Neural Computing and Applications (2021) 33:6147–6159
https://doi.org/10.1007/s00521-020-05388-3

ORIGINAL ARTICLE

# Detection of weather images by using spiking neural networks of deep learning models

Mesut Toğaçar[1] · Burhan Ergen[2] · Zafer Cömert[3]

Received: 29 April 2020 / Accepted: 24 September 2020 / Published online: 8 October 2020

## Abstract
The transmission of weather information of a location at certain time intervals affects the living conditions of the people there directly or indirectly. According to weather information, people shape their behavior in daily life. Besides, agricultural activities are carried out according to the weather conditions. Considering the importance of this subject, it is possible to make weather predictions based on the weather images in today's technology exploiting the computer systems. However, the recent mention of the name of artificial intelligence technology in every field has made it compulsory for computer systems to benefit from this technology. The dataset used in the study has four classes: cloudy, rain, shine, and sunrise. In the study, GoogLeNet and VGG-16 models and the spiking neural network (SNN) were used together. The features extracted from GoogLeNet and VGG-16 models were combined and given to the SNNs as the input. As a result, the SNNs contributed to the success of classification with the proposed approach. The classification accuracy rates of cloudy, rain, shine, and sunrise classes were 98.48%, 97.58%, 97%, and 98.48%, respectively, together with SNN. Also, the use of SNNs in combination with deep learning models to obtain a successful result is proved in this study.

**Keywords** Spiking neural network · Deep networks · Weather images · Feature extraction and combination

## 1 Introduction

Weather information given over a certain time is important for people. People shape their daily lives and behaviors directly or indirectly according to weather conditions. For example, cycling, traveling by plane, going on vacation, etc. Besides, business plans, driving systems, sports activities, and sightseeing tours, such as the weather of the places where events are taken into account [1].

✉ Mesut Toğaçar
mtogacar@firat.edu.tr

Burhan Ergen
bergen@firat.edu.tr

Zafer Cömert
zcomert@samsun.edu.tr

1 Department of Computer Technology, Technical Sciences Vocational School, Fırat University, Elazig, Turkey

2 Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey

3 Department of Software Engineering, Faculty of Engineering, Samsun University, Samsun, Turkey

The weather is local to a region and is usually detected by sensors or by human observations. The cost of sensors with cameras has a negative impact on the economy of the region. Nowadays, it is foreseen that artificial intelligence technology will take place in embedded systems and analysis will be performed more accurately and hardware costs will decrease [1, 2]. Recently, the concept of artificial intelligence has started to take place in people's lives and has facilitated their lives. Today, large global corporations have used artificial intelligence applications in their technologies and continue to develop in this field. Deep learning architectures are a sub-branch of artificial intelligence, and these architectures contain hidden layers in their structure and automatically extract features in images [3].

In the literature review, many studies have been carried out classifying weather images using deep learning architectures. If we examine some of these studies, Zhao et al. [4] classified the five-class weather images with CNN models in their study. The proposed approach in their study was to classify each image in the dataset by assigning multiple tags. They processed each multi-tagged image with CNN and recurrent neural network (RNN) models.

They also used the long short-term memory (LSTM) method to model the relationships between different image tags. The most successful classification rate among different datasets was 92.63%. Lu et al. [5] classified the aerial images into two types (cloudy and sunny). In addition, they performed cloudy and sunshine percentages on each image in their studies. They obtained features such as shadow, reflection contrast, and haze on the images. For this, they used the directional color histogram technique. They trained the dataset with the proposed CNN method. The classification success achieved with support vector machines (SVM) using the data augmentation technique was 98.6%. Elhoseiny et al. [1] divided the weather images into two classes. In their work, they performed the classification process using the features obtained from the fully connected (FC7 and FC8) layers of the AlexNet architecture, one of the CNN models. They achieved 91.1% accuracy success by classifying the features extracted from the last layer with Softmax function. An et al. [6] used two different datasets in their study. The datasets were two classes and four classes, respectively. They used ResNet and AlexNet models in their studies. They preferred the multi-class SVM method as a classifier. The best accuracy success in the two-class dataset was 90% with the ResNet model. Their average accuracy rate in the four-class dataset was 97%. Villarreal Guerra et al. [7] used three classes of weather data in their study. They applied to the dataset the superpixel method and augmentation technique. Thus, they increased the pixel seeds homogeneously in each image. They used 25, 50, 75, and 100 seeds for each pixel, respectively. In their study, the success of the 100-seeded superpixel method was found to increase more. They were then classified using the SVM method by training the dataset with multiple CNN models. They achieved the best overall accuracy rate of 80.7% with the ResNet-50 model.

In this study, the classification of weather images was performed, and the use of convolutional neural network (CNN) models and spiking neural networks (SNNs) together has been proposed as a novel approach. The CNN models used in the study were GoogLeNet and VGG-16. SNNs contributed to the success of CNN models used in this study.

The sections of the study are as follows: information on materials and models is given in Sect. 2, and additionally, the proposed approach is given in this section. Information and analysis results about the experiment are given in Sect. 3. Discussion and conclusion are given in Sects. 4 and 5.

## 2 Materials and methods

### 2.1 Dataset

The dataset is a set of data contained in the research paper entitled "Multi-class weather recognition from a heterogeneous ensemble method" still. It was created by researchers from the University of South Africa from websites such as Google, Flickr, Getty, and Yahoo. It is aimed to recognize different weather conditions in the creation of the dataset. The image set is divided into four categories. These are cloudy, rain, shine, and sunrise. The number of datasets of the four classes, respectively, is 300, 215, 253, and 357. The dataset consists of a total of 1125 images. The resolution of each image is variable, the file format is JPG, and the image depth is 24 bits [8]. Example images of the four classes are shown in Fig. 1.

In all of the steps performed in the experiment of this study, 30% of the dataset was divided into test data, and 70% of the data as training data.

### 2.2 CNN models

Convolutional models are deep networks that contain hidden layers, automatically extracting the features of input data [9, 10]. CNNs give importance to weight updates in their architectural structures, and the ability to distinguish the features from each other depends on the weight parameters. Optimization methods used in CNN models and metric values used in the architecture play an important role [11, 12]. To mention the models used in this study, the GoogLeNet was developed by Google and it was the winner of the 2014 competition with its successful performance in the ImageNet. GoogLeNet showed an error rate of 6.7% for the accuracy of the Top-5 classification. This was a better result than the Top-5 error rate of the VGG-16 [13]. The GoogLeNet architecture is a convolutional model with approximately 5 million parameters. Since the building consists of blocks, it has low memory usage and gives fast results. The input size of this architecture is 224 $\times$ 224. Another model used in this study is the VGG-16, it is an architecture proposed by Oxford University researchers. The VGG-16 model achieved 92.7% success for the Top-5 class in the ImageNet classification competition held in 2014. This model minimizes the large filter dimensions (5 $\times$ 5, 11 $\times$ 11, etc.) used in the construction of models such as AlexNet, which allows the model to be improved [14, 15]. The GoogLeNet and VGG-16 models include convolutional layers, pooling layers, and fully connected (FC) layers. Also, GoogLeNet architecture includes building blocks in its structure. In the two models, the last layer has the Softmax activation

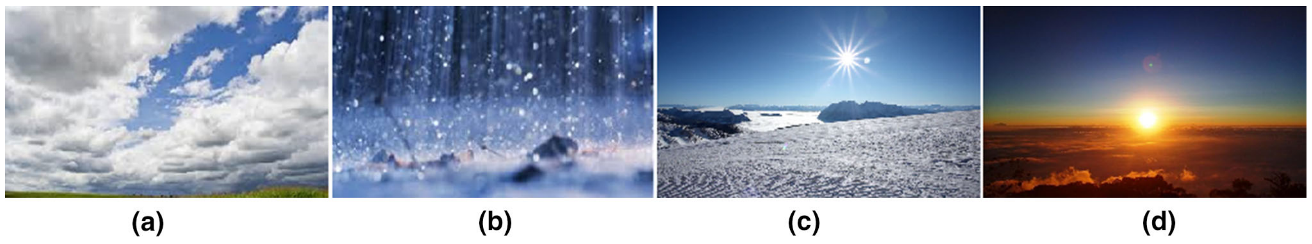

**Fig. 1** Sample images representing classes of weather dataset: **a** cloudy, **b** rain, **c** shine, and **d** sunrise

function [16]. The Softmax function specifies the value probabilities of features from the previous layer to be used in the classification process and allows normalization of nonlinear features. Thus, the classification step is performed with the probability values that the features have [17, 18]. The schematic design of convolutional networks is shown in Fig. 2. In Fig. 2, the convolutional layer transfers the window containing the new features to the one next layer with the filters it moves over the image window [19]. The pooling layer reduces its size so that it does not adversely affect image features and thus reduces the cost of the model [20]. The FC layers are used to optimize the probability scores before the classification process, and they are generally used toward the final layers of the CNN models [21].

The pre-trained CNNs are models that have been trained using large databases consisting of over one million samples to distinguish 1000 different objects [22–26]. This process is very costly and time-consuming. Most of the time, these deep models with high generalization performance can be used on new problems by using a transfer learning approach instead of training from scratch. In the transfer learning approach, the weights of the first layers of the models are not changed too much, the last layers of the models are arranged according to the new problem, and a light training called transfer learning is performed. In this study, pre-trained CNN models were used and the preferred parameters for these models were used with their default values. In this study, CNN architectures were trained by using the transfer learning approach. The 1000 features extracted from the Loss3, FC-8 layers of GoogLeNet, and VGG-16 models, respectively, were used for classification. Other important parameters and related values of the CNN models used in the study are given in Table 1. The mini-batch value allows to process multiple data images in parallel in the model and reduces the cost of the model. However, the mini-batch parameter is directly related to the hardware units where the operation takes place. In this study, since the hardware features were not sufficient, we took the mini-batch value as 32 [27, 28].

In this study, many classification methods such as Sofmax, kNN, and discriminant analysis have been tried. However, since the most successful results were given by the SVM method, this study was preferred in the experimental analysis. The SVM method, which has a strong potential in the solution of data analysis problems encountered in daily life, was preferred. In addition, this method was preferred because it is widely used in the solution of remote pattern recognition and classification problems and successfully executes multiple classification processes [29, 30]. Also, because of the best performance among SVM types (cubic, linear, quadratic, etc.), Cubic SVM was preferred in this study. Preferred parameter values in the Cubic SVM method: the kernel scale parameter was automatically selected, and the box constraint level parameter value was chosen one, and the

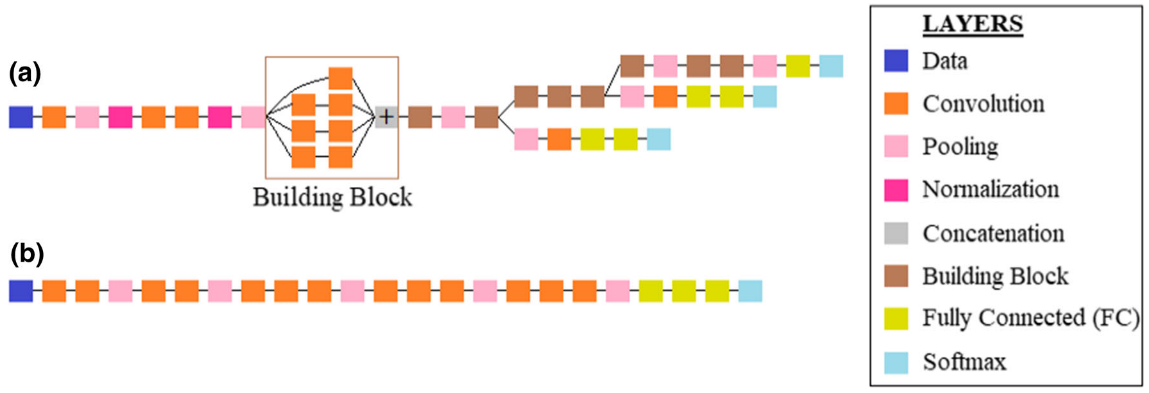

**Fig. 2** Schematic representation of convolutional networks: **a** GoogLeNet and **b** VGG-16

**Table 1** The important parameter values for the GoogLeNet and VGG-16 models used in this study

| Software | CNNs | Filter size | Input size | Optimization | Momentum | Decay | Mini batch | Learning rate | Classifier |
|---|---|---|---|---|---|---|---|---|---|
| MATLAB | GoogLeNet | $5 \times 5$ | $224 \times 224$ | Stochastic gradient descent (SGD) | $9 \times 10^{-1}$ | 1e–6 | 32 | $10^{-4}$ | SVM |
| | VGG-16 | $3 \times 3$ | | | | | | | |

multi-class method parameter one-vs-one was selected. In addition, there were three reasons for using GoogLeNet and VGG-16 models in this study: these models have made their name known in the ImageNet competition and are successful models; the second reason, input sizes were the same for both models; the third reason is that the two models have FC layers that give 1000 features.

## 2.3 Optimization method and machine learning method

Optimization methods aim to facilitate the learning of the model in deep learning architectures. It updates the weight and loss functions of the model [31]. The SGD optimization method updates the weight and cost parameter values in each cyclic round. In this way, with the model optimization method, a faster and more accurate classification process is realized. The mathematical formula that performs the weight update of the SGD method is given in Eq. (1). When the equation is examined; $\Theta$ represents the weight vector, $\alpha$ is the learning coefficient and $\nabla_\Theta J(\Theta)$ is the cost function. In addition, $t$ represents the number of times or steps in the equation, $(t-1)$ refers to the previous step. $x$ and $y$ represent the position value of the features in a two-dimensional plane [32].

$$\Theta_t = \Theta_{t-1} - \alpha \nabla_\Theta J\left(\Theta; x^i, y^i\right) \tag{1}$$

The SVM is a method used by deep learning models in dual/multiple classification and regression processes. This method uses a boundary line to divide the features placed

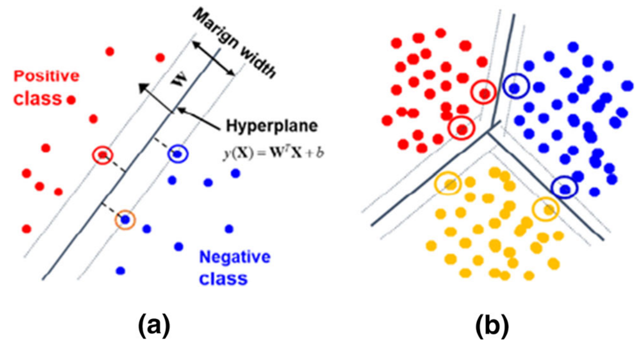

**Fig. 3** Classification scheme of SVM method: **a** two-class classification by SVM and **b** multi-class classification by SVM [35]

on the hyper-plane. When examined in Fig. 3, the marginal width gap is created by the optimization method. The process step which minimizes the problems that occur during the classification process is realized with Eq. (2). With Eq. (3), the classification step is performed. In this equation, $x$ and $y$ parameters represent the features and $i$ represents the number of loop steps. Also, $w$ is the weight parameter and $b$ is the boundary area [33, 34].

$$u = \vec{w} \cdot \vec{x} - b \tag{2}$$

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \quad \forall i \tag{3}$$

In this study, we used a multi-class cubic SVM method for CNNs (Fig. 3b). A graphical representation of a three-class SVM method is given, and each color represents a class. Multiple SVM methods match the number of colors (assuming $N$ is the parameter that represents the color); the number of matches is calculated by the formula $N \times (N-1)/2$. For example, in Fig. 3b, there are three colors. To calculate according to the formula, $3 \times (3-1)/2 = 3$. There are three matches (blue-yellow; blue-red; red-yellow). Each mapping is performed with the logic of binary classification, where mapping operations are performed for each feature. As a result of matching, the feature is passed to the class that receives the highest rating [36, 37].

## 2.4 Spiking neural networks (SNNs)

The SNN architecture is a network model that combines the neuronal and synaptic states with the concept of time and examines the natural nerve structures one by one. The fact that neurons in the SNN structures are not triggered in the propagation cycles allows those neurons to increase their self-quality to a certain level. When a neuron is triggered, it creates a condition that increases or decreases the signal potential of other neurons around it [38]. It uses a certain threshold to perform the activation of neurons. If the triggering value from the neurons exceeds this threshold, the triggering event occurs, otherwise, the decay occurs for the neuron. Triggered lines coming out or descending with certain periods, and their spacing represents the actual number of values. Interpretation of these numbers is done by coding and decoding methods used in the structure of the SNN model. Spiking neurons take into account every transient signal occurring within the model but do not immediately process it. For the signal to be processed, a certain threshold must be exceeded, the spiking neurons waiting for it, and process the signal as soon as the threshold is exceeded. And as a result, it produces an increasing or decreasing signal output. Increasing or decreasing signals occurring in the SNNs form coding schemes. Both the pulse frequency and the pulse range of the signals are taken into consideration and the codes are interpreted as corresponding to a sequence of numbers (decode) [39]. The pulse states for three neurons spiking over the time interval are shown in Fig. 4. For the SNN model, the Leaky integral-and-fire (LIF) method is generally used to calculate sudden rising and falling pulses in the time interval. The integral formula used for the LIF method is shown in Eq. (4) [40]. In this equation, $I$: current, $V$: volt, $C$: capacitor, $R$: resistance, and $t$ represent the time value.

$$I(t) - \frac{V_m(t)}{R_m} = C_m \frac{\partial Vm(t)}{\partial t} \tag{4}$$

Although the SNN model recalls the artificial neural networks (ANN) model, they are complex structures that do not produce a continuous output like them. On the contrary, it produces a binary output, which is not easily interpreted. Ultimately, it increases the spatial learning of spiking neurons, indicating that each neuron is only interested in nearby neurons. This shows that the CNN models are similar to the operation performed by the filter windows. In addition, since spiking neurons use the concept of time for each pulse in the process, it prevents additional complexity, unlike RNN models. This means that Spiking networks have an advantage over RNN models. The SNN model costs more than the ANN model. Because SNNs interpret neural states as simulations [40], some firms perform the smooth simulation of complex neural networks of SNN models. These are NEST, BindsNet, Brian, and GENESIS [41]. The SNN architecture consists of deep networks and contains hidden layers in its structure. The SNN model generally consists of three parts. These are the input layer, hidden layers, and the output layer. This is shown in Fig. 5. Here, the input layer preprocesses the image it takes as input and extracts the features. The

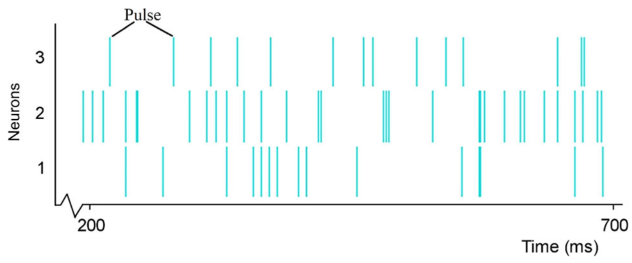

**Fig. 4** Spatiotemporal graph during the training of three neuron networks spiking [40]

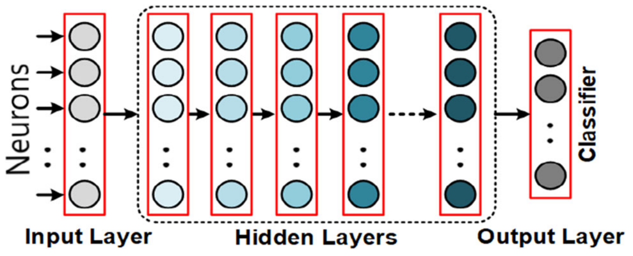

**Fig. 5** General steps of the SNN model: input layer, hidden layers, and output layer [42]

feature set is then transferred to hidden layers. As features spread through hidden layers, they enter a more complex and challenging training process, and the SNN model accomplishes learning. The hidden layers of these networks are also used in the backpropagation method, which facilitates the training process and performs weight updates. The last layer consists of fully connected layers in which the classification process takes place. The Softmax function is usually used for classification. The trained features use hot coding with the Softmax function. With hot coding, each feature corresponds to one of the classes. This is achieved by transferring probability values to the features transferred to the output layer with the Softmax function. The sum of the probabilities is equal to 1. The mathematical formula of the Softmax function is shown in Eq. (5). Herein $y_i$ represents the probability value of the output label. $a_i$ represents input tags. $i$ and $j$ indexes represent the set of input/output units [42]. In addition, the biological functioning of the SNN is shown in Fig. 6a and the representation by deep learning logic is shown in Fig. 6b.

$$y_i = \exp(a_i) \sum_j \exp(a_j) \tag{5}$$

SNNs work with spiking neural, discrete events occurring in that interval, drawing attention to time intervals rather than generating values such as conventional neural networks [43]. SNNs are considered to be the third generation neural network, and there are several algorithms other than Softmax, which serves to classify in the exit layer of fired neurons within this network. The algorithm used in this study was synaptic weight association training (SWAT). The spike-timed plasticity (STDP) supervised learning method performs time-based connection settings of the input and output neurons of the SNN model. SWAT prefers a learning method based on the postsynaptic voltage function to guide training. In addition, SWAT uses the STDP method to facilitate and converge learning. Methods including learning windows such as STDP and SWAT produce more efficient results than gradient descents. Synapses are the connection points that allow the transmission of messages between neurons. Synapses are called

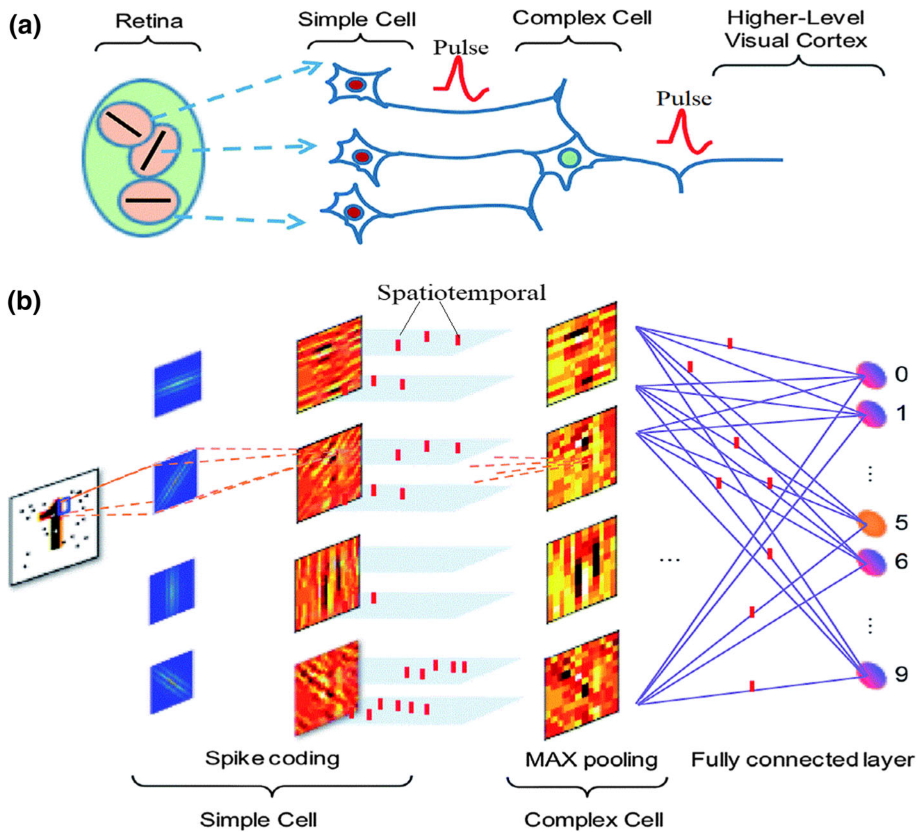

**Fig. 6** The SNN model: **a** design demonstrating the biological working principle and **b** neural networks—design with the principle of deep networks [43]

the presynaptic tip if it is between the neurons and the postsynaptic tip after the neurons. The sum of the voltage values delivered to presynaptic neurons is combined with postsynaptic neurons and provides a sudden increase as soon as the total voltage exceeds the specified threshold [44, 45].

The SNNs appear as a sudden response pattern and each neuron is fired only once during its period. The firing rate distribution is calculated according to Eq. (6). In this equation, $x$ is the input layer, $y = f(x)$ is the output layer, and $N$ is the number of neurons. $R_{\max}$ is the firing rate and the number of neurons adjacent to the neurons in the $XX$ coordinate. $\delta$ represents a constant value. The delay time is important here. Each delay time in SNNs is defined as the difference between the time that the presynaptic is fired and the time after the postsynaptic begin to rise. In SNNs, the learning rate is the process of changing the delay between the difference between the firing time of the presynaptic and the postsynaptic [45, 46]. The important parameter values of the SNN model used in this study are given in Table 2. The model design of the SNNs used in this study is shown in Fig. 7. The parameter values given in Table 2 are the default values used for SNN. Default values were used for this study.

$$f_x(x') = R_{\max} e^{\frac{\cos\left(\frac{2\pi}{N}(XX)\right)}{\delta^2}} \tag{6}$$

## 2.5 Proposed approach

The contributions of the proposed approach to the literature are as follows: it has enabled a successful use of deep learning models with SNNs. There are a limited number of articles in the literature merged with CNN model layers of SNNs. Among them, Cao et al. [47] and Xu et al. [48] have created a combination of CNN models and SNNs in their studies. However, the proposed CNN models in their studies consist of traditional layers and they used datasets with low input resolution (MNIST, CIFAR-10) in their studies. Also, any dataset with high input resolutions can be used in our study. Our approach offers the possibility to use all CNN models (AlexNet, GoogLeNet, DarkNet, ResNet, LeNet, designed deep learning models, etc.) and SNNs together. Our common aspect with these studies is that we can provide efficient features to SNNs by CNN models.

The proposed approach allows us to perform the classification process by giving the features extracted from CNN models to SNN. The GoogLeNet and VGG-16 models used in the study were pre-trained CNNs [49]. These CNNs have fully connected layers that can give 1000 features in both models. This allows you to combine datasets with the same number of features. Therefore, it is

**Table 2** Parameters and values of the SNN model used in the proposed approach

| Parameter | Value |
|---|---|
| Number of receptive field neuron in the population encoding scheme | 8 |
| Presynaptic spike interval in the millisecond | 3 |
| Postsynaptic spike interval in the millisecond | 4 |
| Desired postsynaptic firing time in millisecond | 2 |
| Precision of time-step | 0.1 |
| Learning rate of weight update | 0.50 |
| Sigma of time-varying weight kernel in the millisecond | 0.55 |
| The time constant of spike response function in the millisecond | 3 |
| Number of maximum epochs | 100 |
| Time constant of STDP learning window | 1.7 |

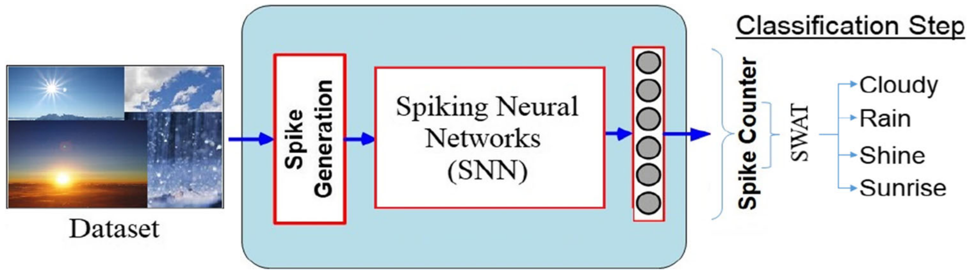

**Fig. 7** The main design of this study showing the functioning of the SNN model

aimed to give the feature set (∗.mat file) extracted from GoogLeNet and VGG-16 models as input to the SNN model. The proposed approach consists of two steps. In the first step, datasets are applied as the input to GoogLeNet and VGG-16 models, and feature sets are extracted separately. Then, the features trained with the SNN model have realized classification by the SWAT method in the last layer of SNNs. In the second step, a new feature set with 2000 features is created by combining two separate 1000 features extracted from the existing CNN models. In this way, more efficient features contribute to the training of the models and aim to increase the success of the classification process. In two steps of the proposed approach, SNNs contributed to the success of the classification of CNN networks in this study. The process steps and overall design of the proposed approach are shown in Fig. 8. In addition, the SNNs used in the proposed approach were designed with the source codes used in the study Jeyasothy et al. [45] called SEFRON.

## 3 Experiment analysis results

In this study, the confusion matrix and the metrics (sensitivity, specificity, F-score, and accuracy) derived from it were used to measure the success of the proposed model. The metric values are calculated according to the following equations. Here, TP is true-positive, FP is false-positive, TN is true-negative, and FN is false-negative [50–52].

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{8}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{9}$$

$$\text{F - Score} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{10}$$

$$\text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{11}$$

The pre-trained CNNs and SNN model training-test processes were performed with MATLAB (2019a) in this study. Python 3.6 was used for file processing of the dataset used for the SNN model. Also, the Spyder interface was used to compile the Python code. The operating system installed on the computer was 64-bit Windows 10. Other hardware information is that the graphics card is NVIDIA GeForce 1 GB and the processor is Intel© i5-Core 2.5 GHz with 4 GB memory.

The experiment for the analysis of the study consisted of two steps, and 30% of the dataset was used as test data in all steps. The cubic SVM method was used as a classifier. The SVM method was preferred because it gives effective results in multiple classification process. In addition, the input data to be given to the SNN model in all steps of the experiment were obtained by converting the feature set with a mat file format (*.mat) provided by CNN models into a CSV file format (*.csv). The Python was used for

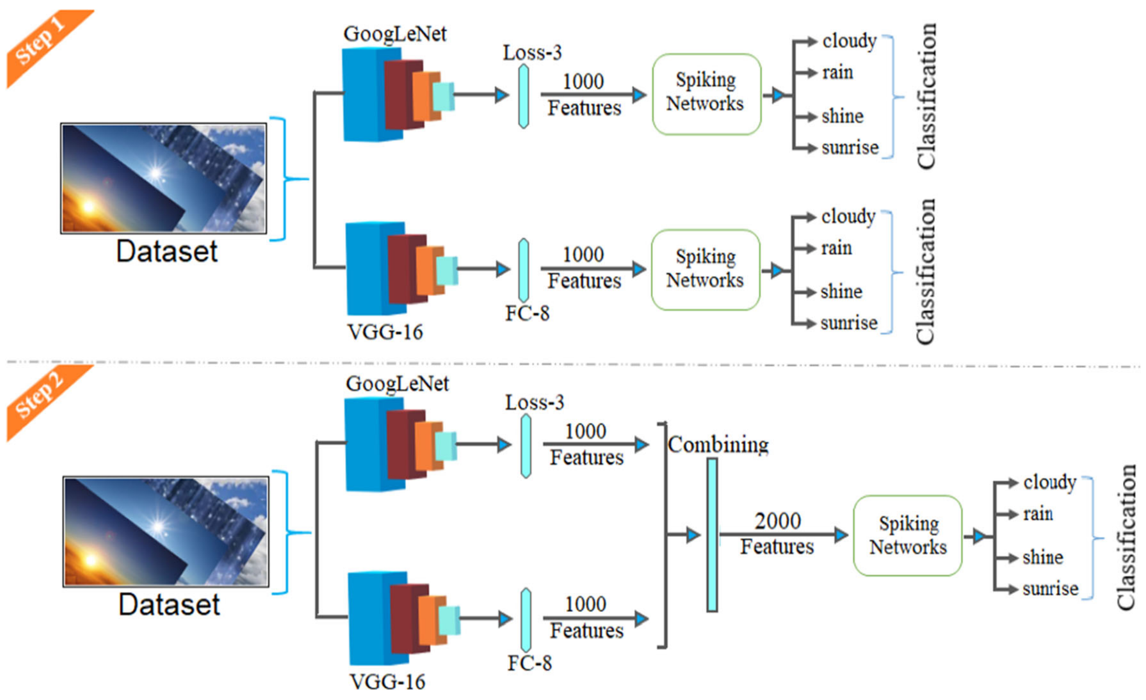

**Fig. 8** General design and process steps of the proposed approach

this file format conversion. In the first step, the dataset was trained and classified by GoogLeNet and VGG-16 models. These models were classified using loss-3 and FC8 layers, which yield 1000 features, respectively. The GoogLeNet model accuracy rates for cloudy, rain, shine, and sunrise classes were 93.94%, 98.10%, 95.68%, and 95.68%, respectively. Likewise, 1000 features extracted in the loss-3 layer of GoogLeNet architecture were given as input to the SNN model. The test accuracy rates for cloudy, rain, shine, and sunrise classes of the dataset trained by the SNN model, respectively, were 96.02%, 97.21%, 95.73%, and 96.91%. The same operations were performed in the VGG-16 model. The achievements of the VGG-16 model in the order of the data classes were mentioned above, respectively: 93.87%, 97.76%, 93.29%, and 95.92%. The features extracted in the FC-8 layer of the VGG-16 model were given to the SNN model, and the accuracy rates obtained were 94.24%, 96.88%, 95.69%, and 97.19%, respectively. In the first step of the experiment, the contribution of the SNN model was observed in all classes except the rain class. The analysis results and obtained scores are given in Table 3. The confusion matrices from which Table 3 analysis results are obtained are given in Fig. 9.

In the second step of the experiment, a new dataset with 2000 features was obtained by combining 1000 features extracted from GoogLeNet and VGG-16. The 2000-feature dataset was classified by the SVM method. The SNN model was not used in this operation. The classification accuracy rates of cloudy, rain, shine, and sunrise classes were 95.18%, 99.37%, 95.47%, and 97.73%, respectively. Besides, the overall accuracy rate of the combined model was 93.77%. In the next process, the dataset with 2000-features was trained by the SNN model, and the test accuracy rates of the classes were 98.48%, 97.58%, 97%, and 98.48%, respectively. In addition, the overall accuracy rate of the SNN model was 95.85%. The analysis results of this step are given in Table 4. When the analysis results of the second step were examined, it was seen that the SNN model contributed to the model achievement using the combined features set. Also, the confusion matrices of the analysis results performed in the second experiment are shown in Fig. 10, and the training-test graphics of the SNN model are shown in Fig. 11.

## 4 Discussion

People have to change their way of life, directly or indirectly, according to the weather conditions. Therefore, experts who analyze the aerial images and transmit the results to people produce results using traditional devices and traditional methods in their work areas. In this study, we have realized the detection of weather images together with the deep learning models and SNNs that have announced its name recently. It was seen in this study that CNN models of SNNs contribute to performance together. However, as a result of our observations, the most important deficiency of SNN models is that the weight

**Table 3** Analysis results obtained in the first step of the experiment

| CNN architecture | Using SNN | Class | Pre. (%) | Se. (%) | Spe. (%) | F-Scr. (%) | Acc. (%) |
|---|---|---|---|---|---|---|---|
| GoogLeNet | No | Cloudy | 88.89 | 88.89 | 95.83 | 88.89 | 93.94 |
| | | Rain | 95.38 | 95.38 | 98.80 | 95.38 | **98.10** |
| | | Shine | 89.61 | 92.0 | 96.79 | 90.79 | 95.68 |
| | | Sunrise | 94.29 | 92.52 | 97.24 | 93.40 | 95.68 |
| | Yes | Cloudy | 94.25 | 91.11 | 97.89 | 92.66 | **96.02** |
| | | Rain | 87.67 | 100 | 96.53 | 93.43 | 97.21 |
| | | Shine | 98.43 | 82.89 | 99.60 | 90 | **95.73** |
| | | Sunrise | 92.92 | 98.13 | 96.31 | 95.45 | **96.91** |
| VGG-16 | No | Cloudy | 87.23 | 91.11 | 94.92 | 89.13 | 93.87 |
| | | Rain | 96.77 | 92.31 | 99.19 | 94.48 | **97.76** |
| | | Shine | 83.54 | 88 | 94.86 | 85.71 | 93.29 |
| | | Sunrise | 96.08 | 91.59 | 98.11 | 93.78 | 95.92 |
| | Yes | Cloudy | 92.77 | 85.56 | 97.50 | 89.02 | **94.24** |
| | | Rain | 86.49 | 100 | 96.11 | 92.75 | 96.88 |
| | | Shine | 93.06 | 88.16 | 97.99 | 90.54 | **95.69** |
| | | Sunrise | 95.37 | 96.26 | 97.65 | 95.81 | **97.19** |

The best performances highlighted in bold

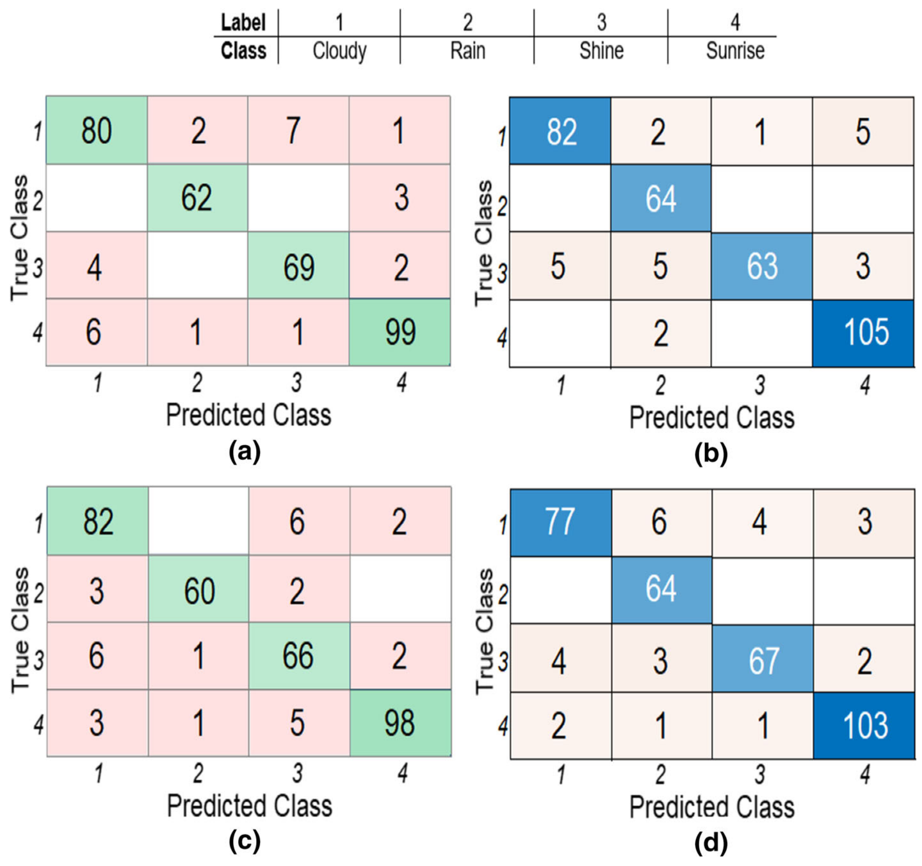

**Fig. 9** Confusion matrices obtained using CNN models of the original dataset: **a** GoogLeNet only, **b** GoogLeNet and SNN together, **c** VGG-16 only and **d** VGG-16 model and SNN together

parameters of the model cannot be realized accurately by backpropagation algorithms. Therefore, the development of SNNs is still open. However, the advantageous aspect of the proposed approach is to use SNNs with together CNN models. The comparison results with the other studies using the dataset used in this study are given in Table 5.

Ajayi et al. [53] used the same dataset in their studies and made the dataset publicly available. In their study, they proposed the selection method (SAID) based on Intuition

**Table 4** Analysis results obtained in the second step of the experiment

| CNN architecture | Using SNN | Class | Pre. (%) | Se. (%) | Spe. (%) | F-Scr. (%) | Acc. (%) |
|---|---|---|---|---|---|---|---|
| GoogLeNet and VGG-16 | No | Cloudy | 88.54 | 94.44 | 95.45 | 91.40 | 95.18 |
| | | Rain | 100 | 96.92 | 100 | 98.44 | **99.37** |
| | | Shine | 90.54 | 89.33 | 97.27 | 89.93 | 95.47 |
| | | Sunrise | 97.12 | 94.39 | 98.62 | 95.73 | 97.23 |
| | Yes | Cloudy | 95.70 | 98.89 | 98.32 | 97.27 | **98.48** |
| | | Rain | 88.89 | 100 | 97.01 | 94.12 | 97.58 |
| | | Shine | 100 | 86.84 | 100 | 92.96 | **97.0** |
| | | Sunrise | 98.11 | 97.20 | 99.10 | 97.65 | **98.48** |

The best performances highlighted in bold

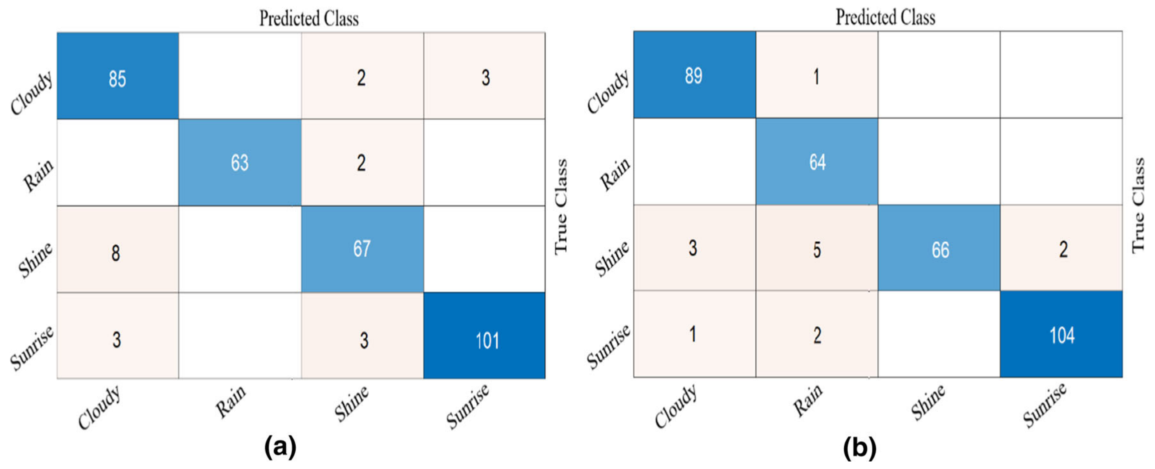

**Fig. 10** The confusion matrix obtained using the test data of the combined dataset: **a** confusion matrix of the CNN model and **b** confusion matrix of the SNN model

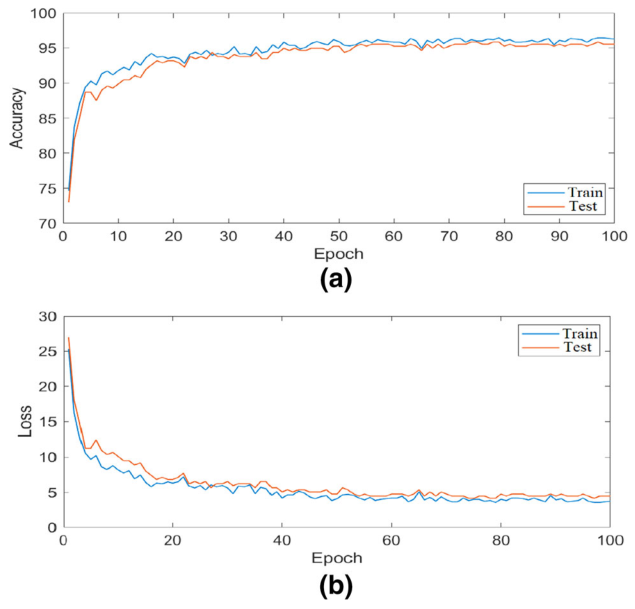

**Fig. 11** Training graphs using the combined dataset of the SNN model: **a** accuracy graph and **b** loss graph

and Diversity, using heterogeneous weather images. With this method, histograms of different types of features (saturation, contrast, local binary pattern, value, etc.) are extracted. Then, they were stacked and classified by various classification methods (k-nearest neighbor, SVM, Random Forest and Naive Bayes). The overall accuracy success of their proposed model was 86%. We believe that the histogram-based methods used in their approaches are not effective due to the low resolution of the image data. Instead of completely extracting histogram values on the image, they could also integrate CNN models into the proposed approach. This classification also had positive effects. As a result, in this study, we observed the positive effect of this situation using the CNN model.

## 5 Conclusion

In this paper, the classification of weather images was realized. For this purpose, CNNs performing automatic feature extraction were used in the proposed model. Also, it

**Table 5** Analysis results of other studies using the same dataset and analysis results of this study

| Article | Year | Class | Models/methods | Acc. (%) |
|---|---|---|---|---|
| Ajayi et al. [53] | 2019 | Cloudy | Selection based on Accuracy Intuition and Diversity of stacked ensemble algorithms (SAID) | 81.5 |
| | | Rain | | 95.2 |
| | | Shine | | 88.4 |
| | | Sunrise | | 81.7 |
| Proposed approach | 2020 | Cloudy | CNNs and SNN | **98.48** |
| | | Rain | | **97.58** |
| | | Shine | | **97.0** |
| | | Sunrise | | **98.48** |

The best performances highlighted in bold

was aimed to enhance the generalization performances of the CNNs using SNNs known as the third generation networks. The analysis results of the study confirm that the combination of CNNs and SNNs ensures better performance results. The proposed approach realized the classification of the dataset, which consists of different resolutions, using deep learning models and SNNs together. It has been observed that the combination of features extracted from CNNs contributes to the classification performance of SNNs. In the first step of the experiment, it was aimed to combine the CNN and SNN models. In the second step, the features extracted from the common layers of CNN architectures were combined, classified, and trained with the SNN model. Here, it was asked whether the combined features contribute to the results of the first step of the SNN model. As a result, the proposed approach contributed to improving the generalization ability of the model.

In future studies, we will train the SNN model using datasets in different fields, and this time we aim to use machine learning methods together with different deep learning models.

## Compliance with ethical standards

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

**Ethical approval** This article does not contain any data, or other information from studies or experimentation, with the involvement of human or animal subjects.

## References

1. Elhoseiny M, Huang S, Elgammal A (2015) Weather classification with deep convolutional neural networks. In: International conference on ımage processing
2. Renda A (2019) Artificial ıntelligence ethics, governance and policy challenges. Report of a CEPS Task Force
3. Fu X (2019) Application of artificial ıntelligence technology in medical cell biology. In: 2019 International conference on robots and ıntelligent system (ICRIS), pp 401–404
4. Zhao B, Li X, Lu X, Wang Z (2018) A CNN–RNN architecture for multi-label weather recognition. Neurocomputing 322:47–57. https://doi.org/10.1016/j.neucom.2018.09.048
5. Lu C, Lin D, Jia J, Tang C (2017) Two-class weather classification. IEEE Trans Pattern Anal Mach Intell 39:2510–2524. https://doi.org/10.1109/tpami.2016.2640295
6. An J, Chen Y, Shin H (2018) Weather classification using convolutional neural networks. In: BT—International SoC design conference, ISOCC 2018, Daegu, South Korea, November 12–15, pp 245–246
7. Villarreal Guerra JC, Khanam Z, Ehsan S et al (2018) Weather classification: a new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks. NASA/ESA Conf Adapt Hardw Syst AHS 2018:305–310. https://doi.org/10.1109/ahs.2018.8541482
8. Ajayi G (2018) Mendeley data—multi class weather dataset for image classification. https://data.mendeley.com/datasets/4drtyfjtfy/1. Accessed 28 Dec 2019
9. Toğaçar M, Ergen B, Cömert Z (2020) Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models. Measurement 153:107459. https://doi.org/10.1016/j.measurement.2019.107459
10. Cömert Z (2020) Fusing fine-tuned deep features for recognizing different tympanic membranes. Biocybern Biomed Eng 40:40–51. https://doi.org/10.1016/j.bbe.2019.11.001
11. Bochinski E, Senst T, Sikora T (2018) Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In: Proceedings of ınternational conference on ımage processing, ICIP 2017-September, pp 3924–3928. https://doi.org/10.1109/icip.2017.8297018
12. Toğaçar M, Özkurt KB, Ergen B, Cömert Z (2020) BreastNet: a novel convolutional neural network model through histopathological images for the diagnosis of breast cancer. Phys A Stat Mech Appl. https://doi.org/10.1016/j.physa.2019.123592
13. Shima Y (2018) Image Augmentation for object ımage classification based on combination of pre-trained CNN and SVM. J Phys: Conf Ser 1004:1–8. https://doi.org/10.1088/1742-6596/1004/1/012001
14. Yadav SS, Jadhav SM (2019) Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data 6:113. https://doi.org/10.1186/s40537-019-0276-2

15. Sertkaya ME, Ergen B, Togacar M (2019) Diagnosis of eye retinal diseases based on convolutional neural networks using optical coherence ımages. In: 2019 23rd International conference electronics, pp 1–5
16. Mungofa P, Schumann A, Waldo L (2018) Chemical crystal identification with deep learning machine vision. BMC Res Notes 11:703. https://doi.org/10.1186/s13104-018-3813-8
17. Nwankpa C, Ijomah W, Gachagan A, Marshall S (2018) Activation functions: comparison of trends in practice and research for deep learning, pp 1–20
18. Toğaçar M, Ergen B, Sertkaya ME (2019) Subclass separation of white blood cell ımages using convolutional neural network models. Elektron Elektrotechn 25:63–68. https://doi.org/10.5755/j01.eie.25.5.24358
19. Ahmadi M, Vakili S, Langlois JMP, Gross W (2018) Power reduction in CNN pooling layers with a preliminary partial computation strategy. In: 2018 16th IEEE ınternational new circuits and systems conference (NEWCAS), pp 125–129
20. Ghosh A, Singh S, Sheet D (2017) Simultaneous localization and classification of acute lymphoblastic leukemic cells in peripheral blood smears using a deep convolutional network with average pooling layer. In: 2017 IEEE International conference on ındustrial and ınformation systems (ICIIS), pp 1–6
21. Qu Y, Ke Y, Yu W (2018) A solution for input limit in CNN due to fully-connected layer. In: 2018 IEEE 9th ınternational conference on software engineering and service science (ICSESS), pp 611–616
22. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Proceedings of the 25th ınternational conference on neural ınformation processing systems—vol. 1. Curran Associates, USA, pp 1097–1105
23. Russakovsky O, Deng J, Su H et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115:211–252. https://doi.org/10.1007/s11263-015-0816-y
24. Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–9
25. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
26. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, pp 770–778
27. Lim S, Bae J-H, Eum J-H et al (2019) Adaptive learning rule for hardware-based deep neural networks using electronic synapse devices. Neural Comput Appl 31:8101–8116. https://doi.org/10.1007/s00521-018-3659-y
28. Başaran E, Cömert Z, Şengür A et al (2019) Chronic tympanic membrane diagnosis based on deep convolutional neural network. In: 2019 4th ınternational conference on computer science and engineering (UBMK), pp 1–4
29. Awad M, Khanna R (2015) Support vector machines for classification BT—efficient learning machines: theories, concepts, and applications for engineers and system designers. In: Awad M, Khanna R (eds) Apress, Berkeley, CA, pp 39–66
30. Doğan Ü, Glasmachers T, Igel C (2016) A unified view on multiclass support vector classification. J Mach Learn Res 17:1–32
31. Zou F, Shen L, Jie Z, et al (2018) A sufficient condition for convergences of Adam and RMSProp. 11127–11135
32. Konecny J, Richtarik P (2017) Semi-stochastic gradient descent methods. Front Appl Math Stat 3:9. https://doi.org/10.3389/fams.2017.00009
33. Huang S, Cai N, Pacheco PP, et al (2017) Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genomics Proteomics 15:41–51. https://doi.org/10.21873/cgp.20063
34. Battineni G, Chintalapudi N, Amenta F (2019) Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM). Inform Med Unlocked 16:100200. https://doi.org/10.1016/j.imu.2019.100200
35. Wu H, Wang L, Zhao Z et al (2018) Support vector machine based differential pulse-width pair brillouin optical time domain analyzer. IEEE Photonics J 10:1–11. https://doi.org/10.1109/jphot.2018.2858235
36. Sharif I, Chaudhuri D (2019) A multiseed-based SVM classification technique for training sample reduction. Turk J Electr Eng Comput Sci 27:595–604. https://doi.org/10.3906/elk-1801-157
37. Govada A, Gauri B, Sahay SK (2015) Centroid based binary tree structured SVM for multi classification. In: 2015 International conference on advances in computing, communications and ınformatics, pp 258–262
38. Lobo JL, Del Ser J, Bifet A, Kasabov N (2020) Spiking Neural Networks and online learning: An overview and perspectives. Neural Netw 121:88–100. 10.1016/j.neunet.2019.09.004
39. (2019) Spiking neural network. In: Wikipedia. https://en.wikipedia.org/wiki/Spiking_neural_network. Accessed 29 Dec 2019
40. Soni D (2018) Spiking neural networks, the next generation of machine learning. In: Towar. Data Sci. https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b. Accessed 29 Dec 2019
41. Stimberg M, Brette R, Goodman DF (2019) Brian 2, an intuitive and efficient neural simulator. Elife 8:e47314. https://doi.org/10.7554/elife.47314
42. Tavanaei A, Ghodrati M, Kheradpisheh SR, et al (2019) Deep learning in spiking neural networks. Neural Netw 111:47–63. https://doi.org/10.1016/j.neunet.2018.12.002
43. Wang W, Pedretti G, Milo V et al (2019) Computing of temporal information in spiking neural networks with ReRAM synapses. Faraday Discuss 213:453–469. https://doi.org/10.1039/c8fd00097b
44. Xie X, Qu H, Liu G et al (2016) An efficient supervised training algorithm for multilayer spiking neural networks. PLoS ONE 11:e0150329
45. Jeyasothy A, Sundaram S, Sundararajan N (2019) SEFRON: a new spiking neuron model with time-varying synaptic efficacy function for pattern classification. IEEE Trans Neural Netw Learn Syst 30:1231–1240. https://doi.org/10.1109/tnnls.2018.2868874
46. Wang X, Lin X, Dang X (2019) A delay learning algorithm based on spike train kernels for spiking neurons. Front Neurosci 13:252. https://doi.org/10.3389/fnins.2019.00252
47. Cao Y, Chen Y, Khosla D (2015) Spiking deep convolutional neural networks for energy-efficient object recognition. Int J Comput Vis 113:54–66. https://doi.org/10.1007/s11263-014-0788-3
48. Xu Q, Qi Y, Yu H, et al (2018) CSNN: An augmented spiking based framework with perceptron-inception. IJCAI Int Jt Conf Artif Intell 1646–1652. https://doi.org/10.24963/ijcai.2018/228
49. Olga R, Deng J, Su H, et al (2019) ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014). http://www.image-net.org/challenges/LSVRC/2014/. Accessed 30 Dec 2019
50. Qureshi AS, Khan A, Shamim N, Durad MH (2020) Intrusion detection using deep sparse auto-encoder and self-taught learning. Neural Comput Appl 32:3135–3147. https://doi.org/10.1007/s00521-019-04152-6
51. Toğaçar M, Ergen B, Cömert Z (2020) Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models. Measurement 158:107703. https://doi.org/10.1016/j.measurement.2020.107703

52. Akosa JS (2017) Predictive accuracy: a misleading performance measure for highly ımbalanced data. SAS Glob Forum 942:1–12
53. Ajayi GO, Wang Z (2019) Multi-class weather classification from still ımage using said ensemble method. In: Proceedings of 2019 South African Univ Power Eng Conf Mechatronics/Pattern Recognit Assoc South Africa, SAUPEC/RobMech/Prasa 2019, pp 135–140. https://doi.org/10.1109/RoboMech.2019.8704783