Computers, Materials & Continua
DOI:10.32604/cmc.2022.018860
Article

Tech Science Press

An Improved Convolutional Neural Network Model for DNA Classification

Naglaa. F. Soliman[1,*], Samia M. Abd-Alhalem[2], Walid El-Shafai[2], Salah Eldin S. E. Abdulrahman[3], N. Ismaiel[3], El-Sayed M. El-Rabaie[2], Abeer D. Algarni[1] and Fathi E. Abd El-Samie[1,2]

[1]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia
[2]Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, 32952, Egypt
[3]Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, 32952, Egypt
[*]Corresponding Author: Naglaa. F. Soliman. Email: nfsoliman@pnu.edu.sa
Received: 24 March 2021; Accepted: 11 June 2021

Abstract: Recently, deep learning (DL) became one of the essential tools in bioinformatics. A modified convolutional neural network (CNN) is employed in this paper for building an integrated model for deoxyribonucleic acid (DNA) classification. In any CNN model, convolutional layers are used to extract features followed by max-pooling layers to reduce the dimensionality of features. A novel method based on downsampling and CNNs is introduced for feature reduction. The downsampling is an improved form of the existing pooling layer to obtain better classification accuracy. The two-dimensional discrete transform (2D DT) and two-dimensional random projection (2D RP) methods are applied for downsampling. They convert the high-dimensional data to low-dimensional data and transform the data to the most significant feature vectors. However, there are parameters which directly affect how a CNN model is trained. In this paper, some issues concerned with the training of CNNs have been handled. The CNNs are examined by changing some hyperparameters such as the learning rate, size of minibatch, and the number of epochs. Training and assessment of the performance of CNNs are carried out on 16S rRNA bacterial sequences. Simulation results indicate that the utilization of a CNN based on wavelet subsampling yields the best trade-off between processing time and accuracy with a learning rate equal to 0.0001, a size of minibatch equal to 64, and a number of epochs equal to 20.

Keywords: DNA classification; CNN; downsampling; hyperparameters; DL; 2D DT; 2D RP

1 Introduction

Technological advances in DNA sequencing allowed sequencing of the genome at a low cost within a reasonable period. These advances induced a huge increase in the available genomic data. Bioinformatics addresses the need to manage and interpret the data that is massively generated by

genomic research. Computational DNA classification is among the main challenges, which play a vital role in the early diagnosis of serious diseases. Advances in machine learning techniques are expected to improve the classification of DNA sequences [1]. Recently, survey studies have been presented by Leung et al. [2], Mamoshina et al. [3], and Greenspan et al. [4]. These studies discussed bioinformatic applications based on DL. The first two are limited to applications in genomic medicine and the latter to medical imaging. The DL is a relatively new field of artificial intelligence, which achieves good results in the areas of big data processing such as speech recognition, image recognition, text comprehension, translation, and genomics.

There are several contributions based on DL in the fields of medical imaging and genomic medicine. However, the DNA sequence classification issue has received little attention. For an in-depth study of DL in bioinformatics, we can consider the review study conducted by Seonwoo et al. [5]. In addition, several studies have been devoted to the utilization of CNNs and recurrent neural networks (RNNs) in the field of bioinformatics and DNA classification [6,7].

1.1 The CNNs

The classification task based on CNNs depends on several layers. Tab. 1 provides a list of the basic functions of a variety of CNN layers [5].

Rizzo et al. [8] presented a DNA classification approach that depends on a CNN, and the spectral representation of DNA sequences. From the results, they found that their approach provided similar and good results between 95% and 99% at each taxonomic level. Moreover, Rizzo et al. [1] suggested a novel algorithm that depends on CNNs with frequency chaos game representation (FCGR). The FCGR was utilized to convert the original DNA sequence to an image before feeding it to the CNN model. This method is considered as an expansion of the spectral representation that was reported to be efficient. This work is a continuation of the work of Rizzo et al. [1] for the classification of DNA sequences using a deep neural network, and chaos game representation, except for the addition of downsampling layers that can achieve the best trade-off between performance and time of processing, which is the main contribution of this work. The proposed approach is an improved form of the CNN to obtain better classification accuracy.

1.2 Data Reduction Step

A weakness of the convolutional layer performance is that it reports the exact position of features in the input. Slight shifts in the features located in the input image contribute to different feature maps. The pooling layer is used to resize the feature maps to overcome this problem. A simplified representation of the features observed in the input is the outcome of using a pooling layer. In practice, max-pooling works better than average pooling for computer vision fields such as image recognition [9]. We can handle this issue in signal processing by using downsampling methods such as 2D RP, two-Dimensional two-Directional Random Projection ($(2D)^2$ RP)) and 2D DT. As a result, a lower-resolution representation of an input signal is produced, including the significant structural components without fine details that might not be helpful. The important purpose of the RP is to reduce the high dimensionality and preserve the geometrical relationship in the dimensionality reduction.

Table 1: A list of the basic layers used in CNNs

CNN layers	Function
Convolutional Layer	• Feature extraction for the features, which have relative information to create the best possible representation of the input. • The process is a 2D convolution on the inputs. • The "dot products" between weights and inputs are integrated across channels. • The filter has the same number of layers as the input volume, and the output volume has the same depth as the number of filters. • It accepts a volume of size $W_1 \times H_1 \times D_1$ (size of the input image is width × height × number of channels • Four parameters are required to compute the output features. ✓ Number of filters K ✓ Spatial extent F ✓ Stride S ✓ Padding P • The output is of size $W_2 \times H_2 \times D_2$, where: $W_2 = \frac{W_1 - F + 2P}{S} + 1$ $H_2 = \frac{H_1 - F + 2P}{S} + 1$ $D_2 = K$
Pooling Layer	• With the introduction of convolution, the time complexity of learning increases. The issue with maps of the output features is that they are sensitive to the positions of features. Therefore, we use the pooling layer, which handles this sensitivity, reduces the number of parameters, and thus increases the speed of the algorithm. • The pooling layer depends on the non-linear downsampling of activation maps. • The two main methods associated with pooling are maximum and average pooling that measure the maximum and average values for each feature map patch, respectively.
Softmax Loss	• It is used for evaluating the cost function as follows: $S(y_i) = -\sum_{i=1}^{N} \log \frac{e^{\mathbf{w}_{y_i}^T \mathbf{f}_i + b_{y_i}}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T \mathbf{f}_i + b_j}}$ where $\mathbf{f}_i$ denotes the features and y_i is the true class label of the image. $\mathbf{W}_j$ and b_j are the weights and bias of the j^{th} class, respectively. N is the number of training samples and K is the number of classes.

Dimensionality reduction methods can be briefly categorized into two classes, namely subspace and feature selection. Subspace methods include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Random Projection (RP), etc. The RP can be free from training and much faster. Some extensions of one-dimensional RP (1D RP), including two-dimensional RP (2DRP) [10], two-directional two-dimensional RP $(2D)^2$ RP [11,12], sparse RP [13], require far lower computational complexity and storage cost than those of traditional 1D RP.

The authors of [13] used 2D schemes instead of 1D ones to reduce computational complexity and storage costs. In addition, in [10], the authors proposed $(2D)^2$ PSRP methods to generate 2D cancelable faces and palmprints. The authors in [12] showed that 1D cancelable palmprint codes verification performance cannot meet the requirements of accuracy, and their computational and storage costs are large. So, 1D cancelable palmprint codes are extended to 2D cancelable palmprint codes. Moreover, the authors in [11] proposed a novel method called $(2D)^2$ RP for feature extraction from biometrics, where they employed $(2D)^2$ RP and its variations on the face and palmprint databases.

Feature selection methods depend on different spectral transformations such as two-dimensional Discrete Cosine Transform (2D DCT), and two-dimensional Discrete Wavelet Transform (2D DWT) to extract the features to reduce the amount of data, thereby simplifying the subsequent classification problem, and hence decision-making. Adaptive selection/weighting of features/coefficients is typically used for dimensionality reduction and performance improvement. The features that achieve high discrimination [14], high accuracy [15], and low correlation [12] should be selected and provided with high weights. The number of selected features is less than that of the original features. Feature selection methods have several advantages compared with subspace methods, such as PCA. Sometimes, feature selection methods can be fast and training-free, while it is comparable to the subspace methods in terms of accuracy. Furthermore, the selected features maintain their original forms. So, it is easy to observe the true values of the features. The authors in [16] proposed a novel approach for face and palmprint recognition in the DCT domain. In addition, the utilization of fusion rules is also an important tool to reduce computational complexity and storage costs [17].

The rest of this paper is organized as follows. Section 2 presents the proposed CNN models based on different downsampling layers. The max-pooling, DT, and RP are explained in Sections 3–5, respectively. Section 6 introduces the dataset. The results and discussions are given in Section 7. Finally, Section 8 gives the concluding remarks.

2 The Proposed CNNs Based on Different Downsampling Layers

We designed the proposed architecture, inspired by Rizzo et al. [1] architecture that has been reported as an efficient architecture for bacteria classification. We have added one convolutional layer followed by DT or 2D RP or variations of $(2D)^2$ RP layer as compared to the original Rizzo et al. [1] architecture. Fig. 1 shows the proposed model. Firstly, the input DNA sequences are preprocessed using the FCGR algorithm with $k = 6$, 7, and 8. Thus, the output image is of dimension $b_0 = \sqrt{4^k} \times \sqrt{4^k}$. For more details about FCGR, see [1,18]. Then, the normalized output is processed to make the input images suitable to the proposed CNN. The proposed CNN model consists of seven layers. The first four layers (from l_1 to l_4) are convolutional layers, each followed by a max-pooling layer. Additionally, the layers l_5 to l_6 are convolutional layers followed by various downsampling layers, which are applied to reduce the dimensionality of training.

Several downsampling methods are implemented such as DT, 2D RP, and variations of $(2D)^2$ RP. Simulation parameters are specified in Tab. 2.

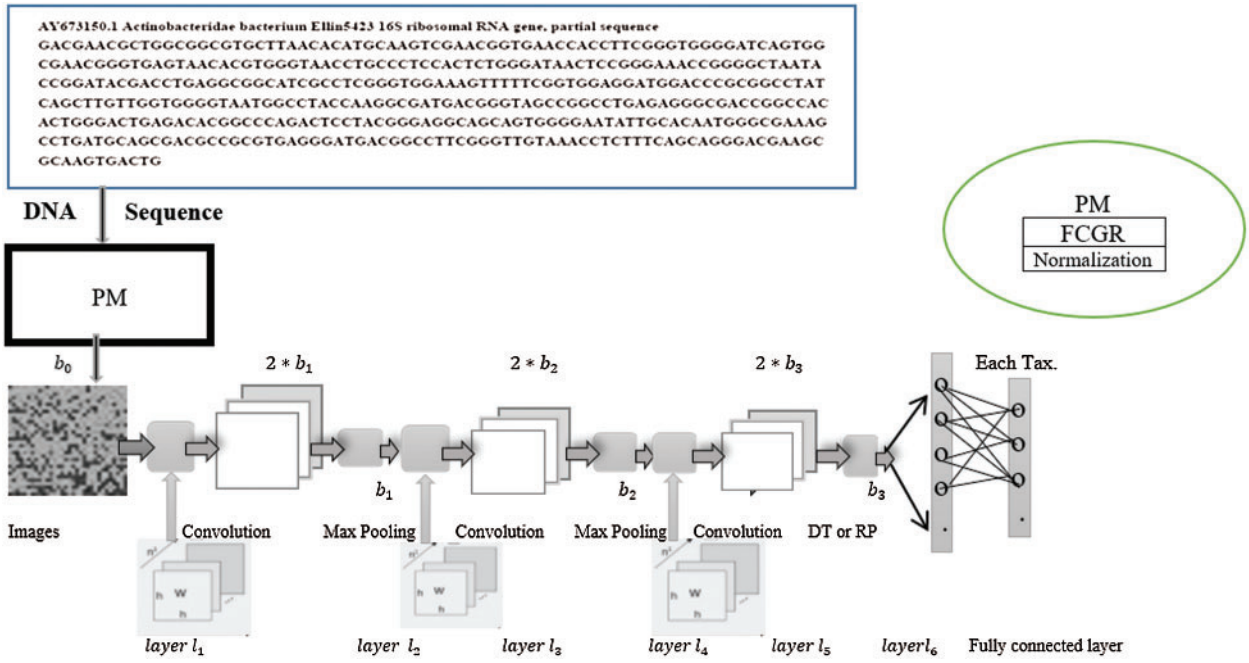

Figure 1: The architecture of the proposed model

Table 2: Simulation parameters

Layer type	Parameter
1 Conv	10, 5 × 5, padding 2
1 Relu Layer	–
2 Max Pooling	2 × 2, Stride 2
3 Conv	15, 5 × 5, padding 1
3 Relu Layer	–
4 Max Pooling	2 × 2, Stride 2
5 Conv	20, 5 × 5
5 Relu Layer	–
6 2D-RP or DCT or DWT or variation of $(2D)^2$ RP	–
7 Fully-connected layer	3 for phylum, 5 for class, 19 for order, 65 for family and 100 for the genus

After the convolutional layers, a set of the output images is generated; each of them of dimension $(2b_{i+1})$, and:

$$b_{i+1} = \frac{b_i - y + 1}{2} \tag{1}$$

For example, let $k = 6$. Hence, $b_0 = \sqrt{4^6} \times \sqrt{4^6} = 64 \times 64$, and the first convolutional layer (*layer* l_1) produces 20 output images of dimension (64 – 5 + 1) = 60. Then, the pooling layer is applied, which produces 20 output images of dimension 60/2 = 30. The proposed CNNs are trained for five different classification tasks, as illustrated in Fig. 2, and the simulation parameters are presented in Tab. 2.

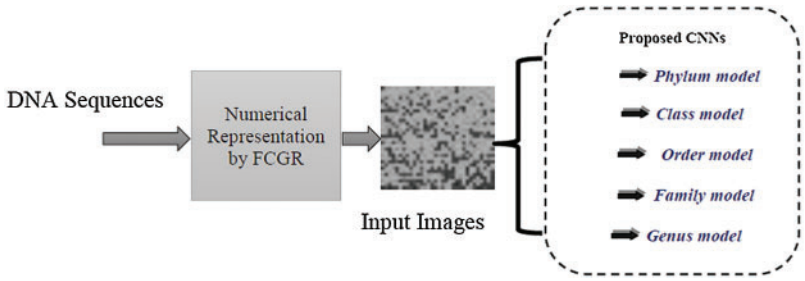

Figure 2: The architecture of the classifier

3 The Max-Pooling

The downsampling layer is another name for the pooling layer. It reduces the dimensionality of data, by dividing the input into rectangular pooling regions. The max-pooling computes the maximum of each region R_{ij} and consequently reduces the number of outputs. The max-pooling function is expressed as:

$$\mathbf{a}_{ij} = \max_{(p,q)\in \mathbf{R}_{ij}} (\mathbf{a}_{pq}) \tag{2}$$

while the average pooling function can be expressed as:

$$\mathbf{a}_{ij} = \frac{1}{|R_{ij}|} \sum\nolimits_{(p,q)\in \mathbf{R}_{ij}} \mathbf{a}_{pq} \tag{3}$$

where a_{pq} is the input at (p, q) within R_{ij} , and $|R_{ij}|$ is the size of the pooling region.

Let us examine the effect of the max-pooling, when a 4 × 4 matrix input image is used, as shown in Fig. 3b.

In the case of an irregular nature of DNA sequences, k-mers recognition, the effective downsampling layer increases the ability of the CNN to achieve high performance. Anyway, the classification results do not critically depend on the feature extraction stage, but strongly depend on how these features are reduced.

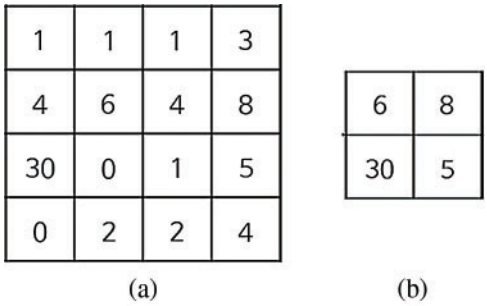

Figure 3: The pooling of a 4 × 4 matrix (a) The 4 × 4 matrix (b) The effect of max-pooling

4 Discrete Transform (DT)

Since the FCGR converts the DNA sequences into the form of images, we can apply the spectral transformations (Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT)) for downsampling, and the feature extraction stage for DNA images. The reason for applying these transformations emerges from their wide and effective use for extracting features, decorrelation, ordering, and dimensionality reduction purposes in the fields of speech, image, and bio-signal processing [19]. In signal processing, the DCT [20] can reveal the discriminative characteristics of the signal, namely, its frequency components. It is considered as a separable linear transformation. The basic idea of the DT is to select a certain sub-band after implementing the transformation. For example, the DCT can be implemented on the numerical sequence representing the DNA, and certain coefficients from the DCT can be selected to represent the whole sequence. The definition of the two-dimensional DCT for an input image A is given by:

$$B_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix} \tag{4}$$

where

$$\propto_p = \begin{cases} \dfrac{1}{\sqrt{M}}, & P = 0 \\ \dfrac{2}{\sqrt{M}}, & 1 \le p \le M-1 \end{cases} \tag{5}$$

and

$$\propto_q = \begin{cases} \dfrac{1}{\sqrt{N}}, & q = 0 \\ \dfrac{2}{\sqrt{N}}, & 1 \le q \le N-1 \end{cases} \tag{6}$$

while M and N are the row and column lengths of A, respectively.

The wavelet transform is faster and more efficient than the Fourier transform in capturing the essence of data [20]. Therefore, there is a growing interest in utilizing the wavelet transform

to analyze biological sequences. The DWT is investigated to predict the similarity accurately and reduce computation complexity compared to the DCT and the DFT techniques.

The wavelet transform has been a very novel method for analyzing and processing of non-stationary signals such as bio-signals in which both time- and frequency-domain information are required. The wavelet analysis is often used for compression and de-noising of signals without appreciable degradations. The wavelet transform can be used to analyze the sequences at different frequency bands. In 2D DWT, the image is decomposed into four sub-bands. After filtering, the signal is downsampling by 2. In this work, the DWT is employed to reduce the dimensionality of features by performing the single-level 2D wavelet decomposition. The decomposition is conducted using a particular wavelet filter. Then, approximation coefficients (LL) can be selected. For example, let the first convolutional layer (*layer* l_1) produce 20 output images of dimension $2b_1$. Then, a DWT pooling layer is applied, which produces 20 output images of dimension b_1. Fig. 4 displays an example of the proposed DWT pooling.

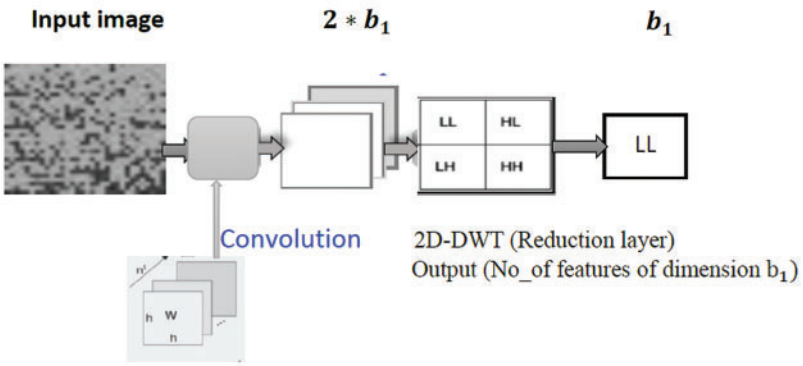

Figure 4: The proposed DWT pooling

5 Two-Dimensional Random Projection (2D-RP)

This method achieves the dimensionality reduction with low computational cost [21,22]. If the original dataset is represented by the matrix $\mathbf{X}_{d\times n}$, then the projection of the data onto a lower k-dimensional space gives $\mathbf{Y}_{\mathrm{k\times n}}$ or $\mathbf{Y}$ as follows:

$$\mathbf{Y} = \mathbf{Y}_{k\times n} = \mathbf{R}_{k\times d}.\mathbf{X}_{d\times n} \tag{7}$$

where $\mathbf{R}_{k\times d}$ is the RP matrix and $k \ll d$.

5.1 Implementation of RP

The following stages of the RP are written using Matlab 2018a:

- Set the input as the features map $\mathbf{X}_{m\times m}$ (the multilayer CNN features).
- Reshape the input to $\mathbf{X}_{d\times n}$.
- Create a $k \times d$ random matrix ($\mathbf{R}_{k\times d}$), where $k \ll d$.
- for $j = 1{:}n$
- $\mathbf{Y}(:, j) = \mathbf{R}\times\mathbf{X}(:, j)$;
- End for.
- Output=$\mathbf{Y}_{k\times n}$.

5.2 Two Directional Two Dimensional Random Projection $(2D)^2$ RP

The 2D RP can be implemented simultaneously in two directions, that is called $(2D)^2$ RP. In this method, the input matrix is projected at row direction and column direction as follows:

$$\mathbf{Y} = \mathbf{Y}_{k\times h} = \mathbf{R}_{k\times d}.\mathbf{X}_{d\times n}.\mathbf{C}_{n\times h} \tag{8}$$

where $\mathbf{R}$ and $\mathbf{C}$ are the left mapping matrix for column-direction and right mapping matrix for row-direction, respectively and $h \ll n$, $k \ll d$. The details of $(2D)^2$ RP were explained in [12]. With Eq. (8), the projection of data onto a lower k and h dimensional subspace is implemented.

5.3 Variations of $(2D)^2$ RP

The dimensionality reduction is the main purpose of pooling layers as introduced in the previous sections. In this work, the DWT and DCT are proposed to make the pooling layer to satisfy this purpose and add more details to feature maps. Hybrid methods that combine $(2D)^2$ RP with DWT or DCT have been proposed. These methods are namely $(2D)^2$ RP DWT and $(2D)^2$ RP DCT based on the matrices $\mathbf{R}$ and $\mathbf{C}$ as indicated in Tab. 3.

$$\mathbf{Y} = \mathbf{Y}_{k\times h} = \mathbf{R}^{RP}_{k\times d}.\mathbf{X}_{d\times n}.\mathbf{C}^{RP}_{n\times h} \tag{9}$$

$$\mathbf{Y} = \ \mathbf{Y}_{k\times h} = \mathbf{R}^{DWT}_{k\times d}.\mathbf{X}_{d\times n}.\mathbf{C}^{RP}_{n\times h} \tag{10}$$

$$\mathbf{Y} = \mathbf{Y}_{k\times h} = \mathbf{R}^{DCT}_{k\times d}.\mathbf{X}_{d\times n}.\mathbf{C}^{RP}_{n\times h} \tag{11}$$

Table 3: Variations of $(2D)^2$ RP

Variations	Right mapping matrix (R)	Left mapping matrix (C)
$(2D)^2$ RP	RP	RP
$(2D)^2$ RP DCT	DCT	RP
$(2D)^2$ RP DWT	DWT	RP

6 Dataset Descriptions

Data were obtained from the Ribosomal Database Project (RDP) [23], Release 11. A file in the FASTA record was obtained from the repository, which includes data on 1423984 outstanding bacterial gene sequences. For each bacterium, we have data on which taxonomic categories belong to certain genetic sequences. In addition, we have information on the phylum, class, order, family, and genus of a given 16S rRNA gene sequence. The bacterial genome contains the small-subunit ribosomal RNA transcript and is useful as a general genetic marker. It is often used to determine bacterial diversity, identification, and genetic similarity, and it is the basis for molecular taxonomy [24]. Two different sequences were used for comparison; (a) full-length sequences with a length of approximately 1200 – 1500 nucleotides and (b) 500 bp DNA sequence fragments. The total set of data includes sequences of the 16SrRNA gene of bacteria belonging to 3 different phylum, 5 different classes, 19 different orders, 65 different families, and 100 different genera, as shown in Tab. 4.

Table 4: 16S Bacteria dataset composition

Dataset	Number of sequences	Labels	Training set (70%)	Test set (30%)
Phylum	300	3	210	90
Class	500	5	350	150
Order	1900	19	1330	570
Family	6500	65	4550	1950
Genus	10000	100	7000	3000

7 Results and Discussions

One of the key parameters that affect the DNA classification based on CNN is avoiding dimensionally problem and the sensitivity to the positions of the features. Even though the complex nature of DNA sequences is improved by convolutional layers, it is still necessary to ensure that the multi-layer CNN feature map has as suitable dimensions as possible. Therefore, there is a bad need to provide a downsampling layer that improves the generation ability of the original features. In this work, the CNN is utilized as a choice for deep learning, FCGR is applied for data preprocessing method, and different types of downsampling layers are introduced, such as DCT, DWT, 2D RP, $(2D)^2$ RP, $(2D)^2$ RP DCT, and $(2D)^2$ RP DWT. A comparison is presented for the performance of CNN based on different downsampling layers. Finally, a random search method is applied to optimize the hyperparameters.

7.1 Comparison between Different Types of Downsampling Based on CNN

The effectiveness of different downsampling layers has been investigated to classify bacterial sequences to reach the highest possible accuracy. First, the given DNA sequences have been mapped using the FCGR algorithm with $k = 6$, 7, and 8. Then, the proposed CNN models based on different downsampling layers have been trained for each taxon. These models are.

- Model_1 (Max-CNN): Rizzo paper [1].
- Model_2 (RP-CNN): CNN classification followed by max-pooling or 2D RP.
- Model_3 (DWT-CNN): CNN classification followed by max-pooling or DWT.
- Model_4 (DCT-CNN): CNN classification followed by max-pooling or DCT.
- Model_5 ($(2D)^2$ZRP-CNN): CNN classification followed by max-pooling or $(2D)^2$ RP.
- Model_6 ($(2D)^2$ RP DCT-CNN): CNN classification followed by max-pooling or $(2D)^2$ RP DCT.
- Model_7($(2D)^2$ ZRP DWT-CNN): CNN classification followed by max-pooling or $(2D)^2$ RP DWT.

To demonstrate the effectiveness of the proposed models, two simulation experiments are conducted. In the first case, the efficiency of the prediction for each taxonomic level is measured separately by taking into account the whole bacteria sequence. In the second case, instead of the whole sequence, we consider only the 500 bp long sequences. The simulation results are demonstrated in Tabs. 5–7, and Fig. 5 introduces the experimental results for the full-length DNA sequences, while Tabs. 8–10 and Fig. 6 present the results for 500 bp-length sequences. The classification is obtained for the same sequence with the representation of images at different values of k. From these tables and figures, it is clear that the proposed CNN model based on DWT and $(2D)^2$ RP DWT always achieves the best performance. Furthermore, the $(2D)^2$ RP DWT-CNN model consumes less running time. The best choice for mapping is at $k = 8$, because

it improves the accuracy and F-score compared with those achieved at $k = 6$ and 7. Moreover, the proposed CNN based on $(2D)^2$ RP DWT has a processing time that is less than that of the max-CNN by about 135 sec on average. From the mentioned results, the proposed $(2D)^2$ RP DWT-CNN model with k equal to 8 provides superior results compared with other models.

Table 5: Comparison of accuracy scores between created models based on different pooling layers considering full length at $k = 6$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	1	0.9980	0.9825	0.9600	0.9230
RP-CNN	1	0.9990	0.9875	0.9725	0.9600
DCT-CNN	1	1	0.9856	0.9703	0.9635
DWT-CNN	1	1	0.9933	0.9733	0.9705
$(2D)^2$ RP DCT-CNN	1	1	0.9856	0.9725	0.97
$(2D)^2$ RP-CNN	1	1	0.9833	0.9725	0.9715
$(2D)^2$ RP DWT-CNN	1	1	0.9933	0.9733	0.972

Table 6: Comparison of accuracy scores between created models based on different pooling layers considering full length at $k = 7$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	1	0.9980	0.9825	0.9615	0.9241
RP-CNN	1	0.9990	0.9875	0.9732	0.9638
DCT-CNN	1	1	0.9856	0.9706	0.9668
DWT-CNN	1	1	0.9933	0.9738	0.9715
$(2D)^2$ RP DCT-CNN	1	1	0.9933	0.9724	0.9694
$(2D)^2$ RP-CNN	1	1	0.9933	0.9754	0.9729
$(2D)^2$ RP DWT-CNN	1	1	0.9933	0.9774	0.9736

Table 7: Comparison of accuracy scores between created models based on different pooling layers considering full length at $k = 8$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	1	0.9980	0.9825	0.9715	0.9338
RP-CNN	1	0.9990	0.9875	0.9767	0.9683
DCT-CNN	1	1	0.9856	0.9783	0.9694
DWT-CNN	1	1	0.9933	0.9835	0.9774
$(2D)^2$ RP DCT-CNN	1	1	O.993	0.9805	0.9767
$(2D)^2$ RP-CNN	1	1	0.9933	0.9829	0.9774
$(2D)^2$ RP DWT-CNN	1	1	0.9933	0.9838	0.9794

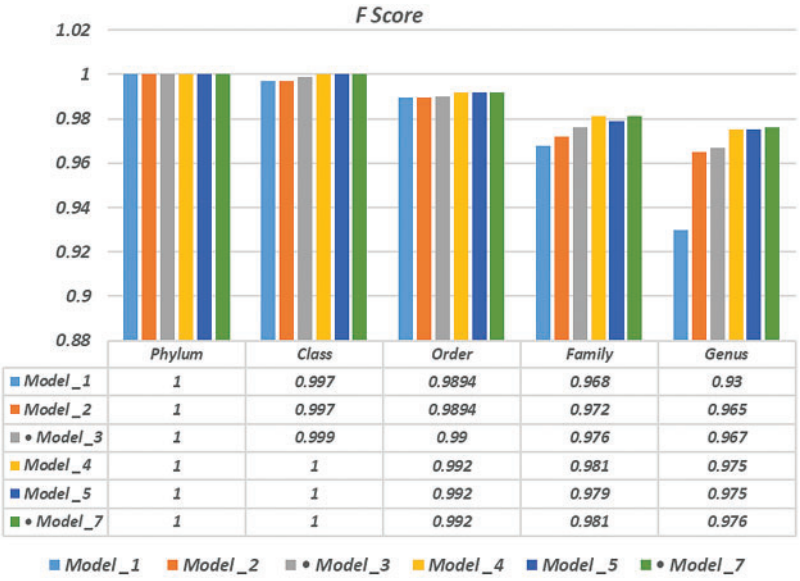

Figure 5: F-scores of the proposed model at $k = 8$, for the full length case

Table 8: Comparison of accuracy scores between created models based on different pooling layers for 500 bp-length sequences at $k = 6$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	0.9955	0.9930	0.8960	0.8130	0.7533
RP-CNN	0.9960	0.9950	0.9322	0.8356	0.8100
DCT-CNN	0.9975	0.9960	0.9455	0.8363	0.8138
DWT-CNN	0.9975	0.9960	0.9455	0.8470	0.8250
$(2D)^2$ RP DCT-CNN	0.9975	0.9960	0.9456	0.8394	0.8167
$(2D)^2$ RP-CNN	0.9975	0.9960	0.9468	0.8450	0.8136
$(2D)^2$ RP DWT-CNN	0.9975	0.9960	0.9468	0.8470	0.8250

Table 9: Comparison of accuracy scores between created models based on different pooling layers for 500 bp-length sequences at $k = 7$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	0.9955	0.9930	0.8960	0.8150	0.7554
RP-CNN	0.9960	0.9950	0.9322	0.8359	0.8115
DCT-CNN	0.9975	0.9960	0.9455	0.8371	0.8158
DWT-CNN	0.9975	0.9960	0.9455	0.8494	0.8267
$(2D)^2$ RP DCT-CNN	0.9975	0.9960	0.9456	0.8394	0.8167
$(2D)^2$ RP-CNN	0.9975	0.9960	0.9468	0.8456	0.8156
$(2D)^2$ RP DWT-CNN	0.9975	0.9960	0.9468	0.8494	0.8268

Table 10: Comparison of accuracy scores between created models based on different pooling layers for 500 bp-length sequences at $k = 8$

Model	Phylum	Class	Order	Family	Genus
Max-CNN [1]	0.9955	0.9930	0.8960	0.8159	0.7567
RP-CNN	0.9960	0.9950	0.9322	0.8368	0.8134
DCT-CNN	0.9975	0.9960	0.9455	0.8386	0.8158
DWT-CNN	0.9975	0.9960	0.9455	0.85	0.83
$(2D)^2$ RP DCT-CNN	0.9975	0.9960	0.9455	0.8405	0.8238
$(2D)^2$ RP-CNN	0.9975	0.9960	0.9468	0.8471	0.8194
$(2D)^2$ RP DWT-CNN	0.9975	0.9960	0.9468	0.85	0.8305

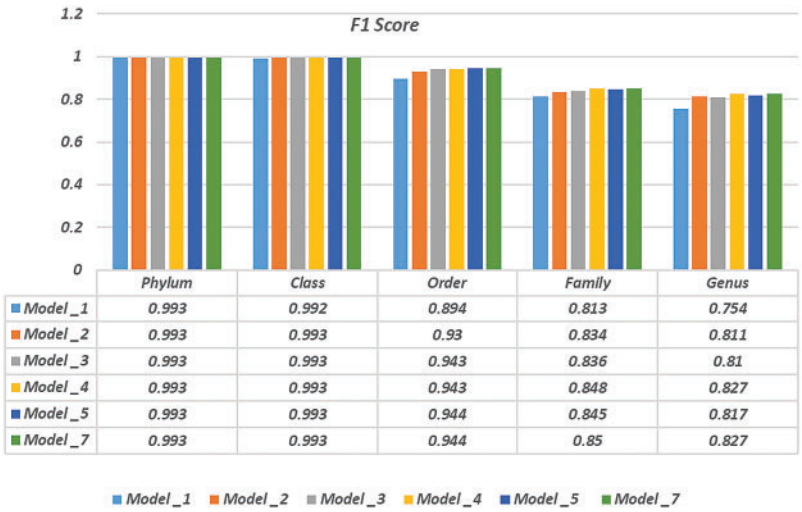

Figure 6: F-scores of the proposed model at $k = 8$, for 500 bp-length sequences

Tabs. 11 and 12 present comparisons between the performance of the proposed $(2D)^2$ RP DWT-CNN and the state-of-the-art models; VGG16, VGG19, and ResNet-50 at $k = 8$ and different DNA sequences using the full-length and 500 bp-length sequences, respectively. The results indicate that the proposed $(2D)^2$ RP DWT-CNN achieves better accuracies at the genus level, by about 4.23% and 7.34% compared to the VGG16 model for the full-length and 500 bp-length sequences, respectively. The proposed model consumes 53 min, which is the lowest computational time compared to the VGG16, VGG19, and ResNet-50. For VGG16, VGG19, and ResNet-50, the computational times were recorded as 62, 87, 134 min, and also they have lower accuracies of classification. Finally, a comparison is conducted among the proposed ($(2D)^2$ RP DWT-CNN model and the mentioned state-of-the-art models based on different datasets for the three most popular taxonomic trees (RDP, SILVA, and green genes) [24].

Table 11: Comparison of the proposed $(2D)^2$ RP DWT-CNN and the state-of-the-art CNNs for Genus level at $k = 8$ and full-length sequences

CNN	Accuracy	Specificity	Precision	Recall	F1-score
VGG16	0.9371	0.9383	0.916	0.9319	0.9238
VGG19	0.9587	0.9595	0.9385	0.9554	0.9468
ResNet-50	0.9658	0.9667	0.9465	0.9613	0.9538
$(2D)^2$ RP DWT-CNN	0.9794	0.9806	0.9636	0.9691	0.976

Table 12: Comparison of the proposed $(2D)^2$ RP DWT-CNN and the state-of-the-art CNNs for Genus level at $k = 8$ and 500 bp-length sequences

CNN	Accuracy	Specificity	Precision	Recall	F1-score
VGG16	0.7571	0.755	0.7383	0.7413	0.7554
VGG19	0.7838	0.781	0.766	0.7738	0.7813
ResNet-50	0.8267	0.8285	0.795	0.8167	0.8238
$(2D)^2$ RP DWT-CNN	0.8305	0.8313	0.811	0.8294	0.827

Tab. 13 indicates the different datasets used for the full-length implementation. Tab. 14 summarizes the experimental results for the proposed model and the state-of-the-art models. It is shown that the proposed model is superior, and it achieves a classification accuracy equal to 97.94% against 97.14%, 96.27%, and 96.27% for RDP 11, SILVA dataset [25], and greengenes dataset [26], respectively.

Table 13: The input datasets for the full-length implementation

Dataset	Number of sequences	Labels	Training set (70%)	Test set (30%)
RDP 11 [23]	10000	100	7000	3000
SILVA [25]	5000	100	3500	1500
Greengenes [26]	2000	100	1400	600

Table 14: Comparison results between the proposed $(2D)^2$ RP DWT-CNN and the state-of-the-art CNNs for different datasets considering the full-length implementation

Dataset	CNN	Accuracy	Specificity	Precision	Recall	F1-score
RDP 11 [23]	VGG16	0.9371	0.9383	0.916	0.9319	0.9238
	VGG19	0.9587	0.9595	0.9385	0.9554	0.9468
	ResNet-50	0.9658	0.9667	0.9465	0.9613	0.9538
	$(2D)^2$ RP DWT-CNN	0.9794	0.9806	0.9636	0.9691	0.976
SILVA [25]	VGG16	0.9361	0.9368	0.9177	0.9288	0.9232
	VGG19	0.9487	0.9493	0.9275	0.9367	0.932
	ResNet-50	0.9589	0.9596	0.9387	0.9517	0.9451
	$(2D)^2$ RP DWT-CNN	0.9714	0.9723	0.9493	0.9623	0.9557
Greengenes [26]	VGG16	0.9262	0.9276	0.9097	0.9214	0.9155
	VGG19	0.9388	0.9397	0.9217	0.9336	0.9276
	ResNet-50	0.9458	0.9467	0.9285	0.9412	0.9348
	$(2D)^2$ RP DWT-CNN	0.9627	0.9638	0.9457	0.9576	0.9516

7.2 Hyperparameter Tuning

The training process may be quite difficult due to the enormous number of initial variables called hyperparameters. These values are defined before the start of the learning process. Some examples of hyperparameters include the learning rate, the minibatch size, and the number of epochs. In this paper, some changes in hyperparameters are applied to iteratively configure and train the proposed model. This section can be divided into subsections as follows:

7.2.1 Learning Rate Results

In this subsection, the effect of the learning rate on the CNNs with different downsampling layers at the genus level is investigated in the case of full-length and 500 bp-length sequences. These downsampling layers include Max-CNN, RP-CNN, DCT-CNN, DWT-CNN, $(2D)^2$ RP DCT-CNN, $(2D)^2$ RP-CNN and $(2D)^2$ RP DWT-CNN. The parameters used in the simulation are mini-batch with 64, and the number of epochs for training is equal to 20. The comparison among the mentioned models at different learning rates is shown in Tabs. 15–18 for the full-length sequences.

Table 15: CNN metrics with different downsampling layers at learning rate = 0.01 considering full-length implementation at the genus level

Learning rate	Model	Measures			
		Accuracy	Precision	Recall	F1-Score
0.01	Max-CNN	0.3650	0.3470	0.3745	0.3602
	RP-CNN	0.3680	0.3433	0.3775	0.3597
	DCT-CNN	0.3833	0.3665	0.3967	0.3810
	DWT-CNN	0.3880	0.3665	0.3960	0.3820
	$(2D)^2$ RP DCT-CNN	03843	0.3665	0.3963	0.3880
	$(2D)^2$ RP-CNN	0.3840	0.3633	0.3945	0.3882
	$(2D)^2$ RP DWT-CNN	0.3856	0.3650	0.3967	0.3894

Table 16: CNN metrics with different downsampling layers at the learning rate = 0.001

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.001	Max-CNN	0.8180	0.7945	0.8267	0.8102
	RP-CNN	0.8280	0.7933	0.8345	0.8133
	DCT-CNN	0.8480	0.8233	0.8467	0.8348
	DWT-CNN	0.8580	0.8313	0.8565	0.8433
	$(2D)^2$ RP DCT-CNN	0.8544	0.8335	0.8633	0.8470
	$(2D)^2$ RP-CNN	0.8540	0.8302	0.8645	0.8470
	$(2D)^2$ RP DWT-CNN	0.8580	0.8313	0.8565	0.8467

Table 17: CNN metrics with different downsampling layers at a learning rate = 0.0001

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.0001	Max-CNN	0.9338	0.9448	0.9389	0.9333
	RP-CNN	0.9683	0.9529	0.9562	0.9504
	DCT-CNN	0.9694	0.9529	0.9643	0.9585
	DWT-CNN	0.9774	0.9593	0.9714	0.9653
	$(2D)^2$ RP DCT-CNN	0.9767	0.9533	0.96	0.9653
	$(2D)^2$ RP-CNN	0.9774	0.9575	0.9691	0.9632
	$(2D)^2$ RP DWT-CNN	0.9794	0.9626	0.9748	0.976

Table 18: CNN metrics with different downsampling layers at a learning rate = 0.00001

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.00001	Max-CNN	0.9338	0.9448	0.9389	0.9333
	RP-CNN	0.9683	0.9529	0.9562	0.9504
	DCT-CNN	0.9694	0.9529	0.9643	0.9585
	DWT-CNN	0.9774	0.9593	0.9714	0.9653
	$(2D)^2$ RP DCT-CNN	0.9767	0.9533	0.96	0.9653
	$(2D)^2$ RP-CNN	0.9774	0.9575	0.9691	0.9632
	$(2D)^2$ RP DWT-CNN	0.9794	0.9626	0.9748	0.976

It can be noted that the highest accuracy is obtained at the learning rate equal to 0.0001 and 0.00001, but processing time increases, where 0.0001 learning rate has a processing time less than that of the 0.00001 learning rate. The same comparison is conducted for 500 bp-length sequences to trust the achieved results as demonstrated in Tabs. 19–21. Therefore, at a 0.0001 learning rate, superior accuracy for the training set can be attained for any length of the DNA sequences.

Table 19: CNN metrics with different downsampling layers at a learning rate = 0.01

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.01	Max-CNN	0.1633	0.1670	0.1645	0.1602
	RP-CNN	0.1652	0.1633	0.1675	0.1697
	DCT-CNN	0.1763	0.1765	0.1767	0.1710
	DWT-CNN	0.1783	0.1765	0.1763	0.1780
	$(2D)^2$ RP –CNN	01773	0.1765	0.1760	0.1720
	$(2D)^2$ RP DWT-CNN	0.1840	0.1833	0.1845	0.1882

Table 20: CNN metrics with different downsampling layers at a learning rate = 0.001

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.001	Max-CNN	0.6045	0.6176	0.5835	0.60
	RP-CNN	0.6045	0.6153	0.5845	0.60
	DCT-CNN	0.6073	0.6303	0.6050	0.6035
	DWT-CNN	0.6238	0.6333	0.6073	0.6253
	$(2D)^2$ RP –CNN	0.6203	0.6315	0.6053	0.6205
	$(2D)^2$ RP DWT-CNN	0.6233	0.6345	0.6073	0.6253

Table 21: CNN metrics with different downsampling layers at a learning rate = 0.0001

Learning rate	Classifiers	Measures			
		Accuracy	Precision	Recall	F1-Score
0.0001	Max-CNN	0.7567	0.7433	0.7573	0.7503
	RP-CNN	0.8134	0.8045	0.8173	0.81
	DCT-CNN	0.8158	0.8013	0.8153	0.8173
	DWT-CNN	0.83	0.8113	0.8253	0.8203
	$(2D)^2$ RP –CNN	0.8194	0.8105	0.8253	0.8153
	$(2D)^2$ RP DWT-CNN	0.8305	0.8145	0.8273	0.8233

7.2.2 Mini-batch Size and Number of Epochs

In this subsection, the evaluation using different mini-batch sizes is investigated in the training process against different iterations for the proposed $(2D)^2$ RP DWT-CNN model (at genus level considering full-length implementation) with the number of epochs = 20 and the learning rate equal to 0.0001. The experimental results are illustrated in Fig. 7. It is clear at mini-batch size equal to 128, the proposed $(2D)^2$ RP DWT-CNN achieved less accuracy performance, while at mini-batch sizes equal to 32 and 64, the proposed model has a better trade-off between the accuracy score and the processing time.

From the mentioned results, we can conclude that the best performance of the proposed DWT-CNN model is achieved at the learning rate equal to 0.0001 and the mini-batch size equal to 64. We can select a suitable number of epochs considering these values. Fig. 8 reveals the training progress of $(2D)^2$ RP DWT-CNN model at k equal to 6 considering the full-length implementation at a different numbers of epochs. It can be observed that best accuracy is obtained at 20 epochs. Finally, after several experiments, we give the best hyperparameters in Tab. 22.

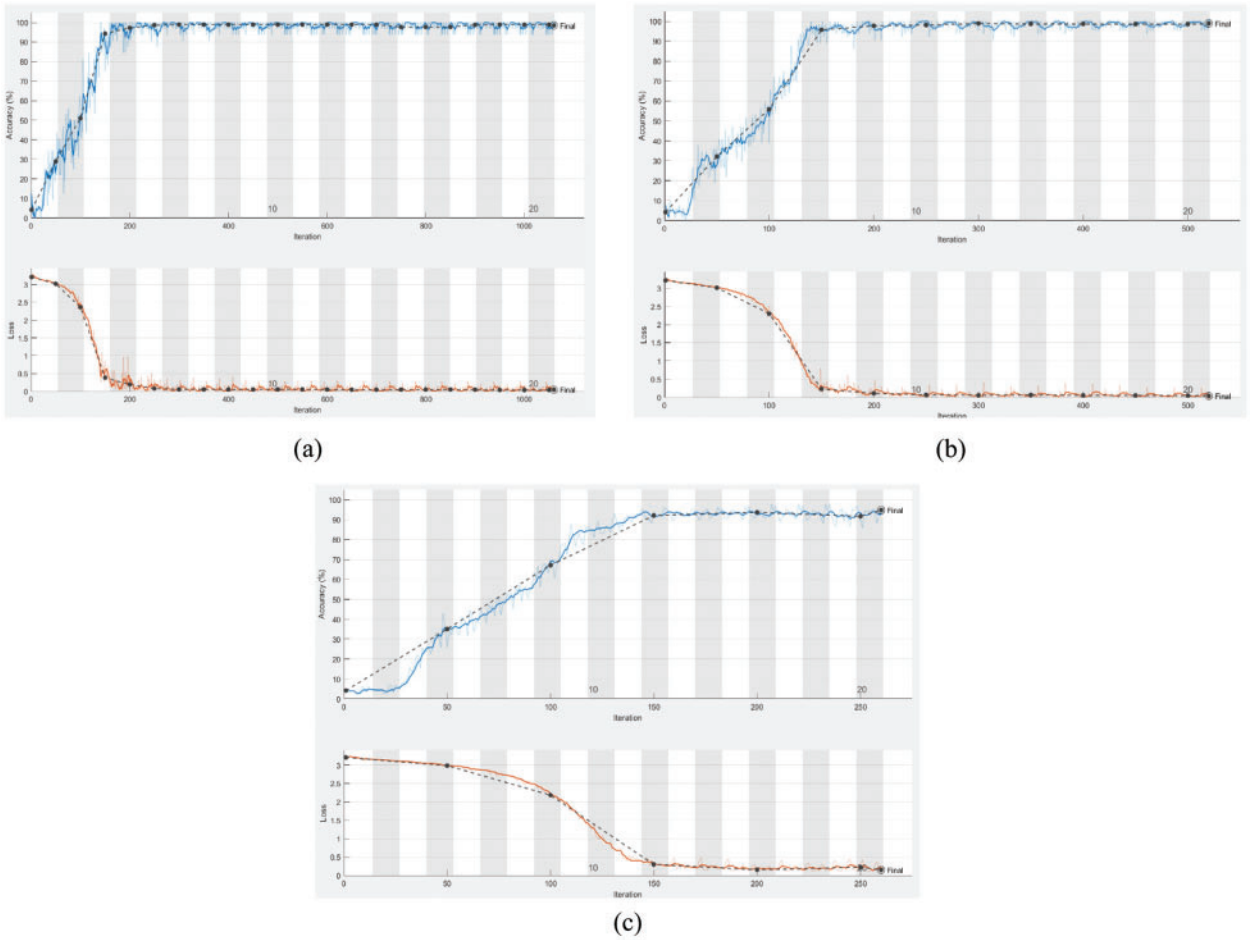

Figure 7: Training progress of $(2D)^2$ RP DWT-CNN model (k = 6) considering full-length implementation at different mini-batch sizes (a) 32, (b) 64, and (c) 128

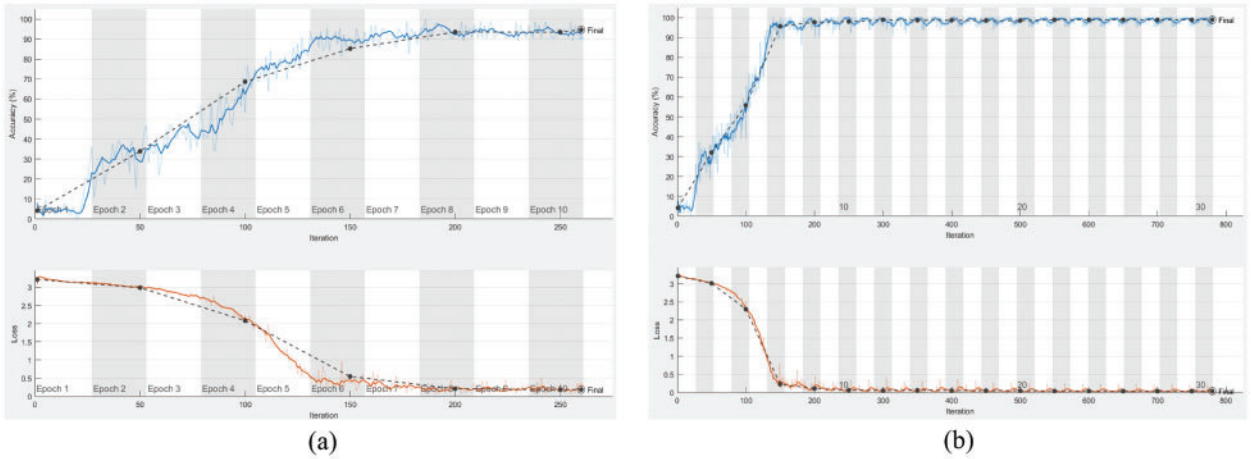

Figure 8: Training progress of $(2D)^2$ RP DWT-CNN model (k = 6) considering full-length implementation at different numbers of epochs (a) 10 and (b) 30

Table 22: The best hyperparameters used

Activation functions	Rectified linear units (ReLU)
Updater	Stochastic Gradient Descent with Momentum (SGDM)
Learning rate	0.0001
Backpropagation method	Mini-batch gradient descent
Mini-batch	64
Loss function	Cross-entropy
Number of epochs	20

8 Conclusions and Future Research Directions

This paper presented two contributions to the bacterial classification of DNA sequences. The first one is represented in the proposed models for bacterial classification using an improved CNN. In these models, the 2D RP, $(2D)^2$ RP, $(2D)^2$ RP DCT, $(2D)^2$ RP DWT, and DT methods are applied to reduce the dimensionality of the feature maps, while preserving the structure information. The proposed models make the data reduction process faster and more reliable. The simulation results revealed that selecting the appropriate downsampling layer with the training CNN could greatly influence the accuracy with an optimized computational time. According to the obtained results, it can be concluded that the CNN based on $(2D)^2$ RP DWT gives a high accuracy. Furthermore, this model can achieve a good trade-off between the accuracy score and the processing time for a suitable size of the frequency k-lengthen words in DNA sequences. Finally, the experimental results on different datasets reveal that the proposed $(2D)^2$ RP DWT model outperforms the state-of-the-art CNNs models. The second contribution lies in evaluating the effectiveness of the hyperparameters through the created CNNs based on different downsampling layers to select the best results. It is possible to say that the best accuracy is provided by using $(2D)^2$ RP DWT as a downsampling layer with $k = 6$. This study confirms that with a learning rate equal to 0.0001, the mini-batch size equal to 64, and the number of epochs equal to 20 are suitable to achieve the best performance on the given DNA dataset. For future work, the performance of different frequency-domain transforms for DNA classification can be investigated. In addition, deep CNN models developed from scratch can be designed to improve the DNA classification efficiency.

Acknowledgement: The authors would like to thank the support of the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University.

Funding Statement: This research was funded by the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University through the Fast-track Research Funding Program.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Rizzo, A. Fiannaca, M. Rosa and A. Urso, "Classification experiments of DNA sequences by using a deep neural network and chaos game representation," in *Proc. Int. Conf. on Computer Systems and Technologies*, Palermo, Italy, pp. 222–228, 2016.

[2] M. Leung, A. Delong, B. Alipanahi and B. Frey, "Machine learning in genomic medicine: A review of computational problems and data sets," *Proceedings of the IEEE,* vol. 104, no. 1, pp. 176–197, 2016.
[3] P. Mamoshina, A. Vieira, E. Putin and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics,* vol. 13, no. 5, pp. 1445–1454, 2016.
[4] H. Greenspan, B. Ginneken and R. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging,* vol. 35, no. 5, pp. 1153–1159, 2016.
[5] M. Seonwoo, L. Byunghan and Y. Sungroh, "Deep learning in bioinformatics," *Briefings in Bioinformatics,* vol. 18, no. 5, pp. 851–869, 2017.
[6] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436–444, 2015.
[7] G. Bosco and M. Gangi, "Deep learning architectures for DNA sequence classification," in *Proc. 11th Int. Workshop of Fuzzy Logic and Soft Computing Applications*, Naples, Italy, pp. 162–171, 2017.
[8] R. Rizzo, A. Fiannaca, M. Rosa and A. Urso, "A deep learning approach to DNA sequence classification," in *Proc. Int. Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, Springer, Cham, vol. 9874, pp. 129–140, 2016.
[9] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*. The MIT Press, 2016.
[10] L. Leng, S. Zhang, X. Bi and M. Khan, "Two-dimensional cancelable biometric schemes," in *Proc. of Int. Conf. on Wavelet Analysis and Pattern Recognition*, Xi'an, China, pp. 164–169, 2012.
[11] A. Alarifi, M. Amoon, M. Aly and W. El-Shafai, "Optical PTFT asymmetric cryptosystem-based secure and efficient cancelable biometric recognition system," *IEEE Access,* vol. 8, pp. 221246–221268, 2020.
[12] L. Leng and J. Zhang, "Palmhash code vs. palmPhasor code," *Neurocomputing,* vol. 108, pp. 1–12, 2013.
[13] L. Leng and J. Zhang, "Palmhash code for palmprint verification and protection," in *Proc. of 25th IEEE Canadian Conf. on Electrical and Computer Engineering*, Montreal, QC, Canada, pp. 1–4, 2012.
[14] L. Leng, M. Li, C. Kim and X. Bi, "Dual-source discrimination power analysis for multi-instance contactless palmprint recognition," *Multimedia Tools and Applications,* vol. 76, pp. 333–354, 2017.
[15] S. Chakraborty and V. Gupta, "DWT based cancer identification using EIIP," in *Proc. of IEEE Second Int. Conf. on Computational Intelligence & Communication Technology*, Ghaziabad, India, pp. 718–723, 2016.
[16] L. Leng, J. Zhang, M. Khan, X. Chen and K. Alghathba, "Dynamic weighted discrimination power analysis: A novel approach for face and palmprint recognition in the DCT domain," *International Journal of Physical Sciences,* vol. 5, no. 17, pp. 467–471, 2010.
[17] L. Leng, M. Li and A. Teoh, "Conjugate 2DpalmHash code for secure palm-print-vein verification," in *Proc. of 6th IEEE Int. Congress on Image and Signal Processing*, Hangzhou, China, pp. 1705–1710, 2013.
[18] Y. Wang, K. Hill, S. Singh and L. Kari, "The spectrum of genomic signatures: From dinucleotides to chaos game representation," *Gene,* vol. 346, pp. 173–185, 2005.
[19] Y. Liu, "Wavelet feature selection for microarray data," in *Proc. IEEE/NIH Life Science Systems and Applications Workshop*, Bethesda, MD, USA, pp. 205–208, 2007.
[20] A. Tsonis and P. Kumar, "Wavelet analysis of DNA sequences," *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics,* vol. 53, no. 2, pp. 1828–1834, 1996.
[21] R. Wu, S. Yang, D. Leng and Z. Luo, "Random projected convolutional feature for scene text recognition," in *Proc. IEEE 15th Int. Conf. on Frontiers in Handwriting Recognition*, Shenzhen, China, pp. 132–137, 2016.
[22] P. Wojcik and M. Kurdziel, "Training neural networks on high-dimensional data using random projection," *Pattern Analysis and Applications,* vol. 22, no. 3, pp. 1221–1231, 2019.
[23] rRNA sequences, [Online]. Available: https://rdp.cme.msu.edu, (Access date 11 January 2021).
[24] M. Balvočiūtė and D. Huson, "SILVA, RDP, greengenes, NCBI, and OTT—How do these taxonomies compare?," *BMC Genomics,* vol. 18, no. 114, pp. 1–8, 2017.

[25] P. Yilmaz, L. Parfrey, P. Yarza, J. Gerken, E. Pruesse *et al.*, "The silva and all-species living tree project (ltp) taxonomic frameworks," *Nucleic Acids Research,* vol. 42*,* no. D1*,* pp. D643–D648, 2014.

[26] D. McDonald, M. Price, J. Goodrich, E. Nawrocki, T. DeSantis *et al.*, "An improved green genes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea," *ISME Journal,* vol. 6*,* no. 3*,* pp. 1–8, 2012.