caption. Some sample images from the dataset are shown in Fig. 1.

## 4 Proposed methodology

In this section, the working principle of the proposed model has been described. The overall methodology is shown in Fig. 2.

### 4.1 Overall architecture

The work of the input layer is to encode the input image in the form of a Poisson-distributed spike. The pixel intensity has a directly proportional relationship with the probability of spike generation. This resultant encoding is passed on to the intermediate stages. The intermediate stages of feature hierarchies consist of hidden layers which are made up of convolution and spatial-pooling (C) layers stacked alternatively. These spikes are then concatenated sequentially and used as features to be fed to external classifiers for binary classification of the inputs. Apart from the C layers, all the other layers consist of trainable parameters. The local features which are having spatial correlation in the input patterns can be detected by the adapted convolutional kernels using convolution, which has an intrinsic property of rendering the network showing invariance to translation (shift) in the object location. Thereafter, downscaling of feature maps in terms of dimension which are produced by the previous layers is done by the P layer. In the whole process, retention of the spatial correlation between neighbourhood pixels in every feature map takes place.

### 4.2 Preprocessing

Every image is first resized to a dimension of  $32 \times 32$  and converted to greyscale. This image (as shown in Fig. 3) is then passed through the preprocessing pipeline as shown in Fig. 4 and the resultant encoded image is passed through the network for further processing.

<sup>2</sup> <https://github.com/UCSD-AI4H/COVID-CT>.



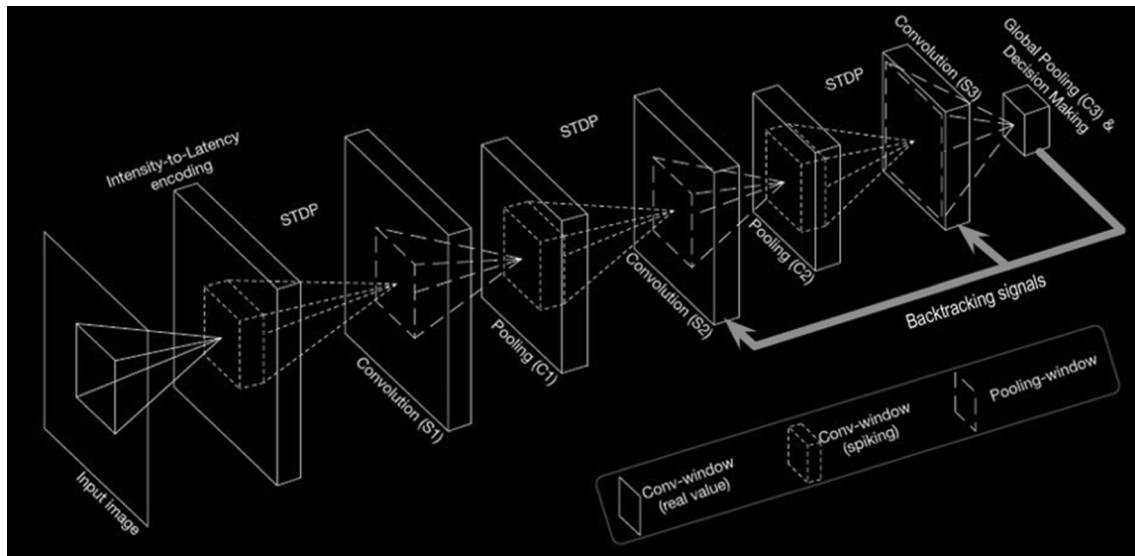


Fig. 2 Overall architecture used for feature extraction

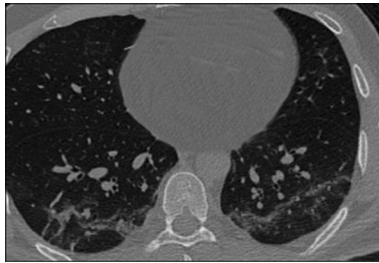


Fig. 3 Sample image before preprocessing

#### 4.2.1 Gabor filters

Gabor filters are a type of band-pass filters which prove to be quite useful in extracting useful features from images. Multiplication of a Gaussian envelope function with a complex oscillating function generates the impulsive response of these filters. [5] had shown in his work that these simple functions help in minimizing the space-time uncertainty product with an immediate implication of orientation selective behaviour of these functions on extension to two dimensions. We have used the filters for extracting various features from the images, both frequency based as well as orientation based features, which have helped the SNNs in better understanding and setting of membrane potentials and spiking thresholds, while filtering out any noise. From these images two features, namely, Gabor filter entropy and Gabor filter energy, are extracted using the real and imaginary components. These features correctly respond to the edges if the edge direction is perpendicular to the Gaussian kernel wave vector.

#### 4.2.2 Output of preprocessing stage

Figure 5 provides a visualization of the features after the preprocessing stage for a single input image. The time values in the bottom are a result of adding the time dimension. The different features are a result of applying the different Gabor filters. The summation of these features produces the final feature vector. The features and hence their summation are different at different time intervals. This is due to the intensity-to-latency encoding which is applied in the last stage of preprocessing.

#### 4.3 Feature extraction

Here we have explained in a detailed manner the feature extraction process of the architecture.

##### 4.3.1 Convolution

Every convolutional (S-layers) in the proposed network consists of many 2D grids of Integrate-and-Fire (IF) neurons, which is basically the representation of feature maps. Every neuron belonging to a layer has a definite three-dimensional input window with same height and width of afferent which is analogous to the units conveying information from sensory organs to central nervous system and depth which is same as the number of feature maps present in the previous layer [17]. The firing threshold is also initialized to be equal across all the neurons in every individual layer. In every time step, the value of internal potential for every IF neuron is enhanced by the incoming spikes within its input window by making use of the magnitude of the synaptic weights. There is absence of any

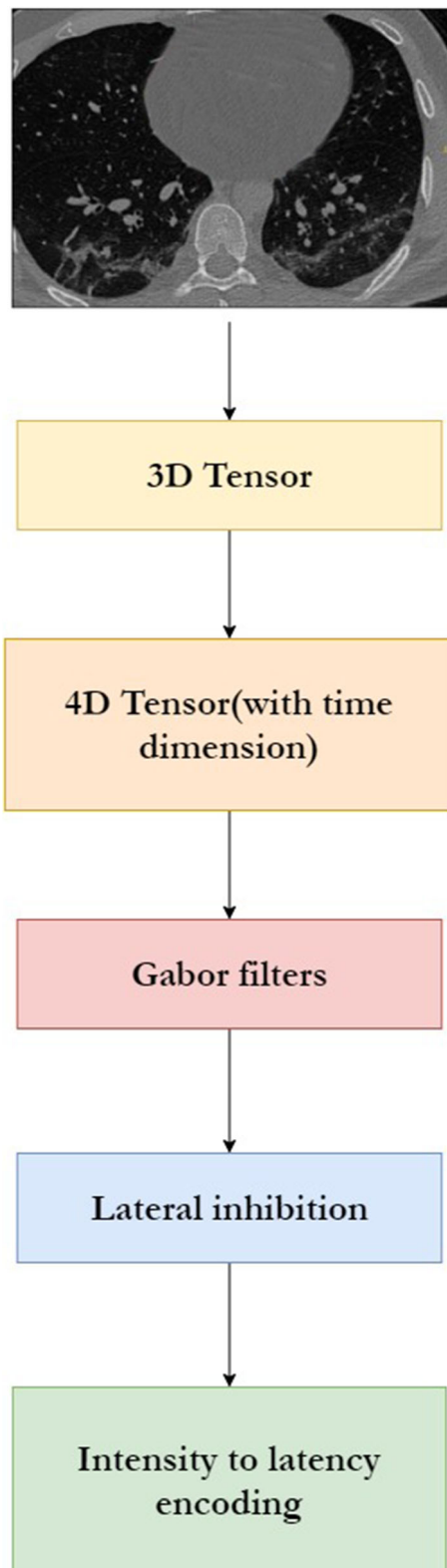


Fig. 4 Image preprocessing pipeline

leakage in these neurons. If any neuron attains the firing threshold, it emits a single spike, and thereafter, it remains

silent until the next input image is passed through the network.

Whenever the network encounters a new input, the internal potentials of all neurons are reset to zero. In a single feature map, a weight-sharing mechanism is applied to all the neurons. A convolutional layer with a kernel size  $\mathbb{B}_h \times \mathbb{B}_w$  performs a convolution over an input 4D tensor depicting a spike-wave of size  $T_{max} \times \mathbb{F}_{in} \times \mathbb{H}_{in} \times \mathbb{S}_{in}$  having value of stride equal to 1 and produces a tensor of output potentials having size  $T_{max} \times \mathbb{F}_{out} \times \mathbb{H}_{out} \times \mathbb{S}_{out}$ , where:

$$\mathbb{H}_{out} = \mathbb{H}_{in} - \mathbb{B}_h + 1, \mathbb{S}_{out} = \mathbb{S}_{in} - \mathbb{B}_h + 1 \quad (1)$$

where  $\mathbb{F}_{in}$  and  $\mathbb{F}_{out}$  represent the values corresponding to number of input and output features, respectively. Potential type tensors ( $\mathbb{P}$ ) are similar to the binary spike-wave tensors, but  $\mathbb{P}[t, f, r, c]$  represents the floating-point potential of any neuron placed at the position  $(r, c)$  of any feature map  $f$ , at time step  $t$  (see Table 1).

#### 4.3.2 Spike timing-dependent plasticity

Spike timing-dependent plasticity (STDP) is a proven technique for detecting hidden patterns from the noise present in spiking data. It belongs to the class of unsupervised learning techniques which works on the basis of the ordering of the synaptic spikes. The ordering of every pair of presynaptic spike and post-synaptic spikes as shown in Fig. 6 decides the potentiation (pre-post) or depression (post-pre) of the synapse. Changes of weights is based on the following two rules:

- Any synapse that contributes to the firing of a post-synaptic neuron should be made strong, that is, its value should be increased.
- Synapses that do not contribute to the firing of a post-synaptic neuron should be diminished, that is, its value should be decreased.

This method helps in learning the repetition of patterns among a large set of incoming spikes.

Since this simulation of SNN works with the coding mechanism of time to first spike, the implemented STDP function that has been used here is shown in Eq. 2:

$$\Delta \mathbb{S}_{ij} = \begin{cases} K^+ \times (\mathbb{S}_{ij} - L) \times (U - \mathbb{S}_{ij}) & \text{if } T_j - T_i \leq 0, \\ K^- \times (\mathbb{S}_{ij} - L) \times (U - \mathbb{S}_{ij}) & \text{if } T_j - T_i > 0 \end{cases} \quad (2)$$

where  $\mathbb{S}_{ij}$  refers to the corresponding change of value of weights of the synapse which connects the post-synaptic neuron  $i$  to the presynaptic neuron  $j$ ,  $K^+$  and  $K^-$  signify the learning rates, and  $(\mathbb{S}_{ij} - L) \times (U - \mathbb{S}_{ij})$  is the regularizer slowing down the change in weight when the synaptic weight  $(\mathbb{S}_{ij})$  is nearer to the lower( $L$ ) and upper( $U$ )

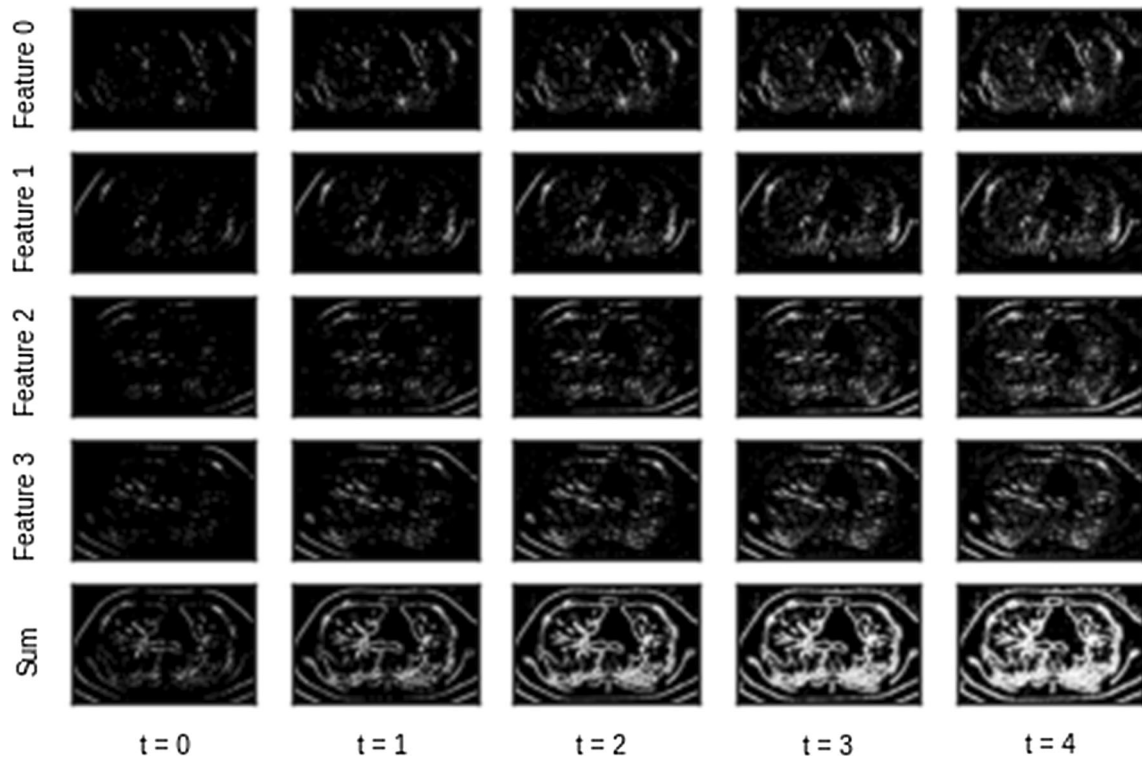


Fig. 5 Visualization of features after preprocessing

Table 1 Parameter configuration for S-layers for feature extraction [18]

Layer	Number of feature maps	Input window (width, height, depth)	Threshold
S1	30	(5, 5, 6)	15
S2	250	(3, 3, 30)	10
S3	200	(5, 5, 250)	$\infty$

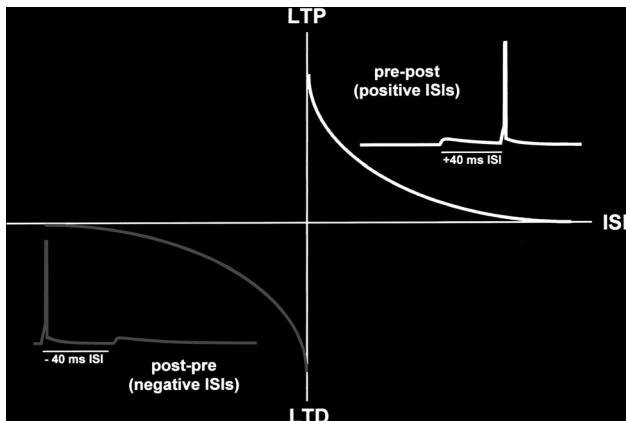


Fig. 6 Presynaptic and post-synaptic spike pairs

bounds.  $T_i$  and  $T_j$  depict the spike times of the presynaptic (input) and the post-synaptic (output) neuron, respectively (see Table 2).

Table 2 Parameter configuration for synaptic plasticity for feature extraction

Layer	$K^+$	$K^-$	k	r
S1	0.004	-0.003	5	3
S2	0.004	-0.003	8	2
S3	0.004	-0.003	1	0

For applying STDP in the process of training, provision of the input and output spike waves, as well as output potentials, are necessary in order to find the winners. Winners-take-all (WTA) is a well-known competition-based algorithm which is employed in SNNs. WTA is generally used for plasticity, but is extensible for use for performing other functions such as decision-making. Winners are decided first on basis of the earliest spike times, thereafter based on the maximum potentials. The number of winners is defaulted to take the value 1. The phenomenon of lateral inhibition is followed by means of which the winners' surrounding neurons present in all of

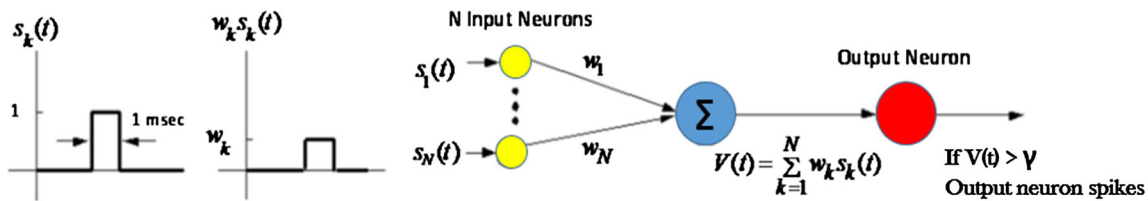


Fig. 7 A simple two-layer network with presynaptic( $s_i$ ) and post-synaptic(output) neurons

the feature maps within a specific distance are completely inhibited. This helps in enhancing probability of learning more diverse features.

Figure 7 [27] represents a simple two-layer network consisting of  $N$  presynaptic neurons (input) and 1 output neuron. The signals of spikes ( $s_i(t)$ ) are designed to be either 0 or 1 in one millisecond of increment. That is, 1 millisecond pulse of amplitude 1 depicts a spike and a value of 0 depicts absence of a spike. Every signal of spike contains a weight or synapse associated with it which gets multiplied with the signal to obtain  $w_i s_i(t)$  which is termed as the post-synaptic potential due to the  $i^{\text{th}}$  input neuron. These potentials are thereafter aggregated using the Eq. 3.

$$\mathcal{V}(t) = \sum_{k=1}^N w_k s_k(t) \quad (3)$$

where,  $\mathcal{V}(t)$  is termed as the membrane potential of the output neuron. At some instant of time  $t$ , if the membrane potential  $\mathcal{V}(t)$  exceeds a specified threshold, that is, if  $\mathcal{V}(t) > \gamma$ , then there is a spike in the output neuron.

#### 4.3.3 Pooling

Pooling layers (C-layers) have been used for the purpose of position invariance and reduction of information redundancy. Every C-layer or S-layer consists of equal number of feature maps as present in its previous layer, thus building a one-to-one relationship between the maps of both layers. For two variants of SNNs, which are spike train-based and potential-based, we have used two types of pooling layers, respectively. Both the variants consist of a 2D input window and a fixed stride. Every neuron in the layers gives indication of the maximum potential and the earliest spike time of the neurons within its input window for the potential based and spike train-based layers respectively. The value of stride is same as the window size by default, but can be customized accordingly.

According to the structure of the spike-wave and potential tensors, if a spike-wave tensor serves as the input, then the earliest spike within each pooling window is extracted, whereas if a potential-based tensor serves as the input, the maximum potential within each pooling window is extracted. The feature maps of the final pooling layer are flattened to give a one dimensional vector which is fed to

an external classifier whose work in turn is to produce inference decisions. The classifier effectively incorporates the composition of features which are the results from the alternating convolutional and pooling layers and classifies into the final output classes. Figure 8 shows an example of conversion of spike times into tensors of spike-wave (see Table 3).

#### 4.4 Classifier

Deep learning-based classifiers are the best when there is presence of a huge number of training instances. However, if a dataset contains lesser quantities of data, machine learning-based classifiers are best suited for the decision making stage [29]. In this work, at the last level of the whole working pipeline, we have used a Random Forest classifier [2] for classifying the output class of COVID or non-COVID on the basis of the features extracted in the upper levels of the working pipeline. A random forest classifier is an ensemble-based classifier that generates a series of decision tree classifiers (shown in Fig. 10) on various sub-samples of the dataset as shown in Fig. 9. It uses the principle of averaging to improve the accuracy of prediction and to control over-fitting. For our work, we have set the maximum depth hyper-parameter to 8, random-state to 32 and other parameters to their default values as initialized in the scikit-learn library [21].

#### 4.5 Comparison with deep learning models

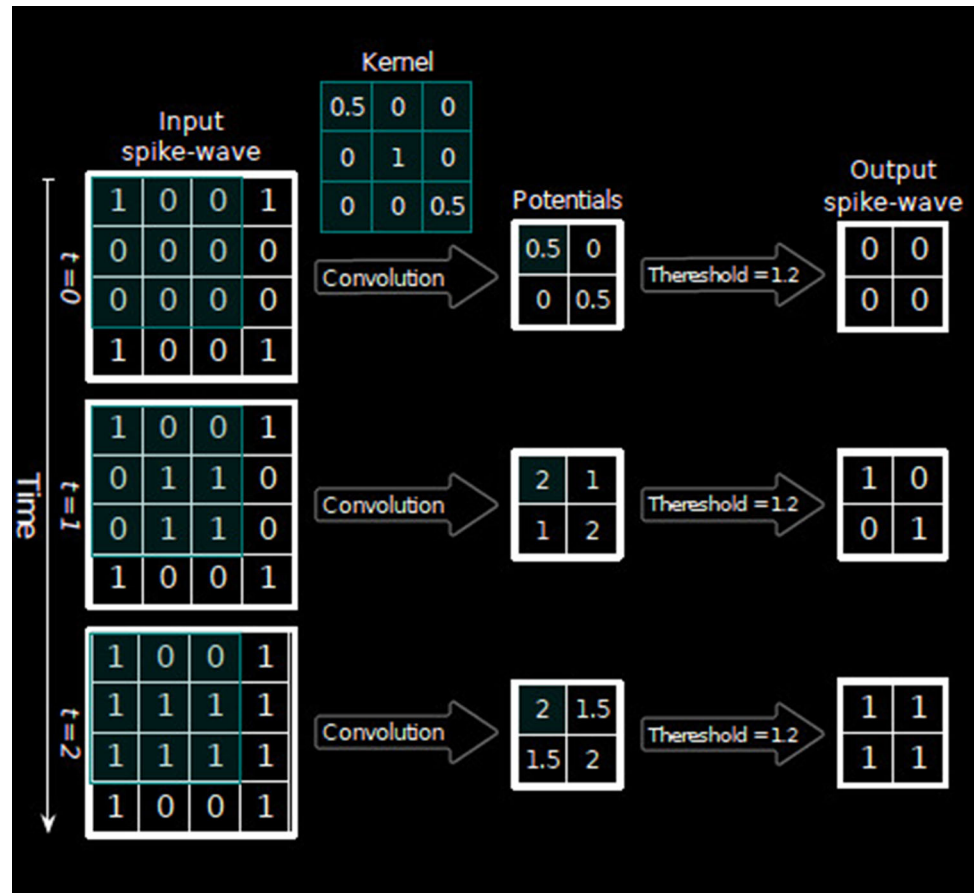
We also train some state-of-the-art deep learning-based classifiers to find the performance of these models on the metrics considered. We use these as baselines in order to judge the performance of the proposed approach against the prevalent deep learning based approaches. The models considered are: VGG16, ResNetV2 and Densenet161 adapted from the works by [, 10, 11, 24] respectively.

We use the transfer learning technique to train the models under consideration. A schematic diagram of the technique is highlighted in Fig. 11. Initially, the models are trained on the ImageNet dataset<sup>3</sup>. After convergence, the last dense layer (Head 1 in the figure) is replaced with a

<sup>3</sup> <http://www.image-net.org/>.



**Fig. 8** Simultaneous processing of spikes over time steps to convert to tensor containing potentials in all of the time steps



**Table 3** Parameter configuration for C-layers for feature extraction [18]

Layer	Input window (width,height)	Stride	Type
C1	(2,2)	2	Spike train-based
C2	(3,3)	3	Spike train-based
C3	(5,5)	0	Potential based

randomly initialized dense layer (Head 2 in the figure) with two outputs corresponding to the two classes. The model is then trained for 20 epochs with only the weights of the last layer being updated. After that, the entire model is fine-tuned for 20 epochs with a very low learning rate.

## 5 Experimental results

### 5.1 Metrics used

We have used various classification metrics [16] for measuring the effectiveness of our model in classifying COVID from the CT scan images, which are as follows:

- F1-score: Weighted average of precision and recall values.
- Accuracy: Overall probability of a patient being correctly classified.
- Sensitivity: Probability of positive test results when person is actually having the disease .
- Specificity: Probability of negative test results when person is actually not having the disease.
- Positive likelihood ratio: Ratios of the probabilities of a test result being positive provided the disease is present and a test result being positive given the disease is absent.
- Negative likelihood ratio: Ratios of the probabilities of a test result being negative provided the disease is present and a test result being negative given the disease is absent.
- Positive predictive value: Probability that provided the test is positive, the disease is present.
- Negative predictive value: Probability that provided the test is negative, the disease is absent.

Fig. 9 Overview of working of a random forest classifier

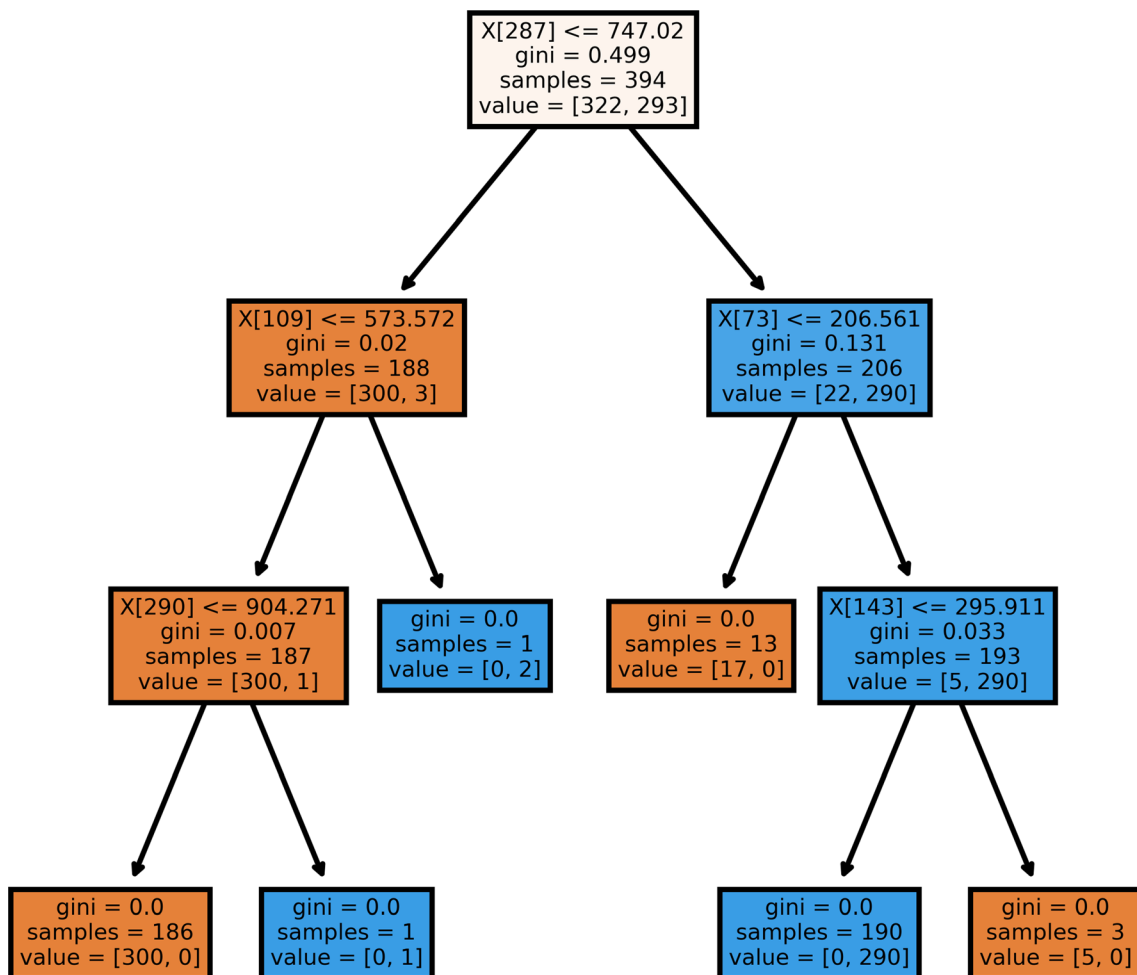
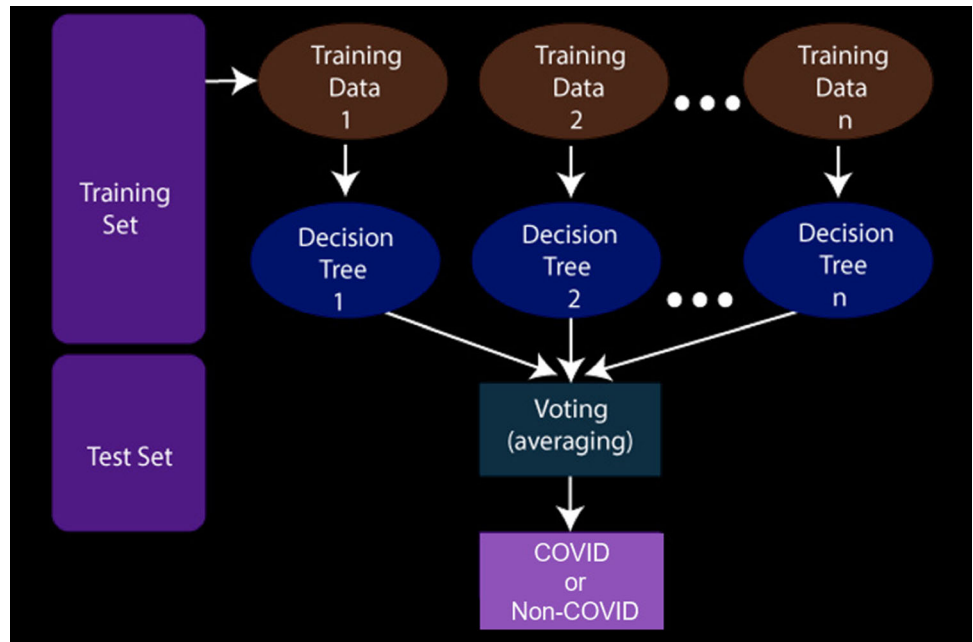
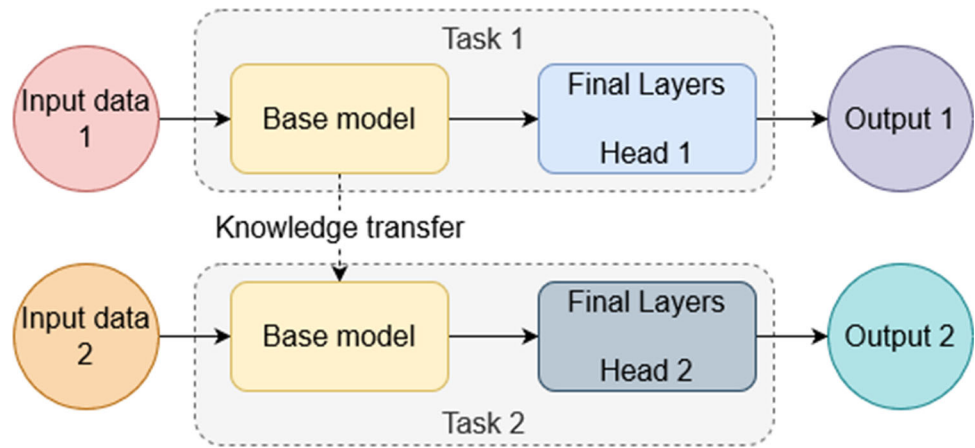


Fig. 10 Sample decision tree of an estimator of the random forest classifier

**Fig. 11** A schematic diagram of the transfer learning approach



**Table 4** Classification results of potential-based model

Class	Precision	Recall	F1-score	Support
COVID-19	1.00	0.99	0.99	95
Non-COVID	0.99	1.00	0.99	110
<b>Macro avg.</b>	1.00	0.99	0.99	205

**Table 5** Medical results [23] associated with the classification using potential-based model

Statistic	Value	95% CI
Sensitivity	98.96%	94.33% to 99.97%
Specificity	100.00%	96.67% to 100.00%
Negative likelihood ratio	0.01	0.00 to 0.07
Disease prevalence (*)	46.83%	39.84% to 53.91%
Positive predictive value (*)	100.00%	
Negative predictive value (*)	99.09%	93.94% to 99.87%
Accuracy (*)	99.51%	97.31% to 99.99%

**Table 6** Classification results of spike train-based model

Class	Precision	Recall	F1-score	Support
COVID-19	0.63	0.92	0.74	95
Non-COVID	0.88	0.54	0.66	110
<b>Macro avg.</b>	0.75	0.72	0.70	205

**Table 7** Medical results associated with the classification using spike train-based model

Statistic	Value	95% CI
Sensitivity	91.58%	84.08% to 96.29%
Specificity	53.64%	43.88% to 63.20%
Positive likelihood ratio	1.98	1.60 to 2.44
Negative likelihood ratio	0.16	0.08 to 0.31
Disease prevalence (*)	46.34%	39.37% to 53.42%
Positive predictive value (*)	63.04%	58.03% to 67.79%
Negative predictive value (*)	88.06%	78.79% to 93.61%
Accuracy (*)	71.22%	64.50% to 77.31%

## 5.2 Results and analysis

At the outset, it is to be noted that training the deep learning models from scratch results in overfitting on the training dataset. This leads to poor performance on the test dataset, with the accuracy being close to approximately 60% while the training accuracy is 100%.

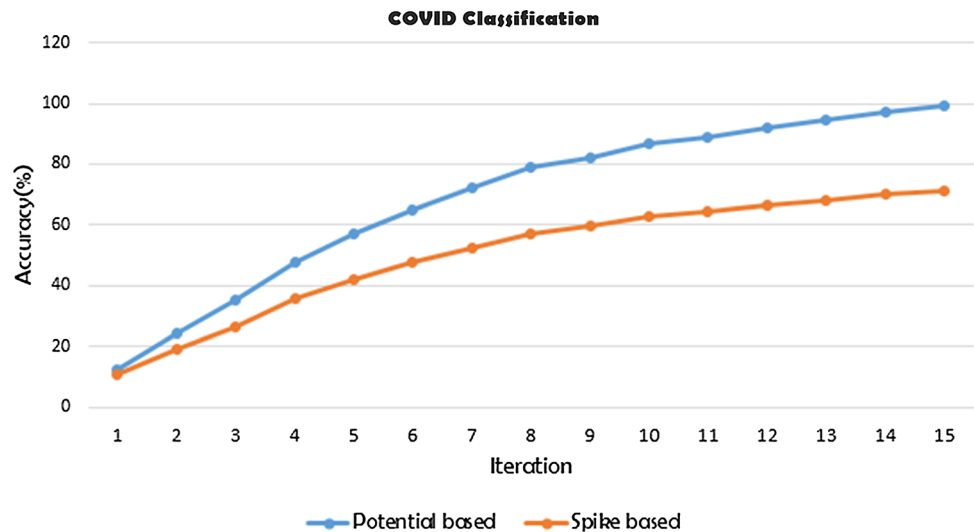
The potential-based classification achieves much better results compared to the spike train-based model as shown in Tables 4, 5, 6 and 7. This occurs due to the collective errors which keeps on accumulating due to improper firing of spikes resulting due to improper thresholding at the IF neurons while using spike train-based classification. This error accumulation is absent in case of potential based

model due to direct relationship of the potential of neurons with the decision making. This can be understood from the natural phenomenon itself which the SNNs mimic.

Action potential marks the beginning of the chain of events which lead to contraction for example in muscle cells, while the temporal sequence of these action potentials are called spike trains. So errors occurring in the action potentials propagate and keep on accumulating in the spike trains.

The learning in the network goes on quite smoothly as it can be seen from the iterative training accuracy graphs in Fig. 12. Both the potential-based and spike train-based

**Fig. 12** Iterative accuracy attained while training in potential and spike train-based feature classifiers



**Table 8** Classification results of the deep learning-based models

Model	Class	Precision	Recall	F1-score	Support
VGG16	COVID	0.83	0.88	0.85	86
	Non-COVID	0.89	0.84	0.87	101
	Macro avg.	0.86	0.86	0.86	187
ResNet101V2	COVID	0.83	0.92	0.87	86
	Non-COVID	0.92	0.84	0.88	101
	Macro avg.	0.88	0.88	0.88	187
DenseNet161	COVID	0.89	0.85	0.87	86
	Non-COVID	0.88	0.91	0.89	101
	Macro avg.	0.88	0.88	0.88	187

**Table 9** Accuracy of the models considered

Model	Accuracy
VGG16	0.86
ResNet101V2	0.88
DenseNet161	0.88
Spike train-based	0.71
Potential-based	1.00

**Table 10** Errors of the two SNN variants considered

Model	Error
Spike train-based	0.29269
Potential-based	0.00488

models start to learn with almost equal accuracy but with time the learning of the spike train-based model does not keep up with the learning capabilities of the potential-based model and hence gives worse results comparatively because of exactly the same reason as explained above.

Table 8 shows the precision, recall and F1-score of the VGG [24], ResNet [9] and DenseNet [12] models, respectively. The models perform similarly, with the VGG

having a F1 score about 0.02 lower than the other two models both of which achieve scores of 0.88. On comparison with the spike-based model (see Table 6), we find that the deep learning models have better performance by a large margin. However, with respect to the potential based model (see Table 4), we find that these models are lacking in their performance based on the metrics being considered.

Table 9 highlights the accuracies of all the models that have been considered in this work. The potential-based model outperforms all the other models considered here. The errors in the spike and potential based models are highlighted in Table 10. It can be noted that the potential-based model performs better than the deep learning models. It can be because the potential-based model learns fewer redundant features with respect to the other convolution-based deep learning models. Having fewer parameters as compared to the deep learning models, it also makes more efficient use of the parameters. However, one trade-off for this higher performance and efficiency would be the higher training time for the SNN based models.

From the above results, it can be seen that the potential-based SNN model performs the best. The spike-train-based

SNN does not provide competitive results as compared to the deep learning-based models. However, the potential-based SNN outperforms the state-of-the-art deep learning models by a significant margin. This demonstrates the usability of SNNs in real-world scenarios. Additionally, the benefits provided by the neuromorphic chips make SNNs a viable option for various practical applications. This is relevant since deep learning models generally require a graphics processing unit to obtain fast inference times. The energy-efficiency and computing power of the chips are more suited as processing elements. The advantages do come at a cost. It takes a large amount of time to train them, even more than the deep learning models which themselves take hours, or even days in some cases.

## 6 Conclusion

In this work, we have designed a three-layer DCSNN for screening of the COVID-19 from CT scan images. In doing so, the input image is first convolved and processed with Gabor filters. Then, by means of an intensity-to-latency encoding, a wave of spikes is generated. After propagating through a series of convolutional and pooling layers, with neurons having the ability of firing at most once, the spike wave reaches the penultimate layer. Finally, the useful features are extracted that are then fed to a classifier for making the final decision.

On evaluation on the COVID-CT dataset, the proposed approach has achieved an impressive F1 score of 0.99 for the potential-based model. The approach also outperforms some state-of-the-art COVID-19 classification models. Although the proposed SNN-based model performs very well on chest CT images, there is a limitation of this model. It takes more time to train the model in comparison with the traditionally used deep learning models. However, the present model is more efficient compared to these deep learning models. In future, we plan to come up with an idea which will help us to cut down the training time. Another plan is to apply the model to other forms of COVID-19 datasets like chest X-rays which will prove the robustness of the model.

**Acknowledgements** The authors gratefully acknowledge financial support from ANID PIA/APOYO AFB180003.

**Funding** Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Bi GQ, Poo MM (1998) Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J Neurosci* 18(24):10464–10472
2. Breiman L (2001) Random Forests. *Mach Learn* 45(1):5–32
3. Davies M, Srinivasa N, Lin TH, Chinya G, Cao Y, Choday SH, Dimou G, Joshi P, Imam N, Jain S et al (2018) Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38(1):82–99
4. Furber S (2016) Large-scale neuromorphic computing systems. *J Neural Eng* 13(5):051001
5. Gabor D (1946) Theory of communication. part 1: The analysis of information. *J Inst Electr Eng-Part III: Radio Commun Eng* 93(26):429–441
6. Gautrais J, Thorpe S (1998) Rate coding versus temporal order coding: a theoretical approach. *Biosystems* 48(1–3):57–65
7. Gerstner W, Kempter R, Van Hemmen JL, Wagner H (1996) A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383(6595):76–78
8. Goel T, Murugan R, Mirjalili S, Chakrabarty DK (2021) Optconet: an optimized convolutional neural network for an automatic diagnosis of covid-19. *Appl Intell* 51:1351–1366. <https://doi.org/10.1007/s10489-020-01904-z>
9. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
10. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision, Springer, 630–645
11. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 4700–4708
12. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2018) Densely connected convolutional networks. 1608.06993
13. Iakymchuk T, Rosado-Muñoz A, Guerrero-Martínez JF, Bataller-Mompeán M (2015) Francés-Víllora JV (2015) Simplified spiking neural network architecture and stdp learning algorithm applied to image classification. *EURASIP J Image Video Process* 1:4
14. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M (2020) Classification of the covid-19 infected patients using densenet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics* 1–8
15. Kheradpisheh SR, Ganjtabesh M, Thorpe SJ, Masquelier T (2018) Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Netw* 99:56–67



16. Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. *Biometrics* 33(1):159–174
17. Mozafari M, Ganjtabesh M, Nowzari-Dalini A, Masquelier T (2019) Spyketch: Efficient simulation of convolutional spiking neural networks with at most one spike per neuron. *Frontiers in Neuroscience* 13:625 <https://doi.org/10.3389/fnins.2019.00625>, <https://www.frontiersin.org/article/10.3389/fnins.2019.00625>
18. Mozafari M, Ganjtabesh M, Nowzari-Dalini A, Thorpe SJ, Masquelier T (2019) Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks. *Pattern Recogn* 94:87–95
19. Ouyang X, Huo J, Xia L, Shan F, Liu J, Mo Z, Yan F, Ding Z, Yang Q, Song B, Shi F, Yuan H, Wei Y, Cao X, Gao Y, Wu D, Wang Q, Shen D (2020) Dual-sampling attention network for diagnosis of covid-19 from community acquired pneumonia. *IEEE Trans Med Imaging* 39(8):2595–2605. <https://doi.org/10.1109/TMI.2020.2995508>
20. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) *Advances in Neural Information Processing Systems* 32, Curran Associates, Inc., 8024–8035
21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830
22. Rajaraman S, Siegelman J, Alderson PO, Folio LS, Folio LR, Antani SK (2020) Iteratively pruned deep learning ensembles for covid-19 detection in chest x-rays. *IEEE Access* 8:115041–115050. <https://doi.org/10.1109/ACCESS.2020.3003810>
23. Schoonjans F (2020) Medcalc statistical software. <https://www.medcalc.org/>, Accessed: 03/07/2020
24. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
25. Singh D, Kumar V, Kaur M (2020) Classification of covid-19 patients from chest ct images using multi-objective differential evolution-based convolutional neural networks. *Eur J Clin Microbiol Infect Dis* 39(7):1379–1389
26. Sun L, Mo Z, Yan F, Xia L, Shan F, Ding Z, Song B, Gao W, Shao W, Shi F, Yuan H, Jiang H, Wu D, Wei Y, Gao Y, Sui H, Zhang D, Shen D (2020) Adaptive feature selection guided deep forest for covid-19 classification with chest ct. *IEEE J Biomed Health Inform* 24(10):2798–2805. <https://doi.org/10.1109/JBHI.2020.3019505>
27. Vaila R, Chiasson J, Saxena V (2019) Deep convolutional spiking neural networks for image classification. arXiv preprint arXiv:1903.12272
28. Velásquez JD (2013) Web mining and privacy concerns: Some important legal issues to be consider before applying any data and information extraction technique in web-based environments. *Expert Syst Appl* 40(13):5228–5239
29. Velásquez JD, Jain LC (2010) *Advanced techniques in web intelligence*, vol 311. Springer, Berlin
30. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection. *IEEE Access* 8:91916–91923
31. Yousefzadeh A, Masquelier T, Serrano-Gotarredona T, Linares-Barranco B (2017) Hardware implementation of convolutional stdp for on-line visual feature learning. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, pp 1–4
32. Zhao J, Zhang Y, He X, Xie P (2020) Covid-ct-dataset: a ct scan dataset about covid-19. arXiv preprint arXiv:2003.13865

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.