Chaos, Solitons and Fractals 144 (2021) 110714

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Intelligent skin cancer detection applying autoencoder, MobileNetV2 and spiking neural networks

Mesut Toğaçar[a,*], Zafer Cömert[b], Burhan Ergen[c]

[a] Department of Computer Technology, Fırat University, Elazig, Turkey
[b] Department of Software Engineering, Faculty of Engineering, Samsun University, Samsun, Turkey
[c] Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey

ARTICLE INFO

Article history:
Received 2 June 2020
Accepted 3 January 2021
Available online 3 February 2021

Keywords:
Biomedical signal processing
Decision support
Spiking neural network
Autoencoder
MobileNet
Skin cancer

ABSTRACT

Melanocytes are skin cells that give color to the skin and form melanin color pigments. The unbalanced division and proliferation of these cells result in skin cancer. The early diagnosis and proper treatment of skin cancer are so important. In this scope, a novel model that relies upon the autoencoder, spiking, and convolutional neural networks is proposed to ensure a useful decision support tool in this study. The experiments were carried out on an open-access dataset called the ISIC skin cancer consisting of 1800 being and 1497 malignant tumor images. In the proposed approach, the dataset is reconstructed using the autoencoder model. The original dataset and structured dataset were trained and classified by the MobileNetV2 model that consists of residual blocks, and the spiking networks. The classification success rate of the study was 95.27%. As a result, it was seen that the autoencoder model and spiking networks contributed to enhancing the performance of the MobileNetV2 model. Thanks to the proposed model, a novel fully automated decision support tool with high sensitivity was ensured for skin cancer detection.

1. Introduction

Skin cancer is a disease caused by abnormally dividing and proliferating cells that give color to the skin [1]. An average of 2 to 3 million people is diagnosed with skin cancer every year in the world [2]. Skin cancer is one of the most common types of cancer in the United States [3]. The countries with the most skin cancer cases in the world are New Zealand, Australia, Switzerland, Sweden and Norway [4]. Experts suggest that the human body is exposed to harmful ultraviolet rays among the most important causes of skin cancer [5]. Other causes of skin cancer are tanning, light-colored skin and freckled bodies, the use of drugs that lower the immune system at a certain time, continuous exposure to sunburn and genetic transmission [6]. Early diagnosis is important for the treatment of such diseases and the survival rate of skin cancer cases in the early diagnosis is about 99% [7].

Tumor cells are divided into two classes: benign and malignant. Benigns are well-structured cells that do not show growth or spread whereas malignants are cells that can disproportionately grow and spread [8]. Since malignant cancer cells are lethal, they undergo treatment processes such as chemotherapy, radiotherapy or surgical intervention. Tumors may show different distributions according to gender and age range. Cancers show different symptoms depending on their species and the organ they are harming. Therefore, early diagnosis is important in the treatment process of all cancer patients, including skin cancer patients [9]. Recently, deep learning architectures have exhibited an active role in the field of medical studies, and this situation is increasing day by day [10]. Detection of diseases on image data can now be realized with computer systems designed with deep learning models [11,12].

In this study, the classification has been made by using malignant tumor images that cause skin cancer and benign tumor images that do not cause skin cancer. As deep learning models, the MobileNetV2 model, autoencoder model, and spiking networks are used together. A new dataset was obtained by reconstructing the dataset with the autoencoder model. The training and classification process of the datasets was realized by combining the MobileNetV2 model with the Spiking Neural Network (SNN) model. Here, the SNN model is intended to contribute to the MobileNetV2 model. With the approach we proposed, the use of spiking networks with the deep learning model has been realized. And the contribution of spiking networks in the classification process was seen on the MobileNetV2 model. It is thought that it can contribute to the literature with this innovative aspect.

Many studies have been carried out using deep learning models in the classification of skin cancer images. If we examine some

* Corresponding author.
E-mail address: mtogacar@firat.edu.tr (M. Toğaçar).

https://doi.org/10.1016/j.chaos.2021.110714
0960-0779/

of these studies, Titus Brinker et al. [13] divided the skin cancer images into two classes as Nevus and Melanoma. They chose the ResNet-50 architecture as the Convolutional Neural Network (CNN) model. In the CNN model, a different learning rate was used for each CNN layer instead of a fixed learning rate. They also used new methods based on the cosinos function to reduce learning speeds. The sensitivity rate in the success of the classification they performed in the study was 82.3%. Khalid M. Hosny et al. [14] classified three sets of images using the International Skin Imaging Collaboration (ISIC) dataset. They processed each image in the dataset using the data augmentation technique. They then trained the dataset with pre-trained and fine-tuned AlexNet model. In the last layer of the AlexNet model, they chose the Softmax method as a classifier. In their study, the average accuracy rate of classification success was 95.91%. Anand Pandey et al. [15] divided the dataset used in their study into two classes: benign and malignant. As preprocessing, they processed the dataset images with the Gaussian filtering method. They used the data augmentation method for their dataset. In their study, they proposed a new CNN model (Adnet) for the training of the dataset. They used the Softmax method as a classifier and the success rate in the study was 87.81%. Ardan Adi Nugroho et al. [16] used the HAM10000 skin cancer data set into seven classes. As a preprocessing step, they set each image in the dataset to a resolution of 90 × 120 pixels. Then they increased the dataset allocated for training with the image augmentation method. They designed a new CNN model in their work. The test classification success with the CNN model used as the classifier of the Softmax method was 78%. Ali Mohammad Alqudah et al. [17] trained three classes of skin cancer images with CNN models. In the training process, AlexNet and GoogLeNet were used as the convolutional architectures. They divided the data set into two formats that are the dataset without segmentation, and the dataset obtained by using segmentation. They used the adaptive momentum learning rate (ADAM) as the optimization method. The classification accuracy was 92.2% in the segmentation procedure and 89.8% in the classification of non-segmentation images.

Preliminary information about the other parts of this study: information about the dataset, deep learning methods/models and the proposed approach is given in Section 2. Experimental analysis and results of the study are given in Section 3. The discussion and conclusion sections are presented in Sections 4 and 5, respectively.

2. Material and models

2.1. Dataset

The dataset consists of tumor images that occur on the skin and can be accessed on the ISIC website. The dataset consists of two classes: benign tumor and malignant tumor images. The file extension of the images is JPG, the depth is 24 bits, and each image has a fixed resolution of 224 × 224 pixels. Images were provided by specialist physicians. Biopsy samples were taken from the patients by specialist doctors and the disease was diagnosed and dermatoscopy devices were used to make the cancer cells appear more clearly. The dataset consists of 3297 images. 1800 of these images are benign and 1497 are malignant. The images were shared publicly and 25% of the dataset was shared on the website as test data and 75% as training data [18]. For this reason, we set 25% of the data set as test data and 75% as training data in all steps of the experiment (except the last step). In the last step, the classification of datasets was carried out using the cross-validation method. Example images of the dataset classes are shown in Fig. 1.

Table 1
General structure and parameters of MobileNetV2 architecture.

Type	Stride	Filter Size	Input Size
Convolution	2 × 2	3 × 3 × 3 × 32	224 × 224 × 3
Convolution dw	1 × 1	3 × 3 × 32	112 × 112 × 32
Convolution	1 × 1	1 × 1 × 32 × 64	112 × 112 × 32
Convolution dw	2 × 2	3 × 3 × 64	112 × 112 × 64
Convolution	1 × 1	1 × 1 × 64 × 128	56 × 56 × 64
Convolution dw	1 × 1	3 × 3 × 128	56 × 56 × 128
Convolution	1 × 1	1 × 1 × 128 × 128	56 × 56 × 128
Convolution dw	2 × 2	3 × 3 × 128	56 × 56 × 128
Convolution	1 × 1	1 × 1 × 128 × 256	28 × 28 × 128
Convolution dw	1 × 1	3 × 3 × 256	28 × 28 × 256
Convolution	1 × 1	1 × 1 × 256 × 256	28 × 28 × 256
Convolution dw	2 × 2	3 × 3 × 256	28 × 28 × 256
Convolution	1 × 1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Convolution dw	1 × 1	3 × 3 × 512	14 × 14 × 512
5 × Convolution	1 × 1	1 × 1 × 512 × 512	14 × 14 × 512

2.2. MobileNetV2 architecture

MobileNetV1 is known as a convolutional architecture that reduces network cost and size, developed for use with mobile devices or low-cost devices. This has led to the ease of use of image processing, and classification with deep networks in mobile devices. The MobileNetV2 model was developed on the MobileNetV1 model, and the problems related to nonlinearities in the narrow layers of the model that contain building blocks were solved in this model [19]. With the MobileNetV2 model, segmentation, classification and object recognition can be performed. The MobileNetV2 has brought two new features to its predecessor. First, some bottlenecks can occur linearly between the layers; the second is the shortcuts developed between bottlenecks. The operating principle of the MobileNetV2 model that illustrates this situation is shown in Fig. 2 [20].

The MobileNetV2 model has fewer parameters than the MobileNetV1 model, and uses less time than hardware components such as mobile devices, making it more efficient. The MobileNetV2 model is an architecture that incorporates depthwise (dw) separable filters and combination steps. This model applies a deep convolution filter with a 1 × 1 pixel resolution for each layer input. Depthwise separable convolutional filters examine inputs by separating them into two separate layers. This reduces both the speed and cost of the model. In the combination steps, the features obtained by separating them with filters are combined and a new layer is formed. MobilNetV2 model uses batchnorm and ReLU linearity in its structure [21]. In batchnorm, the model provides relaxation. It is a technique that enables and supports a high learning rate. The ReLU activation function contributes to success by ensuring that the model is non-linear [22]. The input size of the MobileNetV2 has a resolution of 224 × 224 pixels. In the last layer of this model, the Softmax function is used as a classifier. Parameters and dimensions used in the model are given in Table 1. In the input convolutional layers of this model, the filter is circulated over the input image. Thus, activation maps containing the features obtained from the image as output are created. Besides, the pooling layer reduces the input size in this model and transfers it to the next layer [23,24].

In this study, we used the MobileNetV2 model as a pre-trained model and compiled it in MATLAB [25]. The parameters used in the study for this model are the default values and the values in Table 1 are used. Support Vector Machines (SVM) were used only in the classification stage of the model in this study. Also, the other important parameters and values of the MobileNetV2 model used in the study are given in Table 2. The mini-batch is the state of the model's processing of multiple inputs at the same time, and the value selected is directly related to the performance

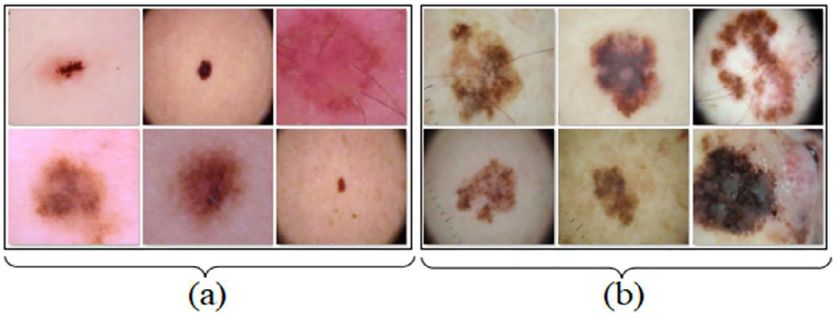

Fig. 1. Sample images used in this study; (a) benign tumors and (b) malignant tumors.

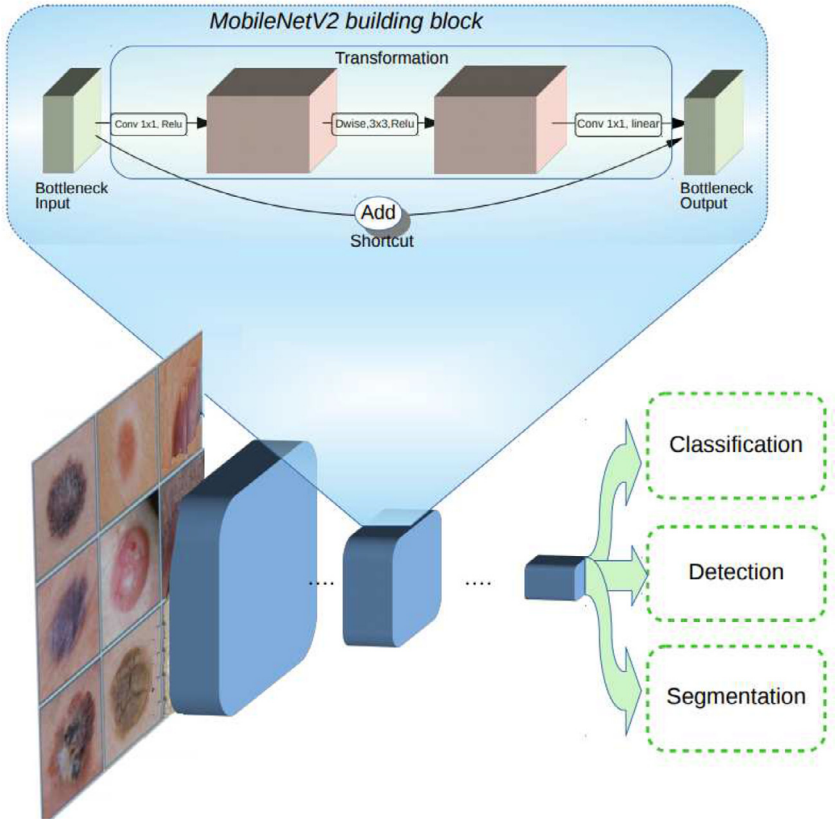

Fig. 2. General structure of MobileNetV2 model [20].

Table 2
Important parameters and values of the MobileNetV2 model used in this study.

Software	CNNs	Input Size	Optimization	Momentum	Decay	Mini Batch	Learning Rate	Classifier
MATLAB	MobileNetV2	224 × 224	Stochastic Gradient Descent (SGD)	9×10^{-1}	1e-6	64	10^{-4}	SVM

of the hardware. In this study, the mini-batch value was selected as 64.

2.3. SGD optimization method and SVM machine learning method

SGD is an optimization method used in machine learning and deep learning models. Stochastic random means a probability-linked process. The SGD method performs operations by selecting several random values instead of all the values of the dataset. Selecting the entire data set here makes the SGD method heavier; therefore, it selects a subset of the dataset and performs the processing step [26]. With optimization methods, it allows the model to improve its dataset training until the best results. The SGD method updates the weight parameters during each refresh process when performing this improvement. Thus, instead of calculating the total cost and aggravating the functioning of the model; cost is calculated step by step to achieve the best results. The SGD method uses the backpropagation technique to accomplish this. Thus, the values of parameters such as weight updates are constantly updated to reduce the loss values of the model [27]. The weight update for the SGD method is calculated according to Eq. (1). In this equation; Θ represents the weight parameter, t is the time state, α is the learning ratio, and the coordinate values of the X and Y dataset samples in the two-dimensional plane [23,28].

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

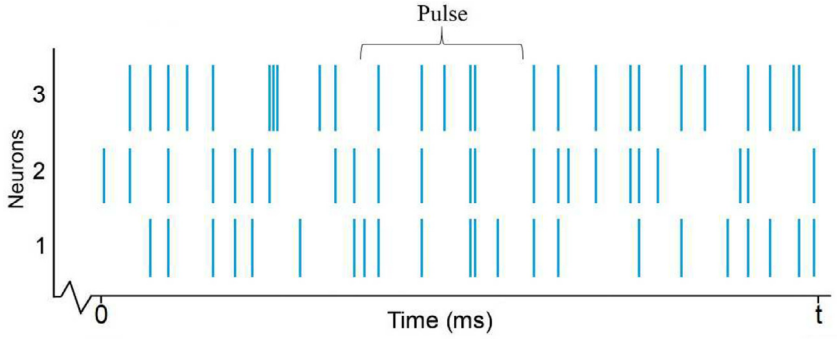

Fig. 3. Graph showing the temporal spike of neurons in spiking networks.

The SVM method is used by deep learning models; perform regression and classification operations in such models. In other words, SVM performs the classification process by finding the hyperplanes that provide the highest level of a margin between the two classes. The hyperplane is performed by the SVM method by maximizing the margin level. The SVM method has to create a hyperplane to split all labels into two classes without problems. If problems occur in dividing labels into two classes; in this case, SVM selects the least problematic hyperplane. The classification of the labels takes place by voting. In the binary classification of labels, the SVM method assigns the corresponding label to that class, whichever class gets the most [29]. In this study, the cubic SVM method was used in all steps (excluding the SNN method) of the experiment performed for MobileNetV2. Because SVM gave the best classification performance among other machine learning methods (discriminant analysis, nearest neighbor, etc.). Also, because of the best performance among SVM types (cubic, linear, quadratic, etc.), Cubic SVM was preferred in this study. Preferred parameter values in Cubic SVM method; the kernel scale parameter was automatically selected, and the box constraint level parameter value was chosen one, and the multiclass method parameter one-vs-one was selected.

2.4. Spiking neural networks (SNNs)

SNN architecture is a deep network model that examines neural structures one-on-one and performs temporal measurements of synaptic and neuronal states. In the SNN model, not every neuron is triggered immediately, and this increases the quality value of neurons. When a neuron is triggered, it creates a condition that increases or decreases the threshold value of neighboring neurons. Each neuron that exceeds the threshold value is triggered and influences the potential triggering values of other neurons around it [30]. If a neuron does not exceed the threshold value for a certain period, its activity is impaired. The triggering of neurons over time is represented by sudden rising lines (pulses) and the calculation of their values is related to the distance between the neighboring pulse. The numerical calculation of these values is done by coder and decoder methods in the Spiking network structure. Each neuron in the spiking model considers each signal transmitted to it but is not triggered immediately. It must capture the set threshold for triggering so that the signal from the neuron can be processed. As a result, each neuron producing the threshold value produces an ascending or descending output signal. These increase or decrease signals are supported by the coding scheme by the spiking model. Here, the model has to take into account both the pulse frequency and the pulse range of the signals. The decoder event then occurs, with each code corresponding to a sequence of numbers [31]. An example of the temporal representation graph of neurons exceeding the threshold value in spiking networks is given in Fig. 3. Leakage Integral and Fire algorithm (LIF) is used to calculate the pulse value in spiking networks. The mathematical formula of the LIF algorithm is given in Eq. (2). If we explain the units of the equation, it represents the current unit ($\boldsymbol{I}$), the volt unit ($\boldsymbol{V}$), the capacitor unit ($\boldsymbol{C}$), the resistor unit ($\boldsymbol{R}$) and the time unit ($\boldsymbol{t}$).

$$\boldsymbol{I(t)} - \frac{\boldsymbol{V_m(t)}}{\boldsymbol{R_m}} = \boldsymbol{C_m}\frac{\partial \boldsymbol{Vm(t)}}{\partial \boldsymbol{t}} \tag{2}$$

Spiking networks have a complex network structure, similar to Artificial Neural Networks (ANN), but do not produce continuous output (due to the threshold value). Ultimately, since each neuron interacts with neighboring neurons, it facilitates their learning [32]. Spiking networks are more costly than the ANN model because each output produced in the spiking model is simulated through various software (NEST, Brian, BindsNet and GENESIS) [33]. SNN architecture is a deep network architecture. Therefore, in Spiking networks, the model structure consists of input layer, hidden layers and output layer. This is illustrated in Fig. 4. With the convolution networks in the input layer, it moves the filter structures on the input images and extracts the activation maps. The feature set is then passed to hidden layers where continuous weight parameters are updated and backpropagations occur. As features spread through hidden layers, they engage in a more complex and challenging training process and learn to learn the SNN model. The last layer consists of fully connected layers from which the classification takes place. It performs the classification process with Synaptic Weight Association Training (SWAT) method. In addition, the biological functioning of the SNN is shown in Fig. 5 (a) and the representation by deep learning logic is shown in Fig. 5 (b).

Spiking networks are known as third neural networks, and such networks take into account the neural pulses that occur in this interval, taking into account the time intervals rather than the instantaneous output generation, such as the ANN model [34]. The SNN model uses the SWAT method to classify in the output layer of fired neurons within the network. The Spike Time Plasticity (STDP) supervised learning method makes time-based connection settings of the input and output neurons of the SNN model. SWAT prefers a learning method that facilitates training using postsynaptic voltage parameters. It also uses the STDP method within the network to facilitate 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 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 [35,36].

Fig. 6.

The SNNs appear as a sudden response pattern and each neuron is fired only once during its period. The firing rate distribution is

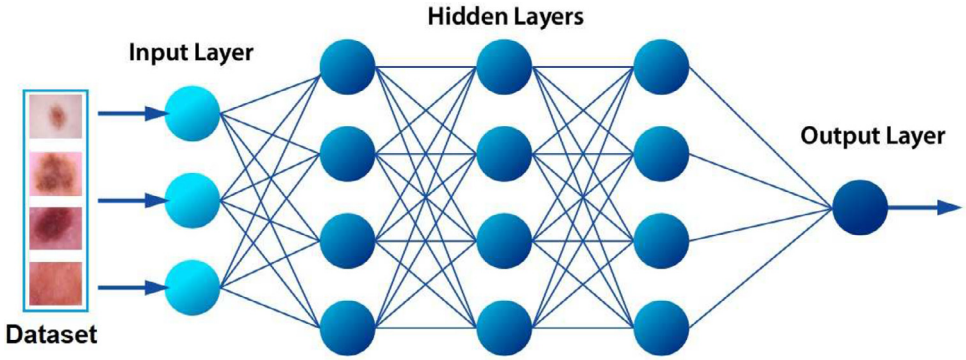

Fig. 4. General design showing layers of spiking networks.

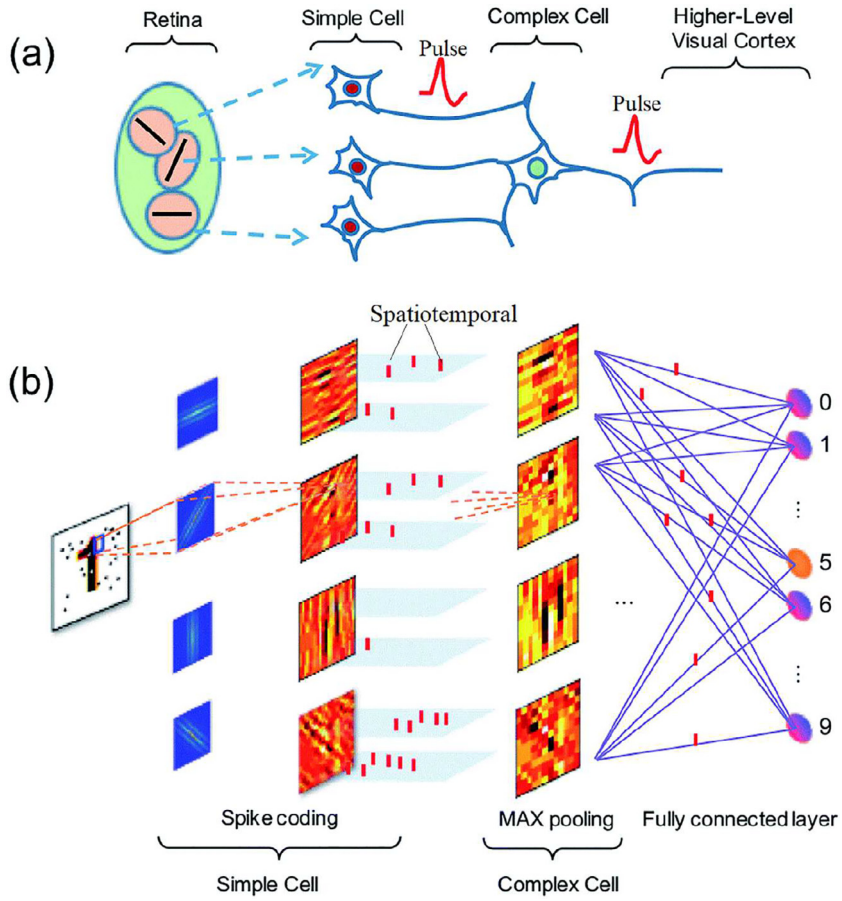

Fig. 5. Spiking networks; (a) the principle of working in a biological environment, (b) the principle of working with deep learning networks [34].

calculated according to Eq. (3). In this Equation, $\boldsymbol{x}$ is the input layer, and $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ is the output layer, $\boldsymbol{N}$ is the number of neurons. $\boldsymbol{R_{max}}$ is the firing rate and the number of neurons adjacent to the neurons in the $\boldsymbol{XX}$ coordinate. $\boldsymbol{\delta}$ represents a constant value. The delay time is important here. Each delay time in spiking networks 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 [37,38]. The parameters and related values of the Spiking model used in this study are given in Table 3.

$$f_x\left(x'\right) = R_{max} e^{\frac{\cos\left(\frac{2\pi}{N}(XX)\right)}{\delta^2}} \tag{3}$$

2.5. Autoencoder model

Deep models can produce successful results in the process of structuring the dataset. The autoencoder is an unsupervised deep learning model that processes the images in the model dataset and reproduces them in the architectural structure. The overall structure of the autoencoder model consists of three layers. These are the input layer, hidden layers, and the output layer. Each image in the dataset is processed from the input layer to extract features. Each feature is not independent of each other. If it were independent, it would reduce the efficiency of the model. The features extracted from the input layer are then processed through hidden layers called bottlenecks. Here compression is performed. This process between the input layer and hidden layers is called coder. Features that are compressed from hidden layers are then

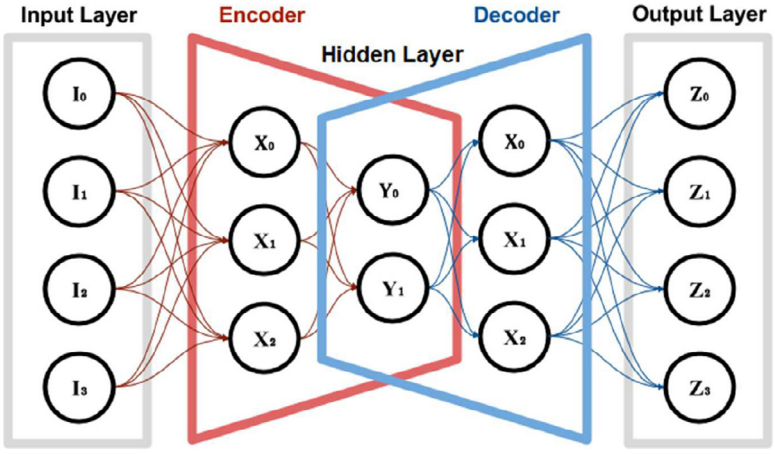

Fig. 6. Block design representing the layers of the autoencoder model [43].

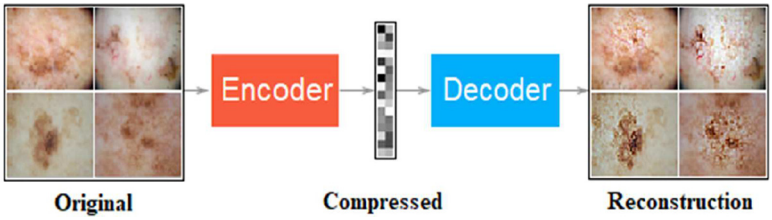

Fig. 7. Reconstruction of the original dataset with the autoencoder model, sample demonstration.

Table 3
Important parameters and values of the Spiking model used in this study.

Parameter	Value
Number of receptive field neuron in the population encoding scheme	17
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	0.01
Learning rate of weight update	0.51
Sigma of time-varying weight kernel in the millisecond	0.57
The time constant of spike response function in the millisecond	3
Number of maximum epochs	100
Time-constant of STDP learning window	2

transferred to the output layer, and the image is reconstructed. This process between the hidden layers and the output layer is called decoder [39]. The input image and the output image have the same pixel resolution. The loss ratio is important in the autoencoder model structure. Because the loss function is the minimum level, the closer to the original image (input image). In the autoencoder model, the loss function is used by the weight parameters of the model [40]. As a result, the value of x' approaches to x that minimizes the loss rate. This case is given in Eq. (4). The output layer value (x') and the hidden layer value (z) are found according to Eqs. (5) and (6). Here, σ represents the activation function value. W represents the weight matrix and b represents the bias vector [41,42]. And the design showing the operation of the images in this study with the autoencoder model is shown in Fig. 7.

$$L(x, x') = x - x'^2 \tag{4}$$

$$z = \sigma(Wx + b) \tag{5}$$

$$x' = \sigma'(W'z + b') \tag{6}$$

In this study, important parameters and values preferred for the autoencoder model are listed as follows: The mini-batch value was set to 64, and the model input and output size were set to 224 × 224 pixels. The learning rate in the autoencoder model was 10^{-3} and the SGD was selected as the optimization method for the model. Besides, the file type of each image in the configured dataset was adjusted to JPG.

2.6. Proposed approach

The proposed approach demonstrates the success of using the MobileNetV2 convolutional model with together spiking networks. Also, the dataset is structured with the autoencoder model, and the MobileNetV2 model is intended to be trained. Here, the Logits layer of the MobileNetV2 model has 1000 features. The file type of the feature set extracted in this layer is the (* .mat) file. In the SNN model (* .mat) file (* .csv) is converted to excel file type is given as input to spiking model [44].

The experiment of the proposed approach consists of two steps. The first step was to classify the dataset consisting of benign and malignant classes directly by the MobileNetV2 model and to perform the classification process by the SVM method. Similarly, the original dataset was rebuilt with the autoencoder model and the dataset created was re-trained with the MobileNetV2 model. And in the proposed approach, the Logits layer of 1000 features of the MobileNetV2 model was used. 1000 features were classified by the SVM method. The graph showing the processing stages of the first step is shown in Fig. 8.

The main step of the proposed approach is the second step. In the second step, 1000 features obtained from the original dataset and 1000 features obtained using the autoencoder model with a structured dataset were combined in the proposed approach. The combined 2000-features dataset was then trained by spiking networks and classified by SWAT in the last layer of the spiking model. The purpose of combining feature sets was to increase the success of the model. The overall design of the second step (the

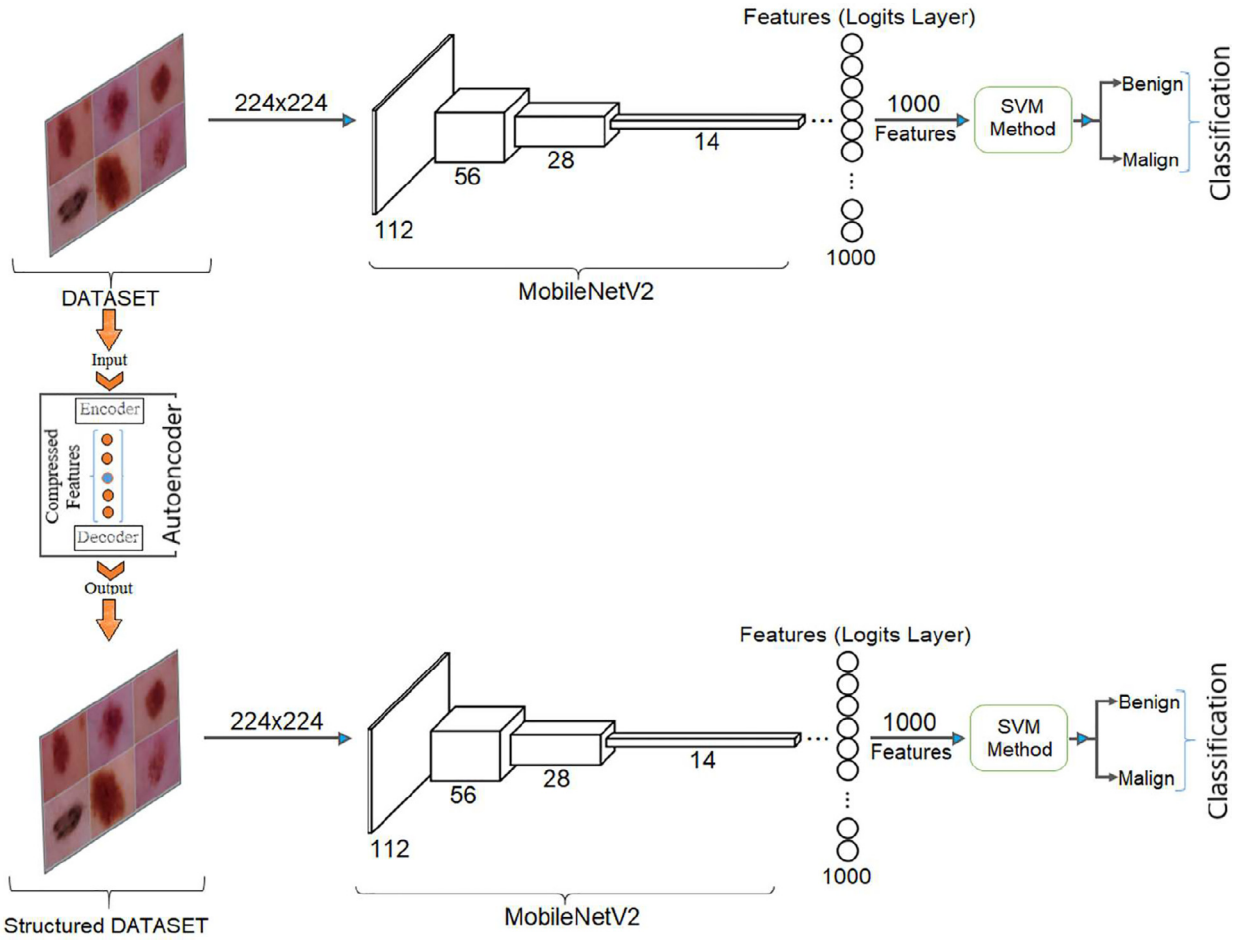

Fig. 8. Diagram showing step one of the experiment performed in the study.

proposed approach) is shown in Fig. 9. In the proposed approach, both the autoencoder model and SNN model were contributed to MobileNetV2 architecture.

3. Experiment analysis results

The confusion matrix was used for the analysis results and metric measurements performed in the experiment of this study. The sensitivity, specificity, precision, F1-score, and accuracy were used as the performance metrics. Mathematical formulas used in the calculation of metric values are given between Eqs. (7) and (11). In the Equations, TP is true positive, TN is true negative, FP is false positive, and FN is false negative [24, 45, 46].

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

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

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

$$F - Scr. = \frac{2xTP}{2xTP + FP + FN} \tag{10}$$

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Python (ver. 3.6.2) and MATLAB (2019a) were used in the experiment. The pre-trained MobileNetV2 model was compiled in MATLAB [25]. Autoencoder and Spiking network models were compiled using Python [36,47–49]. Also, feature sets extracted from the MobileNetV2 model (* mat files) were converted to excel files using Python (* .csv). Jupyter Notebook was used for the interface of Python codes [44]. The hardware information is that the processor is Intel© Xeon© Gold 6132 CPU @2.6 GHz, graphic card is NVIDIA Quadro P6000 24 GB and memory capacity is 64 GB. The operating system is Windows 10 (64 bit).

In this study, 75% of the data were used for training data and 25% for test data. Features were extracted using the Logits layer (1000 features) of the MobileNetV2 model, and the epoch was set to 100. The experiment consists of three steps. The first step consists of two processes. In the first process, training was carried out using the original dataset. In the second process, the training was carried out using the reconstructed dataset with the autoencoder model. In two processes, the data trained with the MobileNetV2 model were then classified using the SVM method. The classification success achieved with the original dataset was 86.53%. Likewise, the classification success achieved with the structured dataset was 87.86%. At the end of these two processes, it was seen that the dataset processed with autoencoder contributed to the success of the study. The performance metric results of the test analysis performed at this stage are given in Table 4. The confusion matrix results of this step are shown in Fig. 10 and the training-test accuracy graphs are shown in Fig. 11.

The second step of the experiment consisted of two processes. In this step, feature sets (1000 + 1000 = 2000) extracted from the original dataset and structured dataset with the combined convolutional models. The combined dataset consists of a total of 2000 features. In the first process of this step, the combined dataset

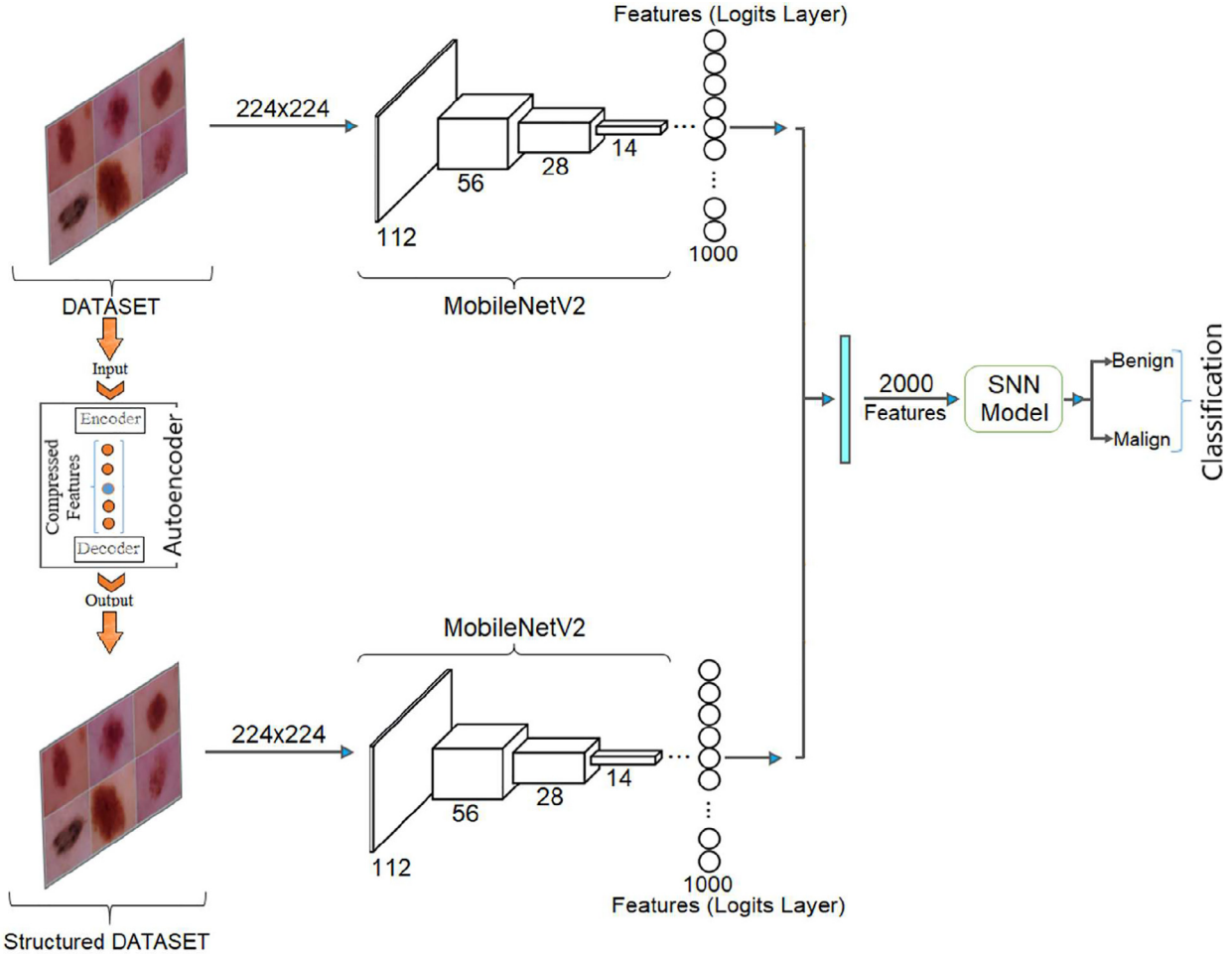

Fig. 9. Diagram showing step two of the experiment performed in the study (the proposed approach).

Table 4
Training results with the MobileNetV2 model; with the original data set and structured data set.

CNN Architecture	Using SNN	Using Autoencoder	Features	Classifier	Se. (%)	Spe. (%)	Pre. (%)	F-scr. (%)	Acc. (%)
MobileNetV2	No	No	1000	SVM	85.53	87.90	90.67	88.03	86.53
	No	Yes	1000	SVM	87.72	88.06	90.44	89.06	87.86

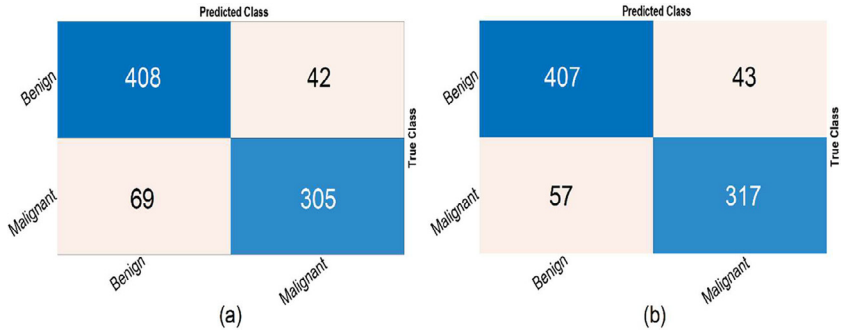

Fig. 10. The analysis results obtained in the first step of the experiment are confusion matrices; (a) the confusion matrix obtained using the original dataset, and (b) the confusion matrix obtained using the structured dataset.

was classified by the SVM method. As a result of the classification, a success rate of 93.20% was achieved. In the second process, the combined feature set was trained by the SNN model. In this process, the number of epochs was determined as 100. In Spiking networks, the SWAT algorithm was used for the classification, and the classification accuracy obtained in this process was 95.27%. The analysis results of the second step are given in Table 5, and the confusion matrix graph from the two processes is shown in Fig. 12. As a result of the second step, it was observed that spiking networks increased the classification success obtained from the

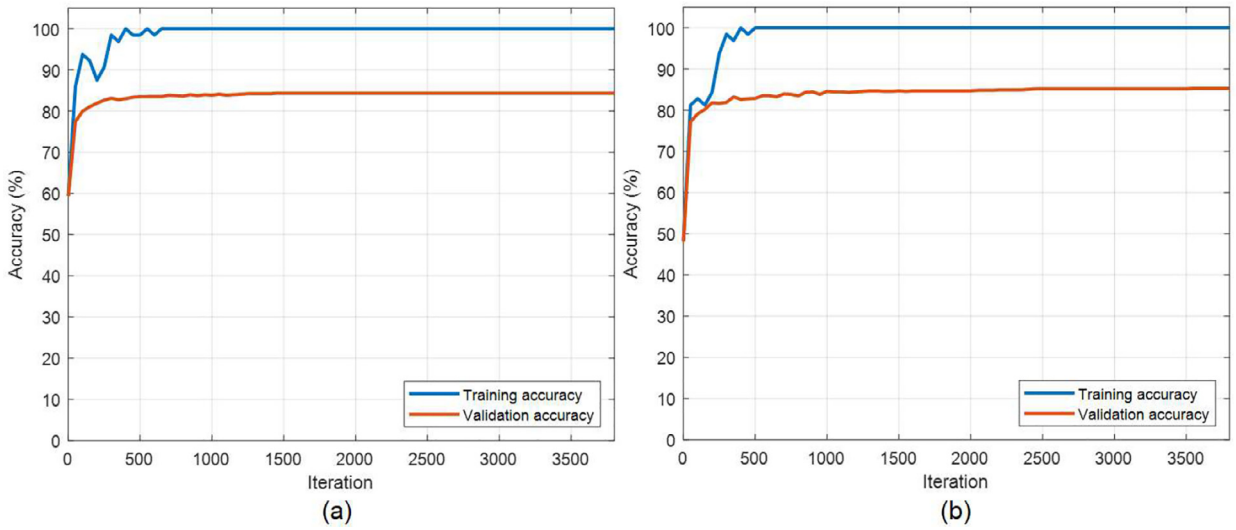

Fig. 11. The analysis results obtained in the first step of the experiment are training-test success graphs; (a) the training-test success graph obtained using the original dataset, and (b) the training-test success graph obtained using the structured dataset.

Table 5
The combined feature set; Classification by SVM method, and classification by training with Spiking model.

CNN Architecture	Dataset Status	Using SNN	Features	Classifier	Se. (%)	Spe. (%)	Pre. (%)	F-scr. (%)	Acc. (%)
MobileNetV2	Half with Autoencoder, half with original (1000 + 1000)	No	2000	SVM	93.01	93.44	94.67	93.83	93.20
		Yes	2000	SWAT	95.36	95.15	96.0	95.68	95.27

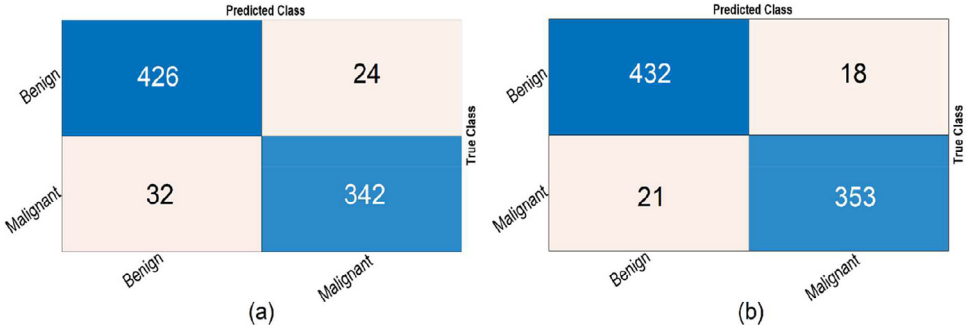

Fig. 12. The analysis results obtained in the second step of the experiment are confusion matrices; (a) the confusion matrix obtained from the classification of the combined dataset by the SVM method, and (b) the confusion matrix obtained from the SNN model of the combined dataset.

Table 6
Analysis results obtained by the SVM classifier by separating datasets with the 10-fold cross-validation method.

CNN Architecture	Dataset Status	Using SNN	Features	Se. (%)	Spe. (%)	Pre. (%)	F-scr. (%)	Acc. (%)
MobileNetV2	Original	No	1000	86.98	87.27	89.83	88.38	87.11
	Autoencoder	No	1000	87.72	86.83	89.28	88.49	87.32
	Combined	No	2000	93.72	93.32	94.50	94.11	93.54

convolutional model (MobileNetV2). The combined data of spiking networks and the success and loss graphs in the training process are shown in Fig. 13.

In the third step of the experiment, the original dataset, the autoencoder structured dataset and the combined dataset were retrained with cross-validation with the MobileNetV2 model. Here, the value of k for the cross-validation was adjusted to 10. The classification was done with the SVM method. As a result, the confusion matrices of the three dataset types obtained by the 10-fold cross-validation are shown in Fig. 14. And the analysis results obtained from the confusion matrix values are given in Table 6. When Table 6 is examined, the accuracy success obtained with the original dataset was 87.11%; the classification success of the dataset structured with the autoencoder model was 87.32% and the classification success of the combined dataset was 93.54%. These results showed us that the analyzes performed in this experiment yielded efficient results.

4. Discussion

Skin cancer is a life-threatening disease caused by unstable division and proliferation of cells under the skin. Many reasons trigger this disease. The most important one; exposure to the harmful effects of ultraviolet rays or genetic transmission. Therefore,

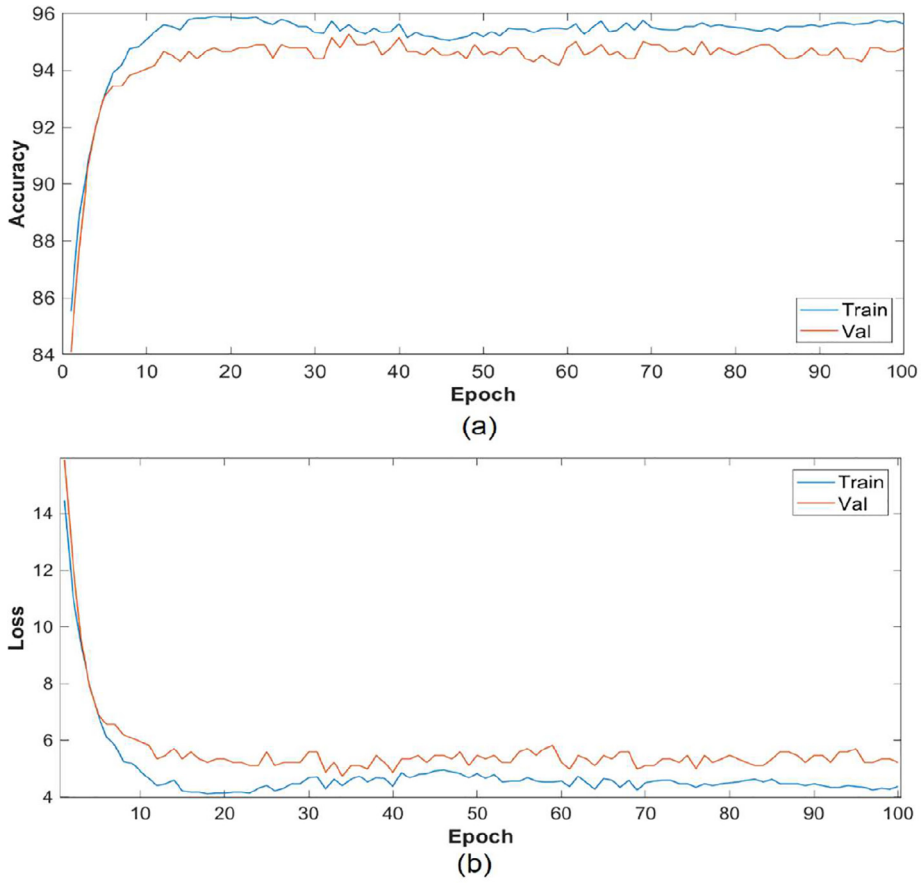

Fig. 13. The training process of the combined feature set with the SNN model; (a) success graph, (b) loss graph.

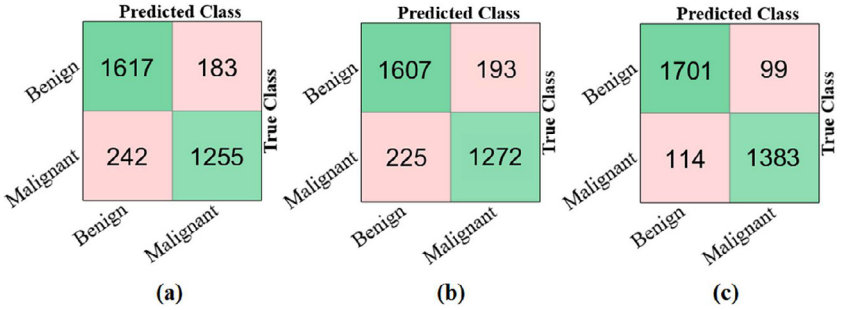

Fig. 14. Confusion matrices obtained by SVM classification of datasets by using the 10-fold cross-validation method; (a) original dataset, (b) autoencoder structured dataset, (c) combined dataset.

early detection is important in skin cancer as well as in all types of cancer. In this study, we classified the malignant tumor images that cause skin cancer and benign tumor images that do not cause skin cancer, using the MobileNetV2 convolution model. To improve the performance of the convolutional model, we reconstructed the dataset with the autoencoder model. We have contributed to performance results by combining the extracted feature sets. We also managed to increase the success achieved by training the combined feature set with spiking networks. This study shows us that different deep learning models can be used together to contribute to validation results. Limitations of this study; in the proposed approach, spiking networks must be used with a deep learning model. Since spiking networks may not alone provide efficient features in input data. Therefore, after obtaining efficient features with a deep learning model, the classification process was performed by giving the input to the spiking networks. Also, in spiking networks, parameter values have to change for different dataset types. In other words, we obtained the parameter values given in Table 3 after a preliminary trial. This may be different values for different datasets. As a result, we began the training of spiking networks by changing these values manually. A comparison of the proposed approach with other studies using the same dataset and analysis results are given in Table 7.

Muhammad A. Farooq and et al. [50] used pre-processed images of data in their study. In the pre-processing step, they were able to remove noise on the images using various filter methods. They also applied filter methods to the dataset to improve the quality of the images. Then they performed the classification by using the Softmax method by using Inception-V3 and MobileNetV1 models separately. The success with the Inception-V3 model was higher, with 86% test success. The process of pre-processing the dataset they performed in this study was extremely successful, perhaps if

Table 7
Analysis results of other studies using the same dataset and analysis results of this study.

Article	Year	Dataset Use	Models/Methods	Acc. (%)
Muhammad A. Farooq et al. [50]	2019	All	Inception-V3 & MobileNetV1 & Softmax & pre-processing methods	86.0
V. Ruthra et al. [51]	2019	Partial	CNN & Segmentation & Texture and Color Feature Extraction Methods & SVM	99.0
Proposed Approach	2020	All	MobileNetV2 & Autoencoder & SNN	95.27

we had applied the pre-processing steps in our study, we could have further improved the classification performance. But they did not use a different methodology for the convolutional models they used after the pre-processing step. If they had classified with a similar approach to our proposed approach, they could have improved the accuracy of success. V. Ruthra et al. [51] proposed a method based on texture and color feature extraction on images. They classified the extracted properties with CNN and SVM methods. They stated that the CNN method gave more successful results than the SVM method. They stated that they achieved 98% success with the SVM method and 99% success with the CNN model. But we do not think that their work is based on a valid analysis. In their study, they explained the sensitivity value of the 99% classification success obtained with the CNN model as 50% and the specificity value as 50%. Accuracy metric value certainly does not take a small value from both Sensitivity and Specificity values, nor does it take a great value. Therefore, we think that their study is not valid and not reliable. In addition, another flaw in their study is that the original dataset of training and testing undergoes both training and testing using only test images (300 malignant and 360 benign). The open-access dataset contains a total of 3297 images. Therefore, we think that analysis should be performed by using all data images in such a study.

Also, the MobileNetV2 model, which can run deep networks on personal mobile devices, has shown its usability in smart mobile devices with the successful result of our approach. Thus, it can be analyzed on mobile devices without using any hospital devices. The fact that the MobileNetV2 used in the proposed approach has fewer parameters compared to other deep models, has gained speed and time performance.

5. Conclusion

In this study, the tumor images formed on the skin were classified into two categories as benign and malignant. For this purpose, a model that provides a combination of deep learning models was proposed. To use the dataset more efficiently, it was restructured with the autoencoder model. To increase the success of the MobileNetV2 model, the model was first trained for two datasets, and feature sets were obtained. Another contribution of this study is that the feature sets obtained from the convolutional model are suitable for combining and contribute to the results. Besides, spiking networks were used together with a convolutional model. The success of the MobileNetV2 model was increased from 86.53% to 95.27% with the proposed approach. As a result, it was seen that spiking networks and the autoencoder model contributed to the success of the MobileNetV2 model.

In future studies, we will train the SNN model to identify different disease types and aim to contribute to system success by selecting less but more efficient features.

Funding

There is no funding source for this article.

Ethical approval

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

Declaration of Competing Interest

The authors declare that there is no conflict to interest related to this paper.

CRediT authorship contribution statement

Mesut Toğaçar: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Zafer Cömert:** Funding acquisition, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Burhan Ergen:** Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing.

References

[1] Kato J, Horimoto K, Sato S, Minowa T, Uhara H. Dermoscopy of melanoma and non-melanoma skin cancers. Front Med 2019;6:180. doi:10.3389/fmed.2019.00180.
[2] WHO | Skin cancers n.d. https://www.who.int/uv/faq/skincancer/en/index1.html (accessed January 6, 2020).
[3] American Academy of Dermatology n.d. https://www.aad.org/ (accessed January 6, 2020).
[4] Vogel L. Canada among top 20 countries for skin cancer risk. Can Med Assoc J 2018;190 E971–E971. doi:10.1503/cmaj.109-5643.
[5] Pratt H, Hassanin K, Troughton LD, Czanner G, Zheng Y, McCormick AG, et al. UV imaging reveals facial areas that are prone to skin cancer are disproportionately missed during sunscreen application. PLoS One 2017;12:e0185297.
[6] Skin cancer overview | cleveland clinic n.d. https://my.clevelandclinic.org/health/diseases/15818-skin-cancer (accessed January 7, 2020).
[7] Skin cancer facts & statistics - the skin cancer foundation. Ski Cancer Found; 2019. https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/ , accessed December 14, 2019.
[8] Dabeer S, Khan MM, Islam S. Cancer diagnosis in histopathological image: CNN based approach. Inf Med Unlocked 2019;16:100231. doi:10.1016/j.imu.2019.100231.
[9] Yuan T-A, Lu Y, Edwards K, Jakowatz J, Meyskens FL, Liu-Smith F. Race-, age-, and anatomic site-specific gender differences in cutaneous melanoma suggest differential mechanisms of early- and late-onset melanoma. Int J Environ Res Public Health 2019;16:908. doi:10.3390/ijerph16060908.
[10] Toğaçar M, Ergen B, Cömert Z. COVID-19 detection using deep learning models to exploit social mimic optimization and structured chest X-ray images using fuzzy color and stacking approaches. Comput Biol Med 2020;121:103805. doi:10.1016/j.compbiomed.2020.103805.
[11] Cömert Z. Fusing fine-tuned deep features for recognizing different tympanic membranes. Biocybern Biomed Eng 2019. doi:10.1016/j.bbe.2019.11.001.
[12] Toğaçar M, Ergen B, Cömert Z. BrainMRNet: brain tumor detection using magnetic resonance images with a novel convolutional neural network model. Med Hypotheses 2020:109531. doi:10.1016/j.mehy.2019.109531.
[13] Brinker TJ, Hekler A, Enk AH, Berking C, Haferkamp S, Hauschild A, et al. Deep neural networks are superior to dermatologists in melanoma image classification. Eur J Cancer 2019;119:11–17. doi:10.1016/j.ejca.2019.05.023.
[14] Hosny KM, Kassem MA, Foaud MM. Classification of skin lesions using transfer learning and augmentation with Alex-net. PLoS One 2019;14:e0217293.
[15] Pandey A., Sharma A., Ibrahim S.P.S. Clinical image analysis for detection of skin cancer using convolution neural networks 2019 61–4. doi:10.32474/trsd.2019.01.00011.
[16] Nugroho A.A., Slamet I., Sugiyanto. Skins cancer identification system of HAMl0000 skin cancer dataset using convolutional neural network. vol. 020039, 2019, p. 020039. doi:10.1063/1.5141652.

[17] Alqudah AM, Alquraan H, Qasmieh IA. Segmented and non-segmented skin lesions classification using transfer learning and adaptive moment learning rate technique using pretrained convolutional neural network. J Biomimetics Biomater Biomed Eng 2019;42:67–78. doi:10.4028/www.scientific.net/JBBBE.42.67.
[18] Fanconi C. Skin cancer: malignant vs. benign processed skin cancer pictures of the ISIC archive; 2019. https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign , accessed January 8, 2020.
[19] Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.C. MobileNetV2: inverted residuals and linear bottlenecks. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2018 4510–20. doi:10.1109/cvpr.2018.00474.
[20] Google AI Blog: MobileNetV2: the next generation of on-device computer vision networks n.d. https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html (accessed January 8, 2020).
[21] Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., et al. MobileNets: efficient convolutional neural networks for mobile vision applications 2017.
[22] Chen G., Chen P., Shi Y., Hsieh C.-Y., Liao B., Zhang S. Rethinking the usage of batch normalization and dropout in the training of deep neural networks 2019.
[23] Toğaçar M, Özkurt KB, Ergen B, Cömert Z. BreastNet: a novel convolutional neural network model through histopathological images for the diagnosis of breast cancer. Phys A Stat Mech Its Appl 2020. doi:10.1016/j.physa.2019.123592.
[24] Toğaçar M, Ergen B, Cömert Z. Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with autoencoders. Med Hypotheses 2020:109503. doi:10.1016/j.mehy.2019.109503.
[25] Pretrained MobileNet-v2 convolutional neural networkMATLAB software. MathWorks 2018. https://www.mathworks.com/help/deeplearning/ref/mobilenetv2.html accessed January 9, 2020.
[26] Cömert Z. Araştirma makalesi / research article otitis media için e vrişimsel sinir a ğlari na d ayali b ütünleşik bir tani sistemi an integrated diagnosis system based on pretrained deep convolutional neural networks for otitis media 2019 8 1498–511.
[27] Yamada KD. YamAdam: a hyperparameter-free gradient descent optimizer that incorporates unit correction and moment estimation. BioRxiv 2018:348557. doi:10.1101/348557.
[28] Toğaçar M, Ergen B, Cömert Z. Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks. Biocybern Biomed Eng 2019. doi:10.1016/j.bbe.2019.11.004.
[29] Ma Y, Zhang Q, Li D, Tian Y. Linex support vector machine for large-scale classification. IEEE Access 2019;7:70319–31. doi:10.1109/access.2019.2919185.
[30] Lobo JL, Del Ser J, Bifet A, Kasabov N. Spiking neural networks and online learning: an overview and perspectives. Neural Netw. 2020;121:88–100. doi:10.1016/j.neunet.2019.09.004.
[31] Spiking Neural Network. Wikipedia 2019. https://en.wikipedia.org/wiki/Spiking_neural_network (accessed December 29, 2019).
[32] Soni D. Spiking neural networks, the next generation of machine learning. Towar Data Sci 2018. https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b accessed December 29, 2019.
[33] Stimberg M, Brette R, Goodman DF. Brian 2, an intuitive and efficient neural simulator. Elife 2019;8:e47314. doi:10.7554/elife.47314.
[34] Wang W, Pedretti G, Milo V, Carboni R, Calderoni A, Ramaswamy N, et al. Computing of temporal information in spiking neural networks with ReRAM synapses. Farad. Discuss 2019;213:453–69. doi:10.1039/c8fd00097b.
[35] Xie X, Qu H, Liu G, Zhang M, Kurths J. An efficient supervised training algorithm for multilayer spiking neural networks. PLoS One 2016;11:e0150329.
[36] Jeyasothy A, Sundaram S, Sundararajan N. SEFRON: a new spiking neuron model with time-varying synaptic efficacy function for pattern classification. IEEE Trans Neural Networks Learn Syst 2019;30:1231–40. doi:10.1109/tnnls.2018.2868874.
[37] Wang X, Lin X, Dang X. A delay learning algorithm based on spike train kernels for spiking neurons. Front Neurosci 2019;13:252. doi:10.3389/fnins.2019.00252.
[38] The neural simulation technology initiative skip to content - spiking neural network n.d. https://www.nest-simulator.org/ (accessed January 11, 2020).
[39] Miao X, Miao H, Jia Y, Guo Y. Using a stacked-autoencoder neural network model to estimate sea state bias for a radar altimeter. PLoS One 2018;13:e0208989.
[40] Zhang Z, Wu Y, Gan C, Zhu Q. The optimally designed autoencoder network for compressed sensing. EURASIP J Image Video Process 2019;2019:56. doi:10.1186/s13640-019-0460-5.
[41] Chen CY, Huang JJ. Double deep autoencoder for heterogeneous distributed clustering. Inf 2019;10:1–15. doi:10.3390/info10040144.
[42] Qadri SF, Zhao Z, Ai D, Ahmad M, Wang Y. Vertebrae segmentation via stacked sparse autoencoder from computed tomography images. Proc.Spie 2019;11179.
[43] Toy B. Autoencoder nedir? TensorFlow ile nasıl uygulanır? - Deep learning türkiye - medium. Mediu Web 2019.
[44] NixoniteConverting mat file to csv file using Python. GitHub Web 2019. https://gist.github.com/Nixonite/bc2f69b0c4430211bcad accessed January 11, 2020.
[45] Brzezinski D, Stefanowski J, Susmaga R, Szczęch I. Visual-based analysis of classification measures and their properties for class imbalanced problems. Inf Sci (Ny) 2018;462:242–61. doi:10.1016/j.ins.2018.06.020.
[46] Budak Ü, Cömert Z, Rashid ZN, Şengür A, Çıbuk M. Computer-aided diagnosis system combining FCN and Bi-LSTM model for efficient breast cancer detection from histopathological images. Appl Soft Comput 2019;85:105765. doi:10.1016/j.asoc.2019.105765.
[47] Varugeese A. Projects - akshath123/RGB to grayscale autoencoder. Github 2019 accessed August 3, 2019. https://github.com/akshath123/RGB_to_GRAYSCALE_Autoencoder-/projects.
[48] GitHub - benjaminirving/mlseminars-autoencoders: jupyter notebook of my autoencoder presentation n.d. https://github.com/benjaminirving/mlseminars-autoencoders (accessed December 1, 2019).
[49] Spiking neural network conversion toolbox — SNN toolbox 0.3.0 documentation n.d. https://snntoolbox.readthedocs.io/en/latest/guide/intro.html (accessed January 11, 2020).
[50] Farooq MA, Khatoon A, Varkarakis V, Corcoran P. Advanced deep learning methodologies for skin cancer classification in prodromal stages. In: 27th AIAI Irish Conf Artif Intell Cogn Sci; 2019. p. 1–12.
[51] Ruthra V, Sumathy P. Color and texture based feature extraction for classifying skin cancer using support vector machine and convolutional neural network. Int Res J Eng Technol 2019;06:502–7.

Update

ctals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium ar
Volume 146, Issue , May 2021, Page

DOI: https://doi.org/10.1016/j.chaos.2021.110833

Chaos, Solitons and Fractals 146 (2021) 110833

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

journal homepage: www.elsevier.com/locate/chaos

Corrigendum

Corrigendum to "Intelligent skin cancer detection applying autoencoder, MobileNetV2 and spiking neural networks" Chaos, Solitons & Fractals 144 (2021) 110714

Mesut Toğaçar[a,*], Zafer Cömert[b], Burhan Ergen[c]

[a] Department of Computer Technology, Fırat University, Elazig, Turkey
[b] Department of Software Engineering, Faculty of Engineering, Samsun University, Samsun, Turkey
[c] Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey

In the publish article, the SWAT term was used wrongly. The SEFRON classifier was used in the proposed model. SEFRON should be understood instead of SWAT method throughout the article. The authors regret this content error. The authors apologize for any inconvenience.

DOI of original article: 10.1016/j.chaos.2021.110714
* Corresponding author.
E-mail address: mtogacar@firat.edu.tr (M. Toğaçar).

https://doi.org/10.1016/j.chaos.2021.110833