Physica D 435 (2022) 133306

Contents lists available at ScienceDirect

Physica D

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

Deep learning based classification of time series of Chen and Rössler chaotic systems over their graphic images

Burak Aricioğlu [a,*], Süleyman Uzun [b], Sezgin Kaçar [a]

[a] *Department of Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, 54187, Turkey*
[b] *Department of Computer Engineering, Sakarya University of Applied Sciences, Sakarya, 54187, Turkey*

ARTICLE INFO

Article history:
Received 16 February 2022
Received in revised form 8 April 2022
Accepted 9 April 2022
Available online 20 April 2022
Communicated by B. Hamzi

Keywords:
Chaotic systems
Time series
Classification
Deep learning
Transfer learning

ABSTRACT

In this study, the graphic images of time series of different chaotic systems are classified with deep learning methods for the first time in the literature. For the classification, a dataset contains images of time series of Chen and Rossler chaotic systems for different parameter values, initial conditions, step size and time length are generated. Then, high accuracy classifications are performed with transfer learning methods. The used transfer learning methods are *SqueezeNet, VGG-19, AlexNet, ResNet50, ResNet-101, DenseNet-201, ShuffleNet,* and *GoogLeNet*. According to the problem, classifications accuracy is varying between 89% and 99.7% in this study. Thus, this study shows that identifying a chaotic system from its graphic image of time series is possible.

1. Introduction

Chaotic systems are nonlinear systems which define chaotic behavior mathematically. There are numerous applications of chaotic systems on many different engineering areas such as data security [1], secure communication [2], image and audio encryption [3], random number generation [4], digital signature applications [5], weak signal detection [6] and DC-DC converters [7]. In the recent years, deep learning becomes very popular subject in the literature. There are many studies on deep learning most of which focus on classification processes in different fields. In this study, deep learning-based classification of time series of chaotic systems is carried out.

In the literature deep learning is used on image processing [8,9], voice recognition [10,11], natural language processing [12–14], and object recognition [15] as a sub-branch of machine learning [16]. The traditional machine learning algorithms employ predetermined rules for learning process, whereas for deep learning algorithm learning process is carried out automatically form a database [17]. To the best of the authors' knowledge, no study was found on the classification of deep learning-based chaotic systems over images of the chaotic system in the literature review. However, studies of the deep learning-based classification of chaotic systems over signals are available in the literature.

Boulle et al. [18], employed deep learning methods of ShallowNet, Multilayer perceptrons (MLP), the Fully Convolutional Neural Network (FCN), Residual Network (ResNet), and Large Kernel Convolutional Neural Network (LKCNN) to classify time series of discrete and continuous time dynamic systems. They found that the LKCNN provides highest classification performance. Yeo [19] used The Long Short Term Memory Network (LSTM) for the prediction of a chaotic system from noisy observations. He concluded that LSTM effectively filters out the noise and makes a very successful prediction for nonlinear dynamics. Kuremoto et al. [20], generated Deep Belief Nets (DBNs) by employing Restricted Boltzmann Machine (RBM) and MLP for prediction of time series of Henon map and Lorenz chaotic systems. They found that the prediction sensitivity of their proposed DBNs is higher than traditional DBNs. Sangiorgio et al. [21] proposed four model based on Feed-Forward (FF), Teacher Forcing (TF) education model and Recurrent Neural Networks methods to predict time series of chaotic oscillator. Their proposed models are FF-recursive, FF-multi-output, LSTM-TF and LSTM-no-TF. They compared classification performance of their proposed models and they found that models based on LTSM provide better prediction performance than models based on FF-recursive and FF-multi-output methods.

One of the important behaviors of dynamic systems is chaotic behaviors. Therefore, when examining dynamic systems, it is an important process to identify the signals obtained from the systems or to determine which system they belong to. Therefore, the main motivation in this study is the classification of chaotic time series with random and nonperiodic variation. One of the best methods that can be used for this process is to classify with deep-learning. As can be seen in the literature review given

[*] Corresponding author.
E-mail address: baricioglu@subu.edu.tr (B. Aricioğlu).

https://doi.org/10.1016/j.physd.2022.133306
0167-2789/© 2022 Elsevier B.V. All rights reserved.

above, there are almost no studies in the literature in which chaotic signals are classified by deep learning methods. However, classification of signals showing chaotic behaviors or random features is needed. In addition, there are deep-learning methods (such as LSTM, CNN, etc.) that can directly classify time series values for time series classification. On the other hand, there are transfer-learning networks that can be trained much faster and can classify with much higher accuracy. However, these networks can classify over images, not time series values. For this reason, the graphics of the chaotic time series obtained in this study were recorded as images and the classifications were made on these images. Consequently, in this study, the main motivation for using images of time series graphics and transfer-learning networks, the success rates of Transfer Learning methods are high, the results are obtained in a shorter period of time and the performance of transfer learning models can be increased by performing fine tuning. Therefore, in this study, classification of time series of chaotic systems based on transfer learning is performed with SqueezeNet, VGG-19, AlexNet, ResNet50, ResNet101, DenseNet201, ShuffleNet and GoogLeNet of transfer learning methods which are commonly used in the literature. So that, an original study is presented with high accuracy classification of signals belonging to two different chaotic systems, for the first time in the literature.

In the second part of the study, the used chaotic systems and the dataset obtained from them are introduced. In the third chapter, the deep learning methods used for classification are briefly explained. In the fourth chapter, simulation and performance results of classification processes are presented. Finally, in the last section the conclusion is presented.

2. Used caotic systems and generating of data set

In this section, Chen and Rossler chaotic system and dataset constructed from images of time series of these chaotic systems is presented. There are many chaotic systems in the literature. For this reason, some criteria were taken into account in the selection of chaotic systems used in this study. Chen and Rossler systems are selected because they are among the most well-known chaotic systems in the literature, they are 3-dimensional, and their mathematical models contain similar nonlinear terms. In addition, these systems can be used for modeling of atmospheric, electrical and chemical systems.

2.1. Chen system

A double scroll chaotic system was proposed by Guanrong Chen and Ueta in 1999, called the Chen System or the Chen Chaotic Attractor [22]. Chen system consists of three ordinary differential equations as given in Eq. (1).

$$\begin{aligned}\dot{x} &= a_1(y-x)\\ \dot{y} &= (a_3-a_1)x - xz + a_3 y \qquad (1)\\ \dot{z} &= xy - a_2 z\end{aligned}$$

Here a_1, a_2 and a_3 are the system parameters and x, y and z are the state variables of the system.

Time series and phase portraits of the Chen System are shown in Fig. 1 for parameter values $a_1 = 40$, $a_2 = 3$, $a_3 = 25$ and initial values $x_0 = 0.2$, $y_0 = 0.75$ and $z_0 = 0.9$. As can be seen in Fig. 1, the time series have random variations and the phase portraits showing the relationship between the two state variables have their own trajectory. By taking advantage of this property of chaotic systems, chaotic systems can be used in areas such as encryption and data security.

2.2. Rossler system

The Rossler system, developed by Otto Rossler in 1976, is known as a very useful model for modeling chemical reactions [23]. Rossler system also consists of three ordinary differential equations shown in Eq. (2).

$$\begin{aligned}\dot{x} &= -y - z\\ \dot{y} &= x + a_1 y \qquad (2)\\ \dot{z} &= a_2 + z(x - a_3)\end{aligned}$$

Here a_1, a_2 and a_3 are the system parameters and x, y and z are the state variables of the system.

While the Rossler system is similar to the Lorenz system in terms of equations, it is a single scroll system that is easier to analyze. Fig. 2 shows the time series and phase portraits of the Rossler system for parameter values $a_1 = 0.1$, $a_2 = 0.1$, $a_3 = 14$ and initial values for $x_0 = 10.5$, $y_0 = 10.5$ and $z_0 = 0$.

2.3. Creating the dataset

This section explains how to generate the dataset that is used in the study. When creating the dataset, the Runge–Kutta 4 (RK4) algorithm was used to calculate the state variables of all systems and to obtain the time series. In addition, both system and calculation parameters' values are varied to diversify time series data. In addition, while creating the dataset, the parameters to be used are determined so that the time series to be obtained are similar to each other. In this way, it is ensured that the classification problem is a difficult enough problem. 750 different results were calculated for each systems' state variable by changing the length of the time series, the step size of the RK4 algorithm, the parameter values of the systems, and the initial values. Since all systems are 3-dimensional system, 2250 different time series are generated for each system. Thus, the dataset contains total 4500 images of the time series of the chaotic systems. The used calculation and system parameters' values are shown in Table 1

Since transfer learning methods are much more effective in classifying images, graphs of all obtained time series were saved as 128 × 128 pixel images. As a result, a dataset which contains 4500 different graphical images of the time series is generated for the classification.

3. The used deep learning methods

In this study, a deep learning-based approach is presented to classify images of the time series of x, y and z state variables of two different chaotic systems. Experimental studies were carried out with the help of eight different pre-trained deep neural networks (SqueezeNet, VGG-19, AlexNet, ResNet50, ResNet-101, DenseNet-201, ShuffleNet and GoogLeNet) on the obtained images.

3.1. Deep learning and convolutional neural networks (CNN)

Deep learning is a subfield of machine learning algorithms and tries to learn differences on data using different architectures [24]. Learning process is carried out by employing multilayer neural networks in the artificial neural network based deep learning methods [25,26] (see Fig. 3).

Machine learning algorithms have traditionally been designed for use in analyzing data with few features. On the other hand, CNN has been developed to process datasets with too many features and where machine learning algorithms are not suitable to process such datasets. CNNs are much more successful than classical machine learning algorithms in processing this type of

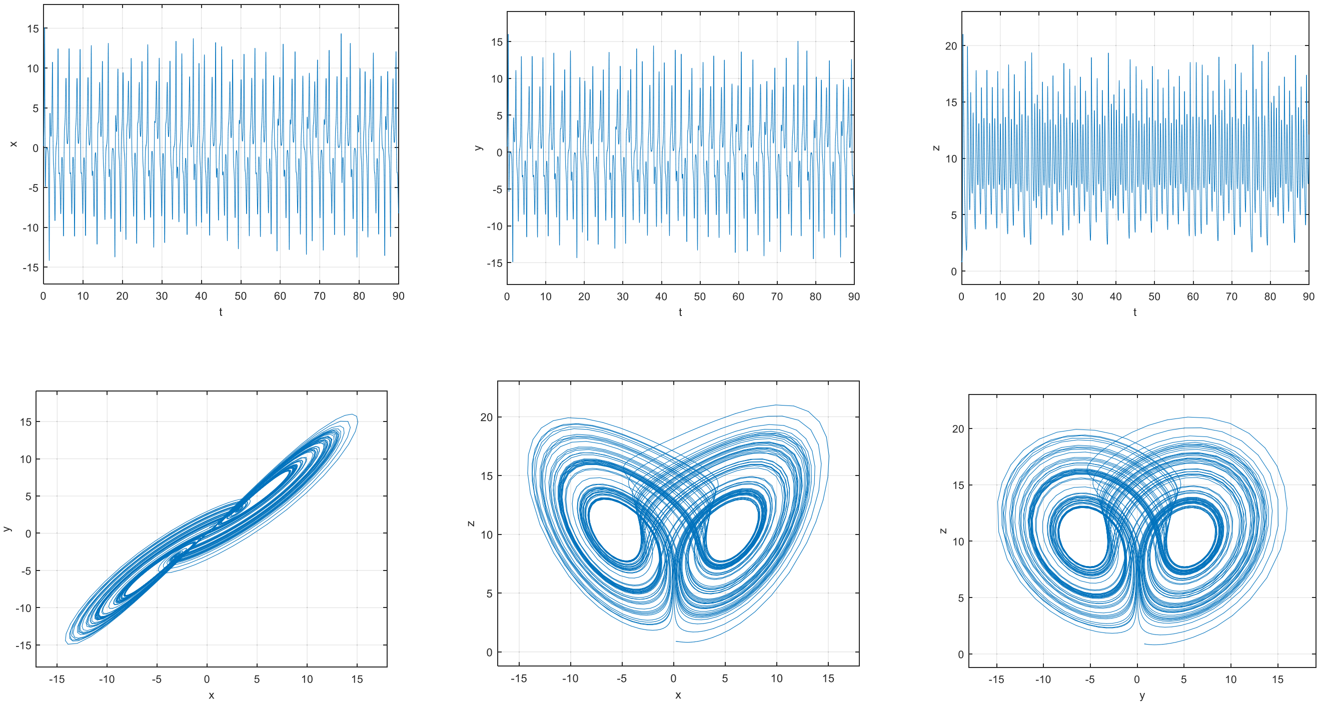

Fig. 1. Time series and phase portraits of the Chen system for system parameters $a_1 = 40$, $a_2 = 3$, $a_3 = 25$ and $\{x_0, y_0, z_0\} = \{0.2, 0.75, 0.9\}$.

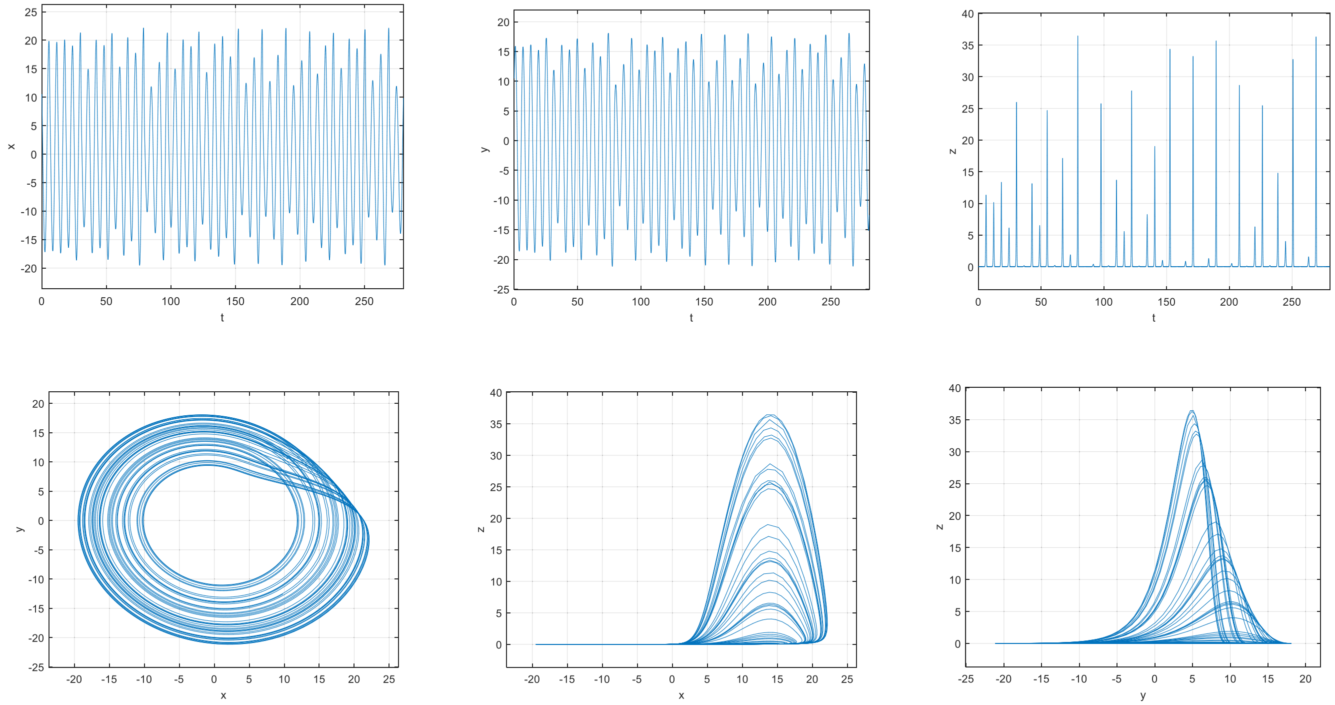

Fig. 2. Time series and phase portraits of the Rossler system for system parameters $a = 0.1$, $b = 0.1$, $c = 14$ and $\{x0, y0, z0\} = \{10.5, 10.5, 0\}$.

data, since the image data is quite large and contains hundreds or even thousands of pixels in each image [27].

Lecun et al. proposed LeNet networks in 1988 for the purpose of analyzing large-scale images, they had continued to develop this network until 1998. LeNet networks are considered to be the first CNN network models [28,29]. LeNet networks consist of sublayers that are formed by successive convolution and maximum pooling layers. The layers corresponding to the fully connected multi-layer perceptron (MLP) form the next upper layers. The weights on the network are trained by minimizing the average error between the obtained results and the predictions [27].

Many CNN network models have been proposed in the literature, some of which are architectures such as *SqueezeNet*, *VGG-19*, *AlexNet*, *ResNet50*, *ResNet-101*, *DenseNet-201*, *ShuffleNet* and *GoogLeNet* [30]. These pre-trained networks are very important in CNN models in terms of faster processing, increased performance and faster learning process. The performances of these networks vary with the problem. Therefore, pre-trained

Table 1
Calculation and system parameters used.

System	System parameters (Only a_2 parameters are changed for every system)	Initial conditions	The length of time series (the number of total calculated points)	The step size of RK4 algorithm
Chen	$a_2 = 0.05$, 0.1, 0.15, 0.2, 0.25, 0.3	$X_0 = 8$, 9, 10, 11, 12 $Y_0 = 8$, 9, 10, 11, 12 $Z_0 = 0$, 1, 2, 3, 4	20 000, 25 000, 30 000, 35 000, 40 000	0.01 0.02 0.05 0.1 0.2
Rossler	$a_2 = 2.5$, 2.55, 2.6, 2.65, 2.7, 2.75	$X_0 = 8$, 9, 10, 11, 12 $Y_0 = -8$, -9, -10, -11, -12 $Z_0 = 13$, 14, 15, 16, 17	2000, 4000, 6000, 8000, 10 000	

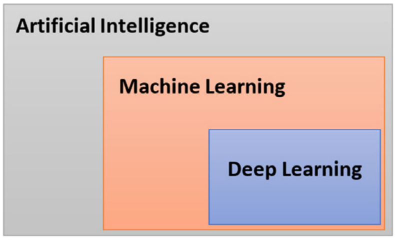

Fig. 3. The relationship between deep learning, machine learning and artificial intelligence.

networks which provide the best solution to the problem are selected. In this study, many different pre-trained networks were tested and the networks with the highest performance were used.

3.2. AlexNet

A CNN network proposed by Krizhevsky et al. which won the ImageNet competition in 2012 [31,32]. The AlexNet architecture consists of successive convolutional layers, pooling layers, and fully connected layers and uses Rectified Linear Unit (ReLU) as the activation function [33]. This architecture is generally preferred for image classification [34]. Fig. 5 shows the architecture of AlexNet (see Fig. 4).

3.3. SqueezeNet

The SqueezeNet CNN network was introduced in 2016 by Iandola et al. It is a network obtained by improving the AlexNet architecture. The difference between these two networks is that AlexNet CNN network has 240 MB of parameters, while SqueezeNet has 5 MB of parameters and they both achieve same level of classification accuracy. There are Fire layers within the SqueezeNet network. In this layer, the number of features to be calculated is reduced by reducing the filter size to 1 × 1 [36]. Accordingly, the workload in the neural network is reduced and it performs faster [37] (see Fig. 6).

3.4. VGG-19

VGG-19 network architecture contains a total of 24 layers. The 16 convolutional layers used in these layers have 3 × 3 sized filters and are used to reduce number of filter parameters. It also has 5 pooling layers and 3 fully connected layers [38] (see Fig. 7).

3.5. ResNet50 and ResNet101

The ResNet50 network architecture consists of 152 layers and won the ImageNet competition held in 2015 [40]. The architecture consists of the convolution layer, the activation layer, the down sampling (pooling) layer, and the fully connected layer. In the structure of the architecture, there are 5 convolutional blocks consisting of 1 × 1, 3 × 3 and 1 × 1 convolution layers [41]. The size of images is reduced with the global average pool layer and two-step sampling process used in the architecture [41]. In the fully connected layer, the softmax activation function is used (see Fig. 8 and Table 2).

3.6. DenseNet-201

DenseNet is a CNN network model used for the classification problem. DenseNet is a dense convolutional network with dense connection model. DenseNet architecture takes 224 × 224 × 3 images as input each time, as shown in Fig. 9.

The dense blocks used in the structure of the architecture consist of normalization layer, ReLU layer and 3 × 3 convolution layer [44]. DenseNet architecture uses concatenation layers instead of aggregating layers used in previous network architectures like ResNet architectures. Concatenation layers combine all the features from the previous layers and transfer them to the next layers. All feature maps extracted in other architectures are transferred to the next layers, while in DenseNet architecture, redundant feature maps are removed and re-learning them in the next layers is prevented.

3.7. ShuffleNet

Zhang et al. presented ShuffleNet, a highly successful CNN network model designed for mobile devices with limited data processing power [45]. This network architecture greatly reduces computational costs while maintaining accuracy. In addition, it provides superior performance compared to other architectures in ImageNet classification and MS COCO object detection [45]. The overall ShuffleNet architecture is shown in Table 3.

This network model consists of a stack of ShuffleNet units grouped in three stages as shown in Table 3. In each stage, the first block structure starts with 2 steps. The output channels for the next stage are doubled while the other parameters in one stage remain the same. The group number g shown in the table is used to control the connection sparsity of point convolutions

3.8. GoogLeNet

Szegedy et al. presented GoogLeNet, a pre-trained CNN network architecture that was selected as the most successful network in the ILSVRC2014 competition with a structure consisting of 22 layers [34,46,47]. As shown in Fig. 10, parallel layers are used in the architecture of the network to reduce the possibility of memorization.

Inception modules, which allow multi-core convolutions and maximum pooling to occur simultaneously in a single layer, enable the network to train with optimum weights and select more useful features [49]. In order to provide these operations, each initial layer contains 1 × 1, 3 × 3 and 5 × 5 variable sized convolutional cores, and an extra 3 × 3 maximum pooling layer is also used to extract more distinctive features with respect to the ones extracted from the previous layer [48].

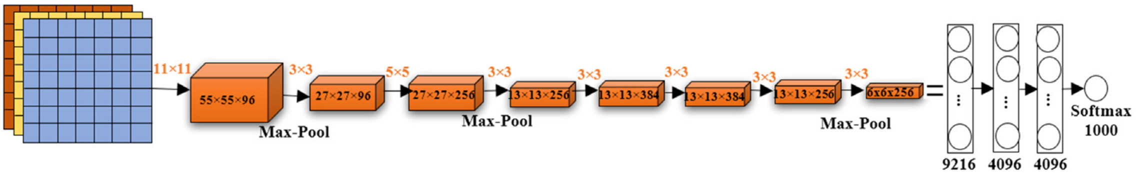

Fig. 4. The architecture of AlexNet network [35].

Table 2
Differences between ResNet architectures [42].

Layer	Output size	18-Layer	34-Layer	50-Layer	101-Layer	152-Layer
Conv1	112 × 112	7 × 7, 64, stride 2				
		3 × 3, max pool, stride 2				
Conv2_x	56 × 56	$\begin{bmatrix}3\times3 & 64\\3\times3 & 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3 & 64\\3\times3 & 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 64\\3\times3 & 64\\1\times1 & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 64\\3\times3 & 64\\1\times1 & 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 64\\3\times3 & 64\\1\times1 & 256\end{bmatrix}\times3$
Conv3_x	28 × 28	$\begin{bmatrix}3\times3 & 128\\3\times3 & 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3 & 128\\3\times3 & 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1 & 128\\3\times3 & 128\\1\times1 & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1 & 128\\3\times3 & 128\\1\times1 & 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1 & 128\\3\times3 & 128\\1\times1 & 512\end{bmatrix}\times8$
Conv4_x	14 × 14	$\begin{bmatrix}3\times3 & 256\\3\times3 & 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3 & 256\\3\times3 & 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1 & 256\\3\times3 & 256\\1\times1 & 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1 & 256\\3\times3 & 256\\1\times1 & 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1 & 256\\3\times3 & 256\\1\times1 & 1024\end{bmatrix}\times36$
Conv5_x	7 × 7	$\begin{bmatrix}3\times3 & 512\\3\times3 & 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3 & 512\\3\times3 & 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 512\\3\times3 & 512\\1\times1 & 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 512\\3\times3 & 512\\1\times1 & 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1 & 512\\3\times3 & 512\\1\times1 & 2048\end{bmatrix}\times3$
	1 × 1	avg pool, 1000-d fully connected, Softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 3
ShuffleNet architecture.

Layer	Output size	K size	Stride	Repeat	Output Channels (g groups)				
					g = 1	g = 2	g = 3	g = 4	g = 8
Image	224 × 224				3	3	3	3	3
Conv1	112 × 112	3 × 3	2	1	24	24	24	24	24
MaxPool	56 × 56	3 × 3	2						
Stage2	28 × 28		2	1	144	200	240	272	384
	28 × 28		1	3	144	200	240	272	384
Stage3	14 × 14		2	1	288	400	480	544	768
	14 × 14		1	7	288	400	480	544	768
Stage5	7 × 7		2	1	576	800	960	1088	1536
	7 × 7		1	3	576	800	960	1088	1536
GlobalPool	1 × 1	7 × 7							
FC					1000	1000	1000	1000	1000
Complexity					143M	140M	137M	133M	137M

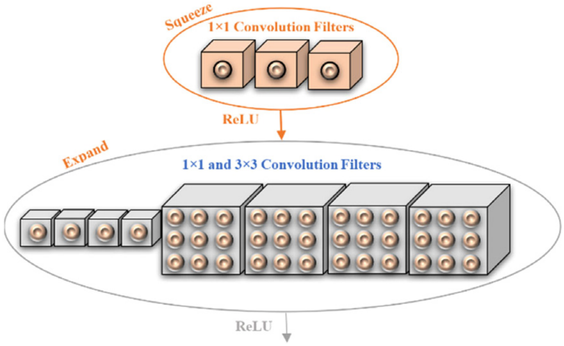

Fig. 5. SqueezeNet network fire module [36].

4. Simulation results and evaluation of performance

Firstly, 720 images of the time series of the x state variable of Chen system were created for the first class, and 600 images of the time series of the x state variable of the Rossler system were created for the second class. Then classification tests were performed using various pre-trained networks on a total of 1320 images have been carried out. In the study, tests were performed with many pre-trained networks used and the 8 pre-trained networks such as Squenzenet, VGG-19, AlexNet, ResNet50, ResNet101, DenseNet201, ShuffleNet and GoogLeNet, which have the best classification performance, were used.

The aim here is to classify the 128 × 128 images obtained from the time series of the x, y, and z state variables of the Chen and Rossler chaotic systems using deep neural networks. The epoch numbers from the parameters of the pre-trained networks indicate how many weight updates have been made in the network. This parameter is initially set to 10. In this study, we determined the amount of images in each lot from 16 pieces. The patch count specifies how many image patches are derived from the source image to constrain weight values and reduce overfitting. Throughout our tests, the value of this parameter was set to 10. In the study, firstly the images obtained from the time series of the x state variable were classified, then the images obtained from the time series of the y and z state were classified. The performances of the pre-trained networks used in these classification processes are evaluated.

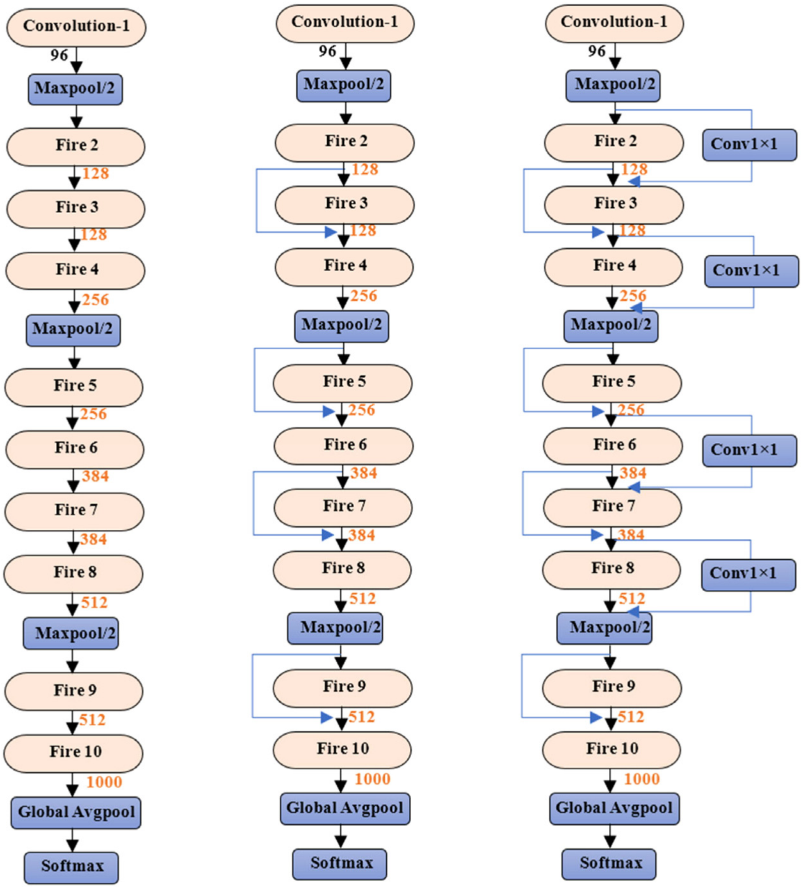

Fig. 6. The structure of SqueezeNet network [36].

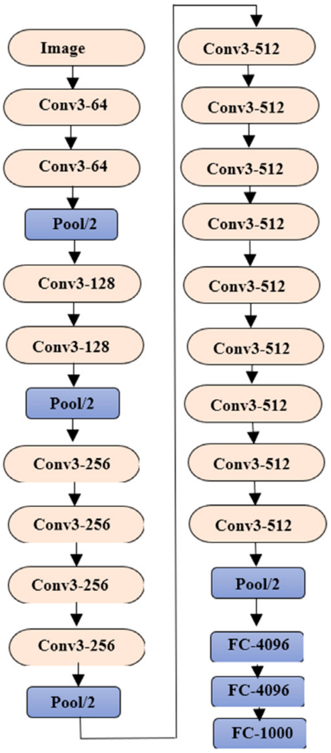

Fig. 7. The structure of VGG-19 network [39].

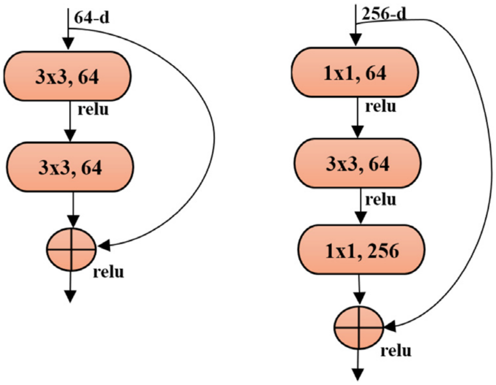

Fig. 8. Training sample of the ResNet architecture with residual layers [42].

In the first experimental study, the images obtained from the time series of the x state variable of the Chen and Rossler chaotic systems were classified. In Fig. 11, some sample images obtained from the time series of the x state variable of the Chen chaotic system used in the study are shown.

Fig. 12 shows some sample images of the time series of x state variable of Rossler chaotic system.

As shown in Figs. 11 and 12, the time series of the x state variable of Chen and Rossler chaotic systems are very close to each other. This makes classification of these time series is difficult. However, it has been observed that very successful results have been obtained thanks to the preferred pre-trained networks and the optimizations made on these networks. Table 4 shows the

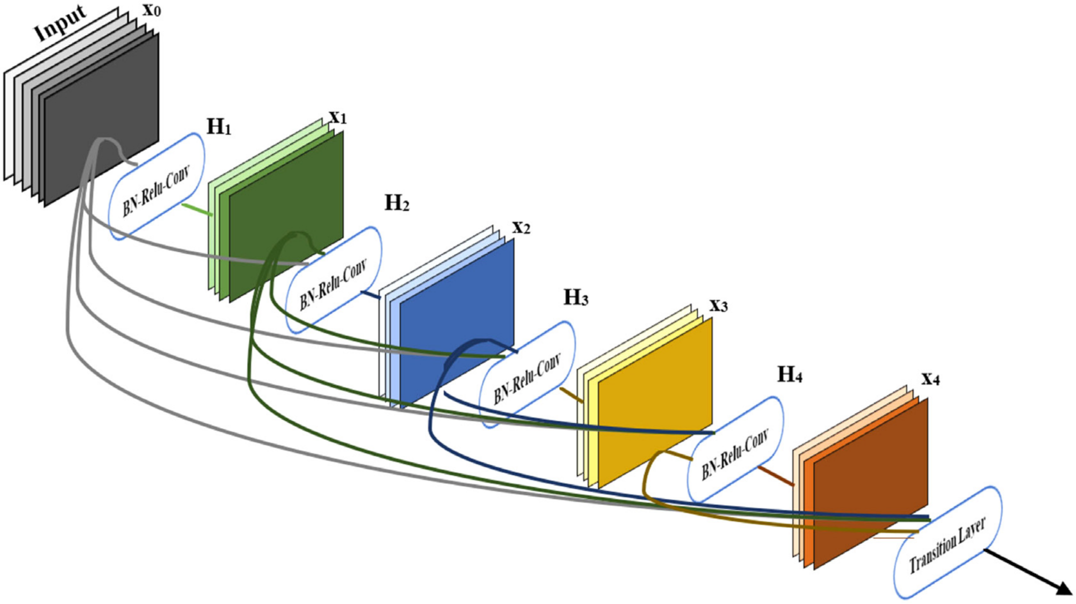

Fig. 9. The structure of DenseNet network [43].

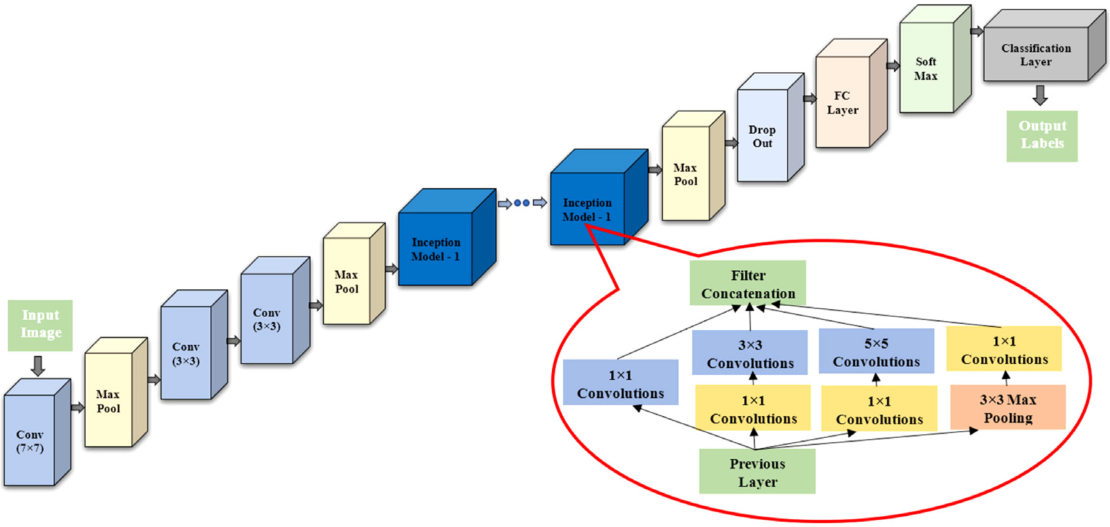

Fig. 10. The architecture of GoogLeNet [48].

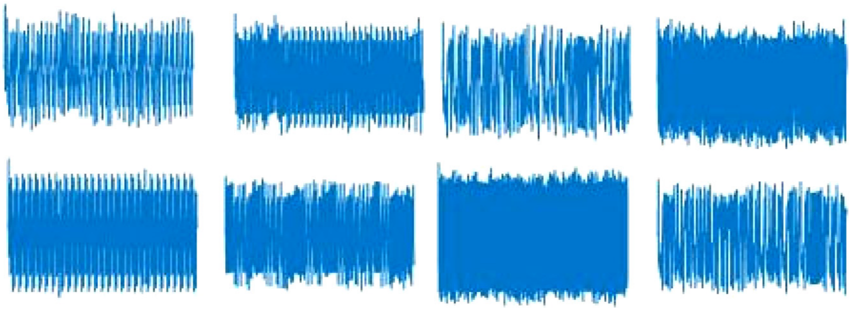

Fig. 11. Some sample images obtained from the time series of the x state variable of Chen Chaotic System.

contrast matrices of the images shown in Figs. 11 and 12. From this matrix, it is clearly seen the classification performance of the pre-trained networks.

Table 5 shows the classification performances of the images shown in Figs. 11 and 12. According to Table 5, SqueezeNet and DenseNet201 have the highest classification accuracy performance.

In the second experimental study, the images obtained from the time series of y state variable of Chen and Rossler chaotic systems were classified. In Fig. 13, some sample images obtained

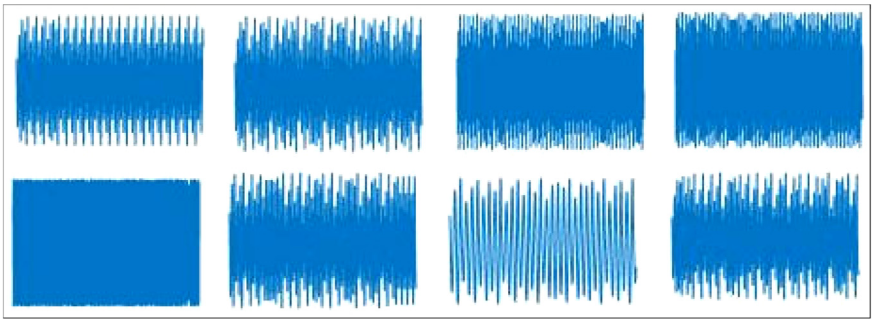

Fig. 12. Some sample images obtained from the time series of the *x* state variable of Rossler Chaotic System.

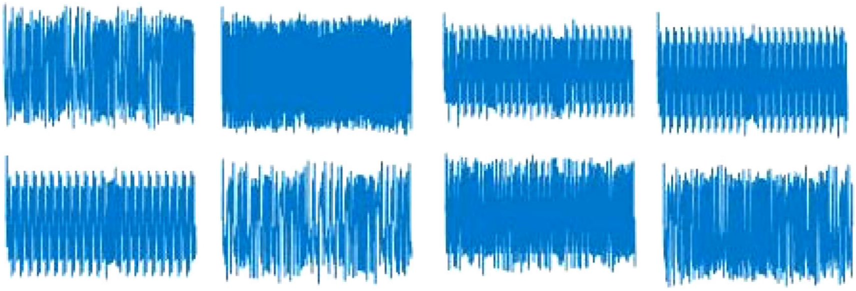

Fig. 13. Some sample images obtained from the time series of the *y* state variable of Chen Chaotic System.

Table 4
Confusion matrix of time series of x state variable of the Chen and Rossler chaotic systems.

Network	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)
SqueezeNet	175	209	5	7
VGG-19	176	178	4	38
AlexNet	162	202	18	14
ResNet50	166	211	14	5
ResNet101	180	192	0	24
DenseNet201	170	214	10	2
ShuffleNet	139	212	41	4
GoogLeNet	159	215	21	1

Table 5
Performance metrics for the time series of x state variable of the Chen and Rossler chaotic systems.

Network	Accuracy	Precision	Sensitivity	Specificity
SqueezeNet	**96.97**	97.22	96.15	97.66
VGG-19	89.39	97.78	82.24	97.80
AlexNet	91.92	90	92.05	91.82
ResNet50	95.20	92.22	97.08	93.78
ResNet101	93.94	**100**	88.24	**100**
DenseNet201	**96.97**	94.44	98.84	95.54
ShuffleNet	88.64	77.22	97.20	83.79
GoogLeNet	94.44	88.33	**99.38**	91.10

Table 6
Confusion matrix of time series of y state variable of the Chen and Rossler chaotic systems.

Network	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)
SqueezeNet	175	209	5	7
VGG-19	150	213	30	3
AlexNet	174	189	6	27
ResNet50	179	204	1	12
ResNet101	180	198	0	18
DenseNet201	179	205	1	11
ShuffleNet	151	205	29	11
GoogLeNet	158	208	22	8

Table 7
Performance metrics for the time series of y state variable of the Chen and Rossler chaotic systems.

Network	Accuracy	Precision	Sensitivity	Specificity
SqueezeNet	**96.97**	97.22	96.15	97.66
VGG-19	91.67	83.33	**98.04**	87.65
AlexNet	91.67	96.67	86.57	96.92
ResNet50	96.72	99.44	93.72	99.51
ResNet101	95.45	**100**	90.91	**100**
DenseNet201	**96.97**	99.44	94.21	99.51
ShuffleNet	89.90	83.89	93.21	87.61
GoogLeNet	92.42	87.78	95.18	90.43

from the time series of *y* state variable Chen chaotic system are shown.

In Fig. 14, some sample images obtained from the time series of *y* state variable Rossler chaotic system are shown.

The sample images given in Figs. 13 and 14 are very close to each other and this makes it difficult to classify these state variables over their images. In this second experimental study uses transfer learning methods, it has been observed that the classification performance is high thanks to the preferred pre-trained networks and the optimizations made on these networks. Table 6 shows the contrast matrices of the images shown in Figs. 13 and 14. From this matrix, it is clearly seen the classification performance of the pre-trained networks.

Table 7 shows the classification performances of the images shown in Figs. 13 and 14. According to Table 7, SqueezeNet and DenseNet201 have the highest classification accuracy performance.

In the last experimental study, the images obtained from the time series of z state variable of Chen and Rossler chaotic systems were classified. In Fig. 15, some sample images obtained from the time series of z state variable Chen chaotic system are shown.

In Fig. 16, some sample images obtained from the time series of z state variable Rossler chaotic system are shown.

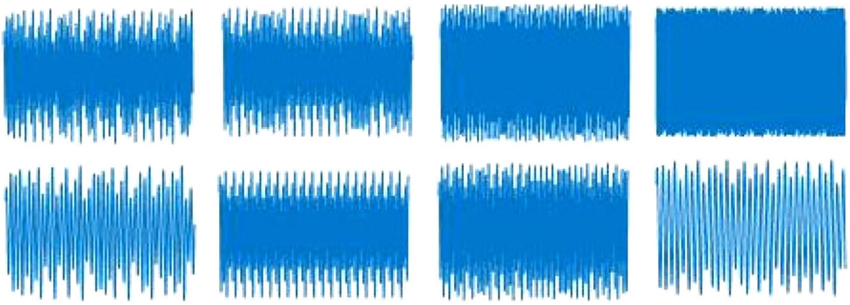

Fig. 14. Some sample images obtained from the time series of the *y* state variable of Rossler Chaotic System.

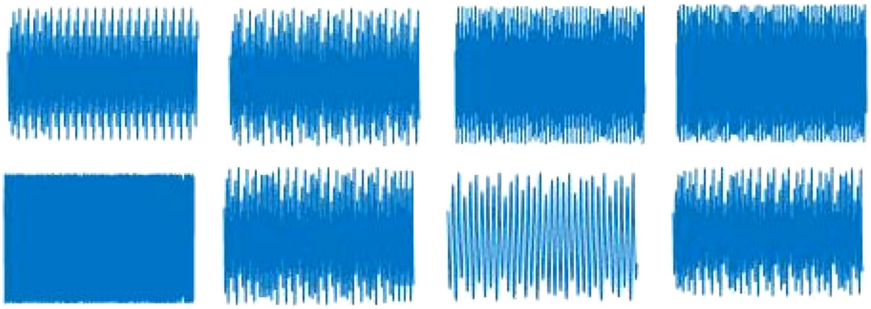

Fig. 15. Some sample images obtained from the time series of the z state variable of Chen Chaotic System.

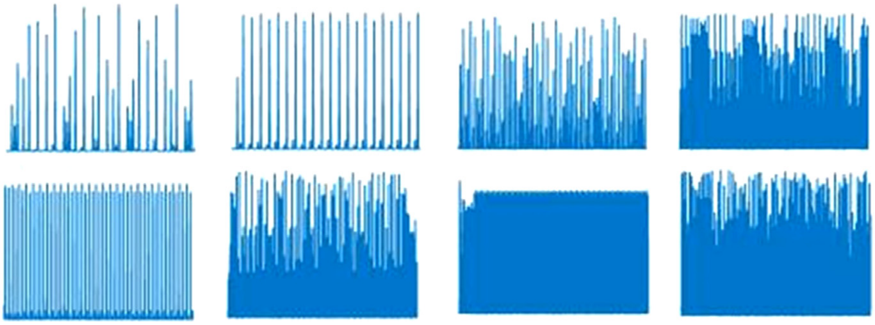

Fig. 16. Some sample images obtained from the time series of the z state variable of Rossler Chaotic System.

Table 8
Confusion matrix of time series of z state variable of the Chen and Rossler chaotic systems.

Network	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)
SqueezeNet	169	216	11	0
VGG-19	152	216	28	0
AlexNet	174	216	6	0
ResNet50	180	215	0	1
ResNet101	179	216	1	0
DenseNet201	180	214	0	2
ShuffleNet	175	214	5	2
GoogLeNet	173	216	7	0

The sample images given in Figs. 15 and 16 are very close to each other and this makes it difficult to classify these state variables over their images. In this last experimental study, it has been observed that the classification performance is high thanks to the preferred pre-trained networks and the optimizations made on these networks. Table 8 shows the contrast matrices of the images shown in Figs. 15 and 16. From this matrix, it is clearly seen the classification performance of the pre-trained networks.

Table 9 shows the classification performances of the images shown in Figs. 15 and 16. According to Table 9, ResNet50 and ResNet101 have the highest classification accuracy performance. SqueezeNet and DenseNet201, which have the best accuracy performance in the two previous experiments, have also very high accuracy performance in the last experiment too.

5. Conclusions

In this study, for the first time in the literature, time series of two different chaotic systems were classified with high accuracy using deep learning methods. For classification, time series of Chen and Rossler systems, which are the most well-known chaotic systems in the literature, were used. The time series were calculated by solving these systems with the RK4 algorithm. Calculations were performed for different step size, initial values, system parameters and time intervals, thus creating data diversity. The dataset was created by obtaining 4500 different time series in total. Afterwards, high accuracy classification was carried out using SqueezeNet, VGG-19, AlexNet, ResNet50, ResNet101, DenseNet201, ShuffleNet and GoogLeNet methods. In addition to obtaining relatively high accuracy values in each method, very high accuracy rate of 97% was obtained with the SqueezeNet

Table 9
Performance metrics for the time series of z state variable of the Chen and Rossler chaotic systems.

Network	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)
SqueezeNet	97.22	93.89	**100**	95.15
VGG-19	92.93	84.44	**100**	88.52
AlexNet	98.48	96.67	**100**	97.30
ResNet50	**99.75**	**100**	99.45	**100**
ResNet101	**99.75**	99.44	**100**	99.54
DenseNet201	99.49	**100**	98.90	**100**
ShuffleNet	98.23	97.22	98.87	97.72
GoogLeNet	98.23	96.11	**100**	96.86

and DenseNet201 method. These results show that time series belonging to chaotic systems can be classified correctly by deep learning methods. In this way, classification of real-life signals or data with chaotic or random characters over images of time series and associating them with a mathematical model is possible by employing deep learning methods.

CRediT authorship contribution statement

Burak Aricioğlu: Design and implementation of the research, Analysis of the results, Writing of the manuscript. **Süleyman Uzun:** Design and implementation of the research, Analysis of the results, Writing of the manuscript. **Sezgin Kaçar:** Design and implementation of the research, Analysis of the results, Writing of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] V. Prakash, C.T. Manimegalai, Data security using RTL algorithm with chaos synchronization for VLC system, J. Opt. (2022) 1–9, http://dx.doi.org/10.1007/s12596-022-00838-8.

[2] F. Aliabadi, M.H. Majidi, S. Khorashadizadeh, Chaos synchronization using adaptive quantum neural networks and its application in secure communication and cryptography, Neural Comput. Appl. 34 (8) (2022) 6521–6533, http://dx.doi.org/10.1007/s00521-021-06768-z.

[3] P. Sathiyamurthi, S. Ramakrishnan, S. Shobika, N. Subashri, M. Prakavi, Speech and audio cryptography system using chaotic mapping and modified Euler's system, in: Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018, 2018, pp. 606–611, http://dx.doi.org/10.1109/ICICCT.2018.8473183.

[4] F. Yu, L. Li, Q. Tang, S. Cai, Y. Song, Q. Xu, A survey on true random number generators based on chaos, Discrete Dyn. Nat. Soc. (2019) http://dx.doi.org/10.1155/2019/2545123, 2019.

[5] N. Abdoun, S. El Assad, O. Deforges, R. Assaf, M. Khalil, Design and security analysis of two robust keyed hash functions based on chaotic neural networks, J. Ambient Intell. Humaniz. Comput. 11 (5) (2020) 2137–2161, http://dx.doi.org/10.1007/s12652-019-01244-y.

[6] Q. Zhu, F. Lin, H. Li, R. Hao, Human-autonomous devices for weak signal detection method based on multimedia chaos theory, J. Ambient Intell. Humaniz. Comput. (2020) http://dx.doi.org/10.1007/s12652-020-02270-x.

[7] C.B. Fu, A.H. Tian, K.N. Yu, Y.H. Lin, H.T. Yau, Analyses and control of chaotic behavior in DC-DC converters, Math. Probl. Eng. (2018) http://dx.doi.org/10.1155/2018/7439137, 2018.

[8] P. Barros, G. Parisi, C. Weber, S.W.-. Neurocomputing, undefined, Emotion-Modulated Attention Improves Expression Recognition: A Deep Learning Model, Elsevier, 2017.

[9] A. Rakhlin, A. Shvets, V. Iglovikov, A.A. Kalinin, Deep convolutional neural networks for breast cancer histology image analysis, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNCS, vol. 10882, 2018, pp. 737–744.

[10] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C.L.Y. Bengio, A. Courville, Towards end-to-end speech recognition with deep convolutional neural networks, 2017, arXiv preprint arXiv:1701.02720.

[11] Y. Qian, M. Bi, T. Tan, K. Yu, Very deep convolutional neural networks for noise robust speech recognition, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (12) (2016) 2263–2276.

[12] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing [review article], IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75, Institute of Electrical and Electronics Engineers Inc..

[13] H. Li, Deep learning for natural language processing: advantages and challenges, Natl. Sci. Rev. (2017).

[14] L. Deng, Y. Liu, Deep learning in natural language processing, 2018.

[15] S. Toraman, Pedestrian detection with deep learning from unmanned aerial imagery, J. Aviat. 2 (2) (2018) 64–69.

[16] U. Kaya, A. Yılmaz, Y. Dikmen, İ. Kavram, Deep learning methods used in the field of health, Eur. J. Sci. Technol. (16) (2019) 792–808.

[17] G. Altan, DeepGraphNet: DEep learning models in the classification of graphs, Eur. J. Sci. Technol. Spec. Issue (2019) 319–329.

[18] N. Boullé, V. Dallas, Y. Nakatsukasa, D. Samaddar, Classification of chaotic time series with deep learning, Physica D 403 (2020) 132261.

[19] K. Yeo, Model-free prediction of noisy chaotic time series by deep learning, ArXiv (2017).

[20] T. Kuremoto, M. Obayashi, K. Kobayashi, T. Hirata, S. Mabu, Forecast chaotic time series data by DBNs, in: Proceedings - 2014 7th International Congress on Image and Signal Processing, CISP 2014, 2014, pp. 1130–1135.

[21] M. Sangiorgio, F. Dercole, Robustness of LSTM neural networks for multi-step forecasting of chaotic time series, Chaos Solitons Fractals 139 (2020) 110045.

[22] G. Chen, T. Ueta, Yet another chaotic attractor, Int. J. Bifurc. Chaos 9 (7) (1999) 1465–1466.

[23] O.E. Rössler, An equation for continuous chaos, Phys. Lett. A 57 (5) (1976) 397–398.

[24] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

[25] G. Haskins, U. Kruger, P. Yan, Deep learning in medical image registration: A survey, Mach. Vis. Appl. 31 (1) (2019).

[26] D. Küçük, N. Arici, A literature study on deep learning applications in natural language processing, Int. J. Manage. Inf. Syst. Comput. Sci. 2 (2) (2018) 76–86.

[27] A. Keles, A. Ibrahim, Deep learning and applications in health, Turk. Stud. Inf. Technol. Appl. Sci. 13 (21) (2018) 113–127.

[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2323.

[29] Y. Le Cun others, Handwritten digit recognition: Applications of neural net chips and automatic learning, in: Neurocomputing, Springer Berlin Heidelberg, 1990, pp. 303–318.

[30] D. Ravi others, Deep learning for health informatics, IEEE J. Biomed. Health Inform. 21 (1) (2017) 4–21.

[31] S. Lu, Z. Lu, Y.D. Zhang, Pathological brain detection based on AlexNet and transfer learning, J. Comput. Sci. 30 (2019) 41–47.

[32] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[33] T.F. Gonzalez, Handbook of approximation algorithms and metaheuristics, 2007.

[34] P. Ballester, R.M. Araujo, On the performance of GoogLeNet and AlexNet applied to sketches, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[35] X. Han, Y. Zhong, L. Cao, L. Zhang, Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification, Remote Sens. 9 (8) (2017) 848.

[36] N. Dhungel, G. Carneiro, A.P. Bradley, Under review as a conference paper at ICLR 2017 SqueezeNet: AlexNet-level accuracy with 50X fewer parameters and <0.5MB model size, in: Advances in Computer Vision and Pattern Recognition (9783319429984), 2017, pp. 225–240.

[37] B. Aksoy, H.D. Halis, O.K.M. Salman, Identification of diseases in apple plants with artificial intelligence methods and comparison of the performance of artificial intelligence methods, Int. J. Eng. Innov. Res. 2 (3) (2020) 194–210.

[38] M. Toğaçar, B. Ergen, F. Özyurt, Classification of Flower Images By using Feature Selection Methods in Convolutional Neural Network Models, Fırat Üniversitesi Mühendislik Bilim. Derg, 2020, pp. 37–45.

[39] L. Wen, X. Li, X. Li, L. Gao, A new transfer learning based on VGG-19 network for fault diagnosis, in: Proceedings of the 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design, CSCWD 2019, 2019, pp. 205–209.

[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-Decem, 2016, pp. 770–778.

[41] M. Talo, Classification of histopathological breast cancer images using convolutional neural networks, Fırat Univ. J. Eng. Sci. 31 (2) (2018) 391–398.

[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[43] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2017.

[44] F. Yilmaz, O. Kose, A. Demir, Comparison of two different deep learning architectures on breast cancer, in: TIPTEKNO 2019 - Tip Teknolojileri Kongresi, Vol. 2019-January, 2019.

[45] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov …, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[47] O. Russakovsky others, ImageNet Large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[48] L. Balagourouchetty, J.K. Pragatheeswaran, B. Pottakkat, G. Ramkumar, GoogLeNet-Based ensemble FCNet classifier for focal liver lesion diagnosis, IEEE J. Biomed. Heal. Inform. 24 (6) (2020) 1686–1694.

[49] Z. Zhu, J. Li, L. Zhuo, J. Zhang, Extreme weather recognition using a novel fine-tuning strategy and optimized GoogLeNet, in: DICTA 2017-2017 International Conference on Digital Image Computing: Techniques and Applications, Vol. 2017-December, 2017, pp. 1–7.