Research Article

Academic Platform Journal of Engineering and Smart Systems (APJESS) 13(2), 61–70, 2025

https://dergipark.org.tr/en/pub/apjess

Received: 17-03-2025, Accepted: 20-05-2025

https://doi.org/10.21541/apjess.1659716

pPromoter-FCGR: Deep Learning on Frequency Chaos Game Representation for Prediction of DNA Promoters

[*1] Gülbahar Merve ŞILBIR

[1] Distance Education Application and Research Center, Trabzon University, Türkiye, gmervecakmak@trabzon.edu.tr

Abstract

A promoter is defined as a DNA sequence that helps to initiate transcription by binding to RNA polymerase. It has a key role in various biological processes, such as gene expression, adaptation and disease development. Promoter identification methods used to be conventional wet-lab approaches, but these can be laborious and costly, so computational methods are now being used instead. In this study, DNA sequences were converted into RGB images using the Frequency Chaos Game Representation method for k-mer values of 5 and 6, and various CNN models were employed to classify promoters and non-promoters. Pretrained models such as ResNet-50, VGG16, and GoogleNet were utilized alongside a custom 17-layer CNN model with optimized hyperparameters. The ResNet-50 model achieved an accuracy of 82% and an AUC of 0.89, while the VGG16 model attained an accuracy of 80% and an AUC of 0.88. The GoogleNet model yielded an accuracy of 74% with an AUC of 0.82. However, the classification performance was observed to be lower compared to existing literature. The proposed 17-layer CNN model demonstrated improved performance, achieving an accuracy of 83% and an AUC of 0.90. The proposed CNN model outperformed pretraned models in promoter prediction.

Keywords: Classification, Deep Learning, Frequency Chaos Game Representation, Pre-training CNN Models, Promoter

1. INTRODUCTION

Sequencing projects aimed at mapping the human genome and determining its primary structure have significantly contributed to understanding genetic material. However, further research is needed to elucidate the mechanisms controlling gene expression. Promoters, which play a crucial role in gene transcription, are essential for understanding genetic regulation. As key elements that initiate gene transcription, promoters control the precise timing, location, and level of gene expression [1].

Promoters are specific deoxyribonucleic acid (DNA) sequences that facilitate ribonucleic acid (RNA) polymerase binding and initiate the transcription process. They are fundamental in regulating gene expression [2], determining gene activity and protein production according to cell type [3], responding to environmental changes and cellular adaptation [4], and influencing genetic mutations and diseases [5] [6] [7]. Therefore, understanding and characterizing promoters is essential for advancing genetic regulation and gene therapy approaches.

Promoters are nucleotide sequences located at the transcription start site. Promoter regions are typically several hundred nucleotides long and determine the location where transcription begins as well as the direction in which transcription will proceed on the DNA. The promoter region of most genes contains various regulatory elements such as TATA, CAAT, and GC boxes. In protein-coding genes, the region where RNA polymerase II binds is known as the TATA box. Positioned approximately 25-30 base pairs upstream from the transcription start site, this box typically consists of a 7-8 base pair sequence formed by AT base pairs, often flanked by GC-rich sequences.

The promoter region is comprised of three distinct regions: the core promoter, the proximal promoter, and the distal promoter. The core promoter is located 35 base pairs upstream from the transcription start site and represents the smallest region of the promoter necessary for initiating transcription. The proximal promoter is a region extending several hundred base pairs upstream from the transcription start site and contains regulatory elements. The distal promoter is a region located several thousand base pairs upstream from the transcription start site, containing various regulatory elements. The promoter regions are illustrated in Figure 1.

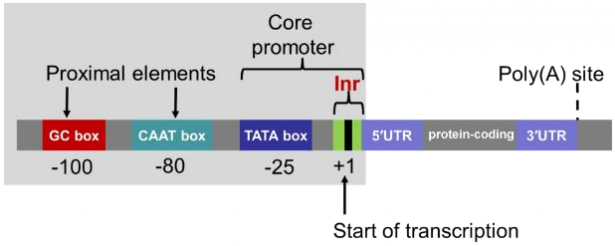


Figure 1. The promoter elements [8]

The identification and classification of promoters, which play a crucial role in the development of genetic modification and gene therapy approaches, are carried out through wet-lab experiments [9], [10]. However, a review of the literature indicates that these experiments are time-consuming and costly, presenting a significant disadvantage [11]. To overcome this, computational methods are employed. Literature studies have reported the development of models that predict promoters using artificial intelligence-based techniques, such as machine learning and deep learning. Escherichia coli (E. coli) is a popular model organism in genetics and molecular biology. It's easy to culture, grows quickly, and its genome is well-studied. E. coli also illustrates basic biological processes. The utilization of Escherichia coli in analyses such as promoter region studies is justified by its numerous advantages. Notably, this organism is employed in studies examining genetic regulation and the development of machine learning–based prediction models [11-15].

Xiao et al. (2019) developed a two-layer support vector machine (SVM) model, where the first layer determines whether a DNA sequence is a promoter, and the second layer predicts the strength of the promoter [11]. In a study by Oubounyt et al. (2019), a model combining convolutional neural networks (CNN) and long short-term memory (LSTM) was developed for promoter prediction, which contributed to improving the promoter identification problem [12]. In the promoter prediction and activity classification study by Le et al. (2022), a BERT model based on natural language processing (NLP) was used for encoding, and feature selection was performed using Shap analysis [13]. Subsequently, promoter classification and activity prediction were conducted using different machine learning classifiers. Li et al. (2024) fine-tuned the DNABERT model, based on natural language processing, for promoter prediction, achieving high performance in predicting promoters but moderate performance in predicting promoter strength [14]. Peng et al. (2024) developed a pre-trained NLP model for predicting E. coli promoters and successfully predicted promoters using various deep learning techniques [15].

In the studies mentioned above, models developed for promoter prediction utilize traditional machine learning methods, deep learning techniques, and natural language processing methods. These studies show that various approaches were used for constructing the feature vector of the DNA sequence, including physicochemical properties of nucleotides [11], one-hot encoding [12], and encoding structures from pre-trained natural language processing models [13]–[15]. However, there is no study in the literature that investigates the transformation of DNA sequences into images and uses deep learning methods for predicting promoter classes. In models developed for exon and intron classification [16] and protein function prediction [17], [18], it has been observed that image representation of sequences and the development of prediction models using deep learning positively contribute to prediction performance. Building on this, we aim to introduce a methodological innovation in the literature of promoter prediction by proposing a model to generate the feature vector of DNA sequences through an image representation. Therefore, in this study, we focus on developing a prediction model to determine whether a DNA sequence is a promoter by converting it into an image.

2. MATERIALS AND METHODS

2.1. Benchmark Dataset

In this study, we aimed to classify whether a DNA sequence is a promoter and categorize the promoter strength as either strong or weak. For this purpose, the dataset used in the training phase consists of E. coli K-12 genome sequence data obtained from the RegulonDB database [19]. To compare with studies in the literature on promoter prediction, the dataset obtained from RegulonDB, which was also used in the study titled iPSW(2L)-PseKNC by Xiao et al. (2019), was included in this research [11]. The promoter DNA sequences in the dataset, ranging from 100 to 1000 base pairs (bp) in length, were divided into 81 bp core promoter sequences based on nucleosome and linker DNA biological characteristics. Non-promoter DNA sequences were created by selecting random regions of 81 bp in length from non-promoter regions. Sequences with a similarity of over 85% were removed using CD-HIT. As a result, 3382 promoter sequences and 3382 non-promoter sequences were obtained (Table 1).

Table 1. Dataset details and train-test set split

	Promoter		Non-promoter	Total
	Strong	Weak		
Dataset	1591	1791	3382	6764
Train	1273	1433	2706	5412
Test	318	358	676	1352

As seen in Table 1, the dataset was randomly split with a ratio of 0.8 for the train set and 0.2 for the test set. Accordingly, the number of promoter class examples in the train set was determined to be 2,706, and the number of non-promoter class examples was also set to 2,706. For the test set, the number of promoter class examples was 676, and the number of non-promoter class examples was 676. Care was taken to ensure that no class imbalance occurred in the distribution of data between the train and test sets. The selection of samples from the promoter and non-promoter classes was conducted to ensure a balanced class distribution, with an equal number of samples from each class to ensure the validity of the study.

2.2. Frequency Chaos Game Representation (FCGR)

With the emergence of fractal geometry by Mandelbrot in the 1970s, fractal generation algorithms began to be developed. One such algorithm, the chaos game algorithm, was designed to generate fractals from random inputs [20]. This algorithm is also used in encoding DNA sequences [21]. The chaos game representation (CGR) takes DNA sequences as inputs, and the outputs are numerical matrices or images that mathematically represent these sequences. Each nucleotide in a DNA sequence is represented numerically at a unique coordinate in a two-dimensional space. The representation of a DNA sequence in CGR is shown in Figure 2.

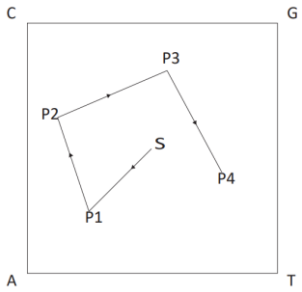
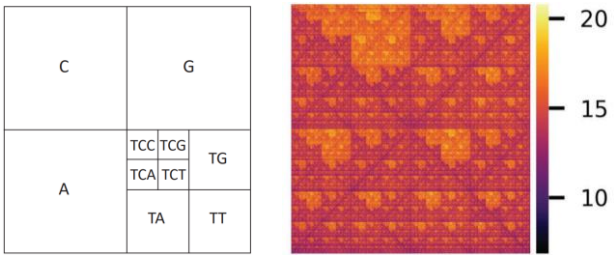


Figure 2. The process of determining the four pixels in the CGR for the nucleotides in a DNA sequence [22]

As seen in Figure 2, the nucleotides of DNA-adenine (A), cytosine (C), guanine (G), and thymine (T)-are placed in the four corners of a square. The CGR coordinates are centered at (0, 0), with nucleotides positioned at (-1, -1) and (1, 1) coordinates. As shown in Figure 3, depending on the orientation of the corners, different visual patterns emerge for genomes. Creating motifs based on nucleotide concentrations of a DNA sequence using CGR enables the visualization of the DNA.



(a) **(b)**

Figure 3.a. Placement of DNA nucleotides onto the coordinate plane using CGR and segmentation through iterative processes [22], **b.** Formation of fractal patterns from DNA sequences (H. sapiens) using FCGR [22]

With the Frequency Chaos Game Representation (FCGR) based on CGR, DNA sequences are defined in terms of kmers, and a matrix is created by visualizing the frequency of kmers in a pre-defined order. This image representation method helps identify patterns and similarities in genetic information. The number of quadrants in an FCGR grid is calculated as 4^k, where k defines the kmer size. In this case, the FCGR matrix size becomes $2^k \times 2^k$ (Figure 3.a). It is observed that random DNA sequences do not form meaningful patterns, whereas applying FCGR to DNA sequences reveals fractal patterns (Figure 3.b). Since FCGR allows the creation of a distinctive image of DNA, in this study, the FCGR-generated images of promoter and non-promoter sequences will be used as inputs in the promoter classification model.

2.3. Pre-Trained Models

Pre-trained models are trained on large datasets to solve a specific problem and consist of extensive networks. By utilizing the starting points of these models, they can be adapted and developed as prediction models for solving different problems. In this study, we will attempt to predict whether DNA sequences generated using FCGR are promoters using commonly used models in the literature, such as ResNet50, VGG-16, and GoogleNet.

2.3.1. Resnet50 Architecture

Residual Network (ResNet) is a 50-layer CNN architecture. Each layer contains the same number of filters, independent of the feature map. Additionally, when the feature map size is halved, each layer has twice as many filters. In the three-layer stack of the ResNet50 architecture, the training of each layer is accelerated. The ResNet50 architecture is illustrated in Figure 4.

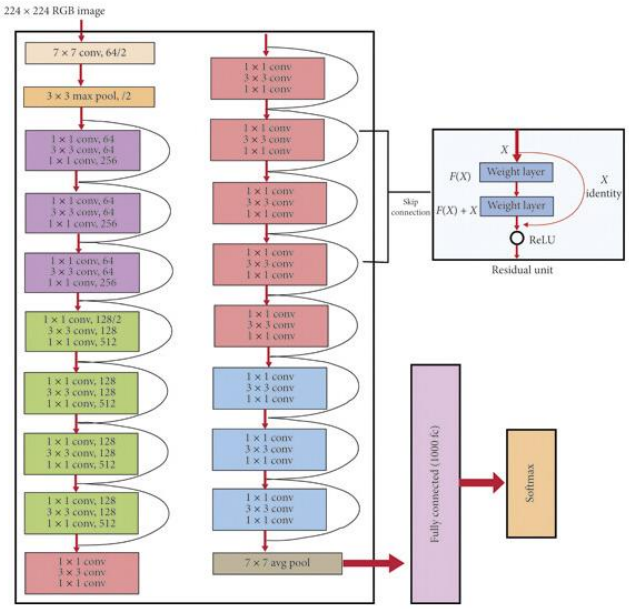


Figure 4. Resnet50 architecture [23]

As shown in Figure 4, the curved lines indicate that the input from the previous layer is transferred to the next layer. The model begins with a 7 x 7 convolutional layer with 64 kernels, followed by a 3 x 3 max pooling layer. The layers shown in different colors represent identical layers. ResNet50 consists of 23.521 million parameters.

2.3.2. VGG-16 Architecture

VGG-16 is a CNN architecture consisting of 13 convolutional layers and 3 fully connected layers. All convolutional layers have 3x3 filters, and all pooling layers are 2x2. After each pooling layer, the feature map is halved. The default input size is considered to be 224x224 with a

depth of 3 (RGB). The three fully connected layers are expanded to vectors of 25,088, 4,096, and 4,096, respectively. The VGG-16 architecture is illustrated in Figure 5.

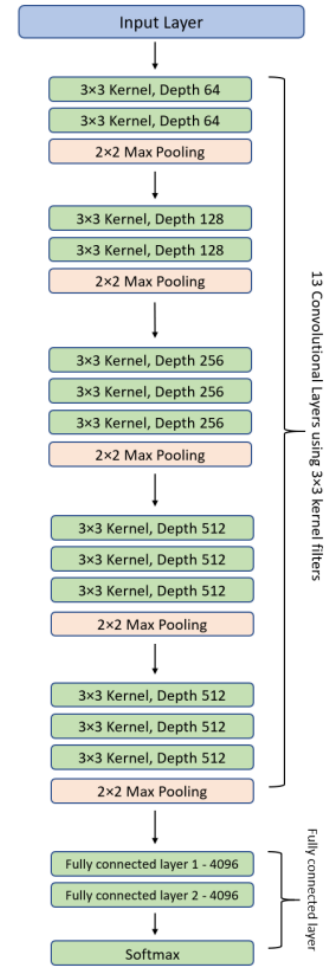


Figure 5. VGG-16 architecture [24]

2.3.3. GoogleNet (InceptionV3) Architecture

GoogleNet architecture is a CNN model consisting of 4 convolutional layers. The default input size is considered to be 224x224 with a depth of 3 (RGB). It includes 1x1, 3x3, and 5x5 convolutional sublayers, along with 3x3 max pooling. These starting points process the data from the previous layer in parallel. The use of parallel processing in this architecture helps address the issue of overfitting. GoogleNet architecture is illustrated in Figure 6.

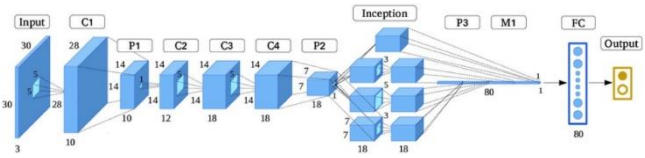


Figure 6. GoogleNet architecture [16]

2.4. Convolutional Neural Networks

Convolutional neural networks (CNNs), a prominent deep learning approach, are widely utilized in computer vision due to their remarkable performance. These networks employ convolutional operations between layers to identify and learn features from input data. By integrating pooling operations, CNNs effectively reduce data dimensionality, enhancing feature extraction and accelerating the learning process. The CNN architecture implemented in this study is a modified version of the LeNet-5 model [25]. The CNNs architecture utilized in this study is illustrated in Figure 7.

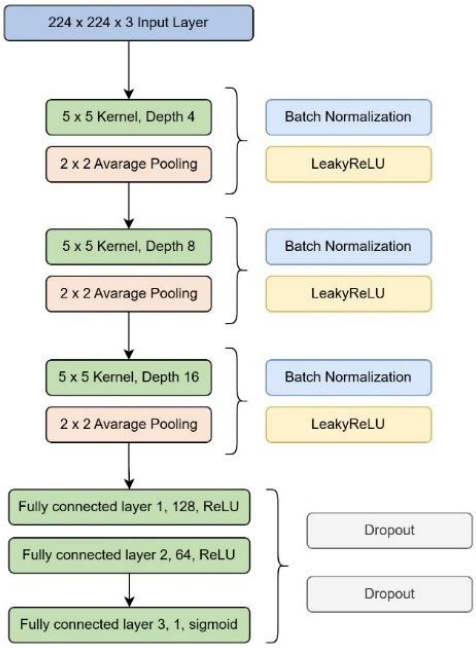


Figure 7. The CNNs architecture

The prediction model incorporates convolutional the 17 layers with 4, 8, and 16 filters, employing Batch Normalization, Leaky ReLU activation and 2x2 Avarage Pooling throughout. Outputs are flattened and fed into fully connected layers comprising 128 and 64 neurons with ReLU, followed by a sigmoid-activated binary classification layer. To mitigate the risk of overfitting, two dropout layers (2 x 0.4) have been incorporated. Binary cross-entropy is utilized as the loss function, optimized through the ADAM algorithm. Finally, promoter sequences are transformed into two-dimensional vectors to enhance feature learning.

2.5. Performance Evaluation Metrics

In this study, the performance evaluation metrics used for promoter prediction are accuracy (ACC), sensitivity (Sn), specificity (Sp), Matthews correlation coefficient (MCC), and area under the receiver operating characteristic curve (AUC).

$$ACC = \frac{TP+TN}{TP+FN+TN+FP} \quad (1)$$

$$Sn = \frac{TP}{TP+FN} \quad (2)$$

$$Sp = \frac{TN}{TN+FP} \quad (3)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \quad (4)$$

In the equations above, TP refers to true positives, TN refers to true negatives, FN refers to false negatives, and FP refers to false positives. Sn indicates the proportion of correctly

identified positive samples, while Sp represents the proportion of correctly identified negative samples. ACC denotes the overall accuracy, reflecting the percentage of correctly classified samples. MCC evaluates the correlation between actual and predicted values, ranging from -1 to 1. To comprehensively assess model performance, it is also essential to consider the area under the receiver operating characteristic (ROC) curve, which illustrates the relationship between true positives and false positives. The AUC value varies from 0 to 1, with higher values signifying superior predictive performance. Generally, higher values across these five metrics indicate better model performance.

3. RESULTS

In this section, the classification results for the ResNet-50, VGG16, GoogleNet model and the CNN model that has been proposed will be exposed. Initially, the promoter sequences were converted into images using the Frequency Chaos Game Representation (FCGR) method. The images used as input for the model in this study were resized to dimensions of 224 × 224 × 3. Figures 8 and 9 show the images generated for the training and test datasets using k-mer values of 5 and 6. In the following figures, a label value of 0 represents the non-promoter class, while a label value of 1 corresponds to the promoter class.

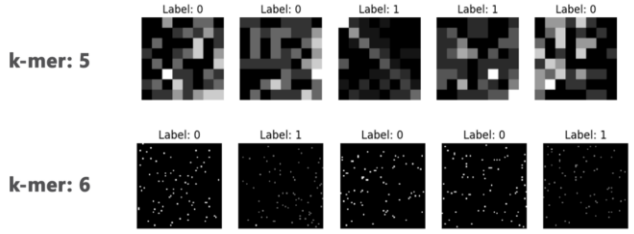


Figure 8. FCGR representation of the training dataset (k-mer 5 and 6)

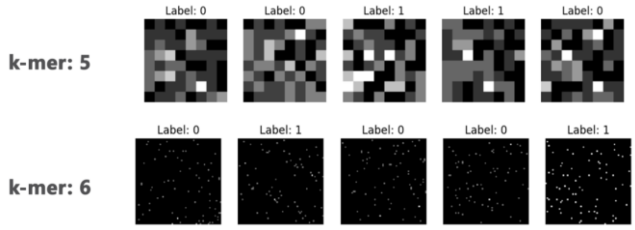


Figure 9. FCGR representation of the test dataset (k-mer 5 and 6)

3.1. Performance Outcomes of the ResNet-50 Architecture in Classification Tasks

The ResNet-50 architecture, a pre-trained CNN with 50 layers initially designed for classifying images into 1,000 categories, was adapted for binary classification in this study. The convolutional layers' weights were frozen to retain their learned feature representations. A fully connected layer with a single neuron and a sigmoid activation function was added for probabilistic output interpretation. The dataset was split into subsets, with 80% used for training and 20% for evaluation. Training was conducted over 30 epochs using the binary cross-entropy loss function and the Adam optimizer (learning rate: 0.001) to balance stability and convergence. The ResNet-50 model was trained using batch sizes of 8, 16, 32, and 64 on promoter sequences converted into images via FCGR with k-mer values of 5 and 6. This configuration minimized overfitting while maintaining computational efficiency. Hyperparameters were kept at default values unless otherwise specified, following standard transfer learning practices.

Table 2 presents the accuracy (Acc), sensitivity (Sn), specificity (Sp), area under the curve (AUC), and Matthews correlation coefficient (MCC) values for different batch sizes and k-mer values classified by the ResNet-50 model.

Table 2. Classification results of the Resnet-50 model

k-mer	Batch Size	Acc	Sn	Sp	AUC	MCC
5	64	0.67	**0.86**	0.47	0.82	0.36
	32	0.80	0.81	0.80	0.88	0.60
	16	0.75	0.56	**0.95**	0.87	0.56
	8	**0.82**	0.80	0.84	**0.89**	**0.64**
6	64	0.79	0.75	0.82	0.87	0.57
	32	0.75	0.77	0.73	0.81	0.50
	16	0.67	0.51	0.83	0.79	0.37
	8	0.80	0.80	0.79	0.86	0.59

The following results were yielded by our image classification using the ResNet-50 model: In the k-mer 5 representation, the highest accuracy (Acc) of 0.82, area under the curve (AUC) of 0.89, and Matthews correlation coefficient (MCC) of 0.64 were achieved with a batch size of 8. The highest sensitivity (Sn) value of 0.86 was observed in the model trained with a batch size of 64, while the highest specificity (Sp) value of 0.95 was found in the model trained with a batch size of 16. Figure 10 presents the predicted values of the ResNet-50 model displayed on the confusion matrix. With k-mer 5 and a batch size of 8, the ResNet-50 model was able to predict 80% of promoters and 84% of non-promoters, achieving an accuracy (Acc) of 0.82 and an area under the curve (AUC) of 0.89.

3.2. Performance Outcomes of the VGG16 Architecture in Classification Tasks

The VGG16 architecture, a pre-trained CNN with 16 layers initially designed for classifying images into 1,000 categories. To adapt the VGG16 model for our binary classification task, the weights of the layers were frozen, and a fully connected layer with a single neuron and a sigmoid activation function was appended. The dataset was divided into 80% for training and 20% for testing, and the model was trained over 30 epochs. Training was performed with batch sizes of 8, 16, 32, and 64 on promoter sequences converted into images using the FCGR method with k-mer values of 5 and 6.

Table 3 presents the accuracy (Acc), sensitivity (Sn), specificity (Sp), area under the curve (AUC), and Matthews correlation coefficient (MCC) values for different batch sizes and k-mer values classified by the VGG16 model.

Table 3. Classification results of VGG16 model

k-mer	Batch Size	Acc	Sn	Sp	AUC	MCC
5	64	**0.80**	0.78	**0.83**	**0.88**	**0.61**
	32	0.77	0.75	0.78	0.84	0.53
	16	0.76	0.68	0.85	0.82	0.55
	8	0.78	0.76	0.80	0.85	0.57
6	64	0.74	**0.82**	0.66	0.82	0.48
	32	0.76	0.66	0.86	0.84	0.53
	16	0.73	0.53	0.92	0.85	0.50
	8	0.78	0.77	0.80	0.85	0.57

The following results were yielded by our image classification using the VGG16 model: In the k-mer 5 representation, the highest accuracy (Acc) of 0.80, Sp of 0.83, area under the curve (AUC) of 0.88, and Matthews correlation coefficient (MCC) of 0.61 were achieved with a batch size of 64. k-mer 6 representation, the highest sensitivity (Sn) value of 0.82 was observed in the model trained with a batch size of 64. Figure 11 presents the predicted values of the VGG16 model displayed on the confusion matrix. With k-mer 5 and a batch size of 64, the VGG16 model was able to predict 78% of promoters and 83% of non-promoters, achieving an accuracy (Acc) of 0.80 and an area under the curve (AUC) of 0.88.

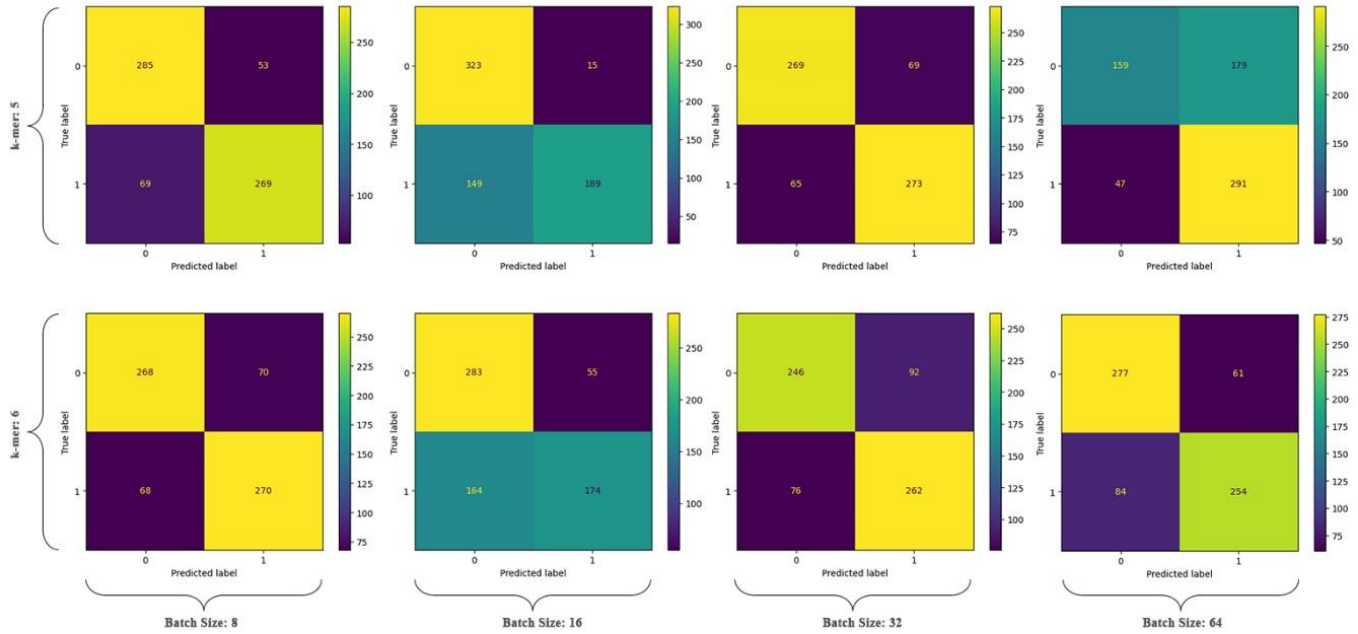


Figure 10. Confusion matrices for the Resnet-50 model

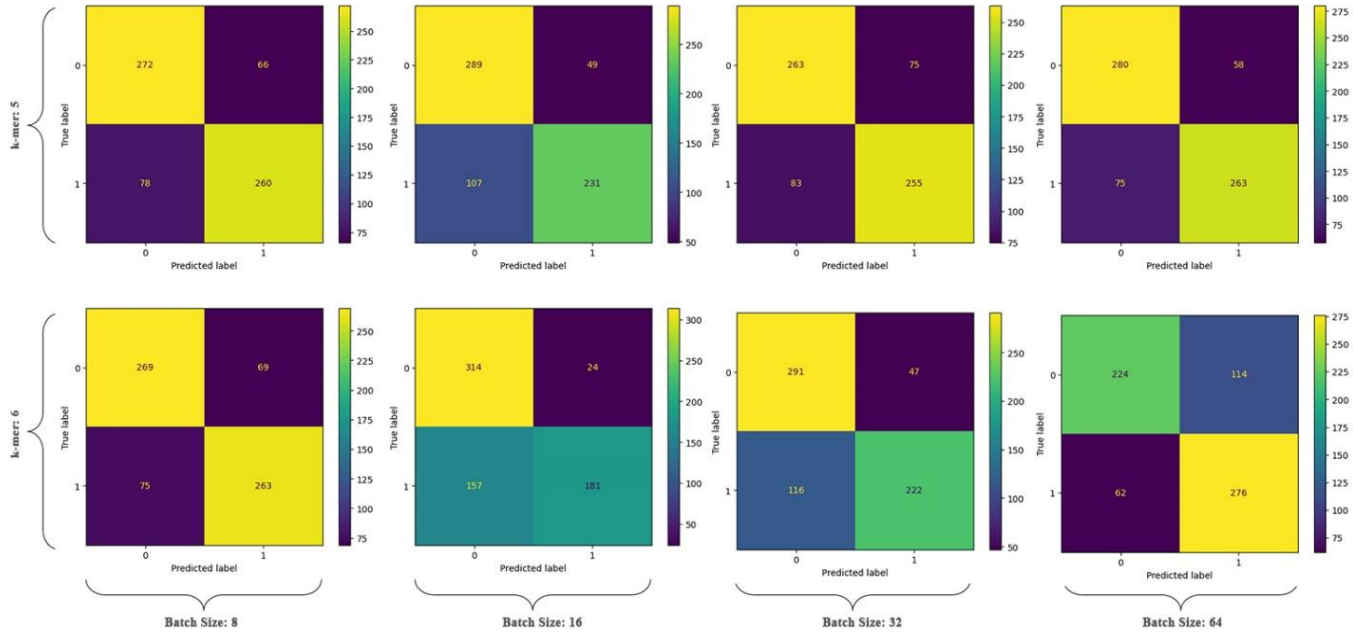


Figure 11. Confusion matrices for the VGG16 model

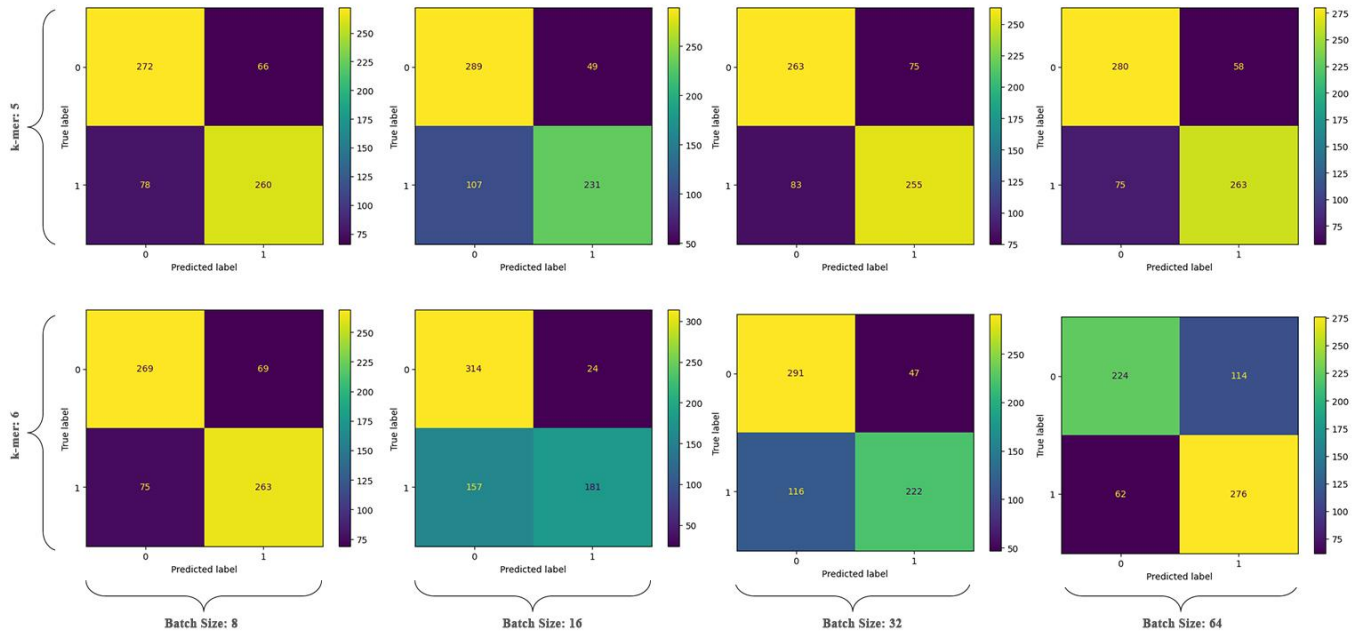


Figure 12. Confusion matrices for the GoogleNet model

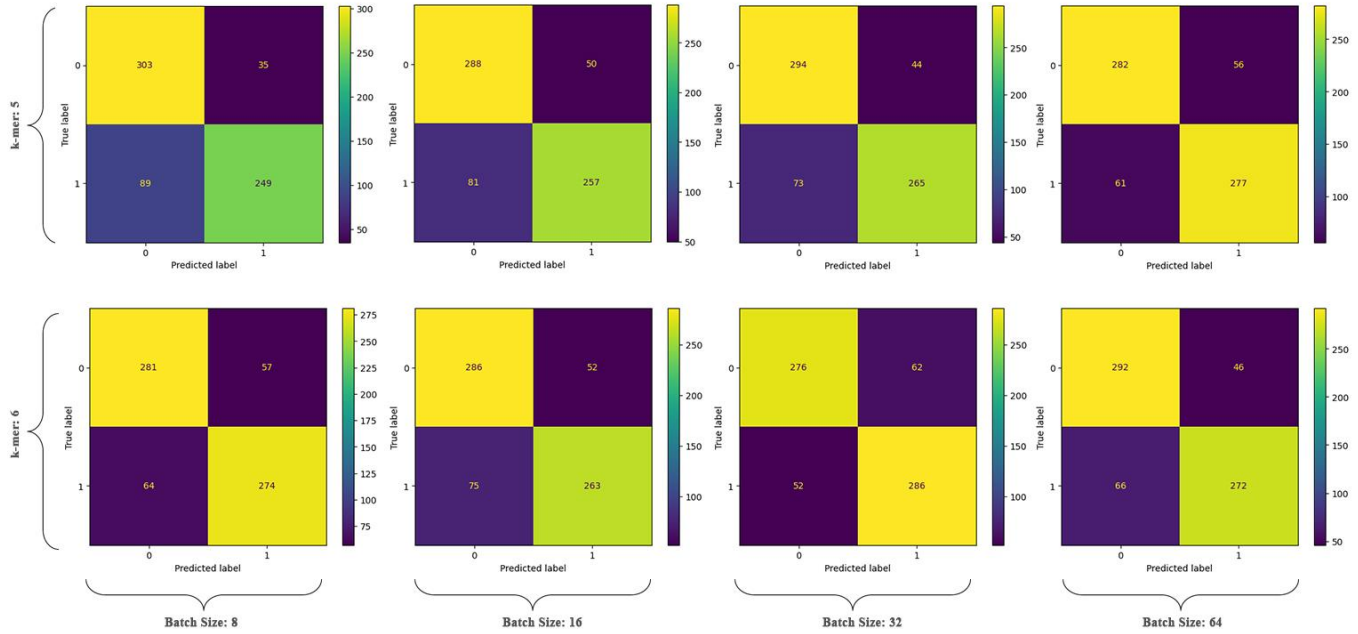


Figure 13. Confusion matrices for our proposed model

3.3. Performance Outcomes of the GoogleNet Architecture in Classification Tasks

The GoogleNet model is a pre-trained convolutional neural network with a depth of 22 layers, capable of classifying up to 1,000 object image classes. To adapt the GoogleNet model to our classification problem, the weights of layers were frozen, and a fully connected layer with a single neuron and a sigmoid activation function was added at the end to address the binary classification task. The dataset was split into 80% for training and 20% for testing, and the model was trained for 30 epochs. The GoogleNet model was trained using batch sizes of 8, 16, 32, and 64 on promoter sequences converted into images via FCGR with k-mer values of 5 and 6.

Table 4 presents the accuracy (Acc), sensitivity (Sn), specificity (Sp), area under the curve (AUC), and Matthews correlation coefficient (MCC) values for different batch sizes and k-mer values classified by the GoogleNet model.

The following results were yielded by our image classification using the GoogleNet model: In the k-mer 6 representation, the highest accuracy (Acc) of 0.74, area under the curve (AUC) of 0.82, and Matthews correlation coefficient (MCC) of 0.49 were achieved with a batch size of 16. The highest sensitivity (Sn) value of 0.92 was observed in the model trained with a batch size of 8, while the highest specificity (Sp) value of 0.95 was found in the model trained with a batch size of 64. Figure 12 presents the

predicted values of the GoogleNet model displayed on the confusion matrix. With k-mer 6 and a batch size of 16, the GoogleNet model was able to predict 74% of promoters and 76% of non-promoters, achieving an accuracy (Acc) of 0.74 and an area under the curve (AUC) of 0.82.

Table 4. Classification results of GoogleNet model

k-mer	Batch Size	Acc	Sn	Sp	AUC	MCC
5	64	0.73	0.60	0.86	0.81	0.47
	32	0.73	0.64	0.81	0.79	0.45
	16	0.72	0.58	0.87	0.79	0.47
	8	0.71	0.62	0.80	0.79	0.42
6	64	0.66	0.36	**0.95**	0.80	0.38
	32	0.73	0.59	0.87	0.81	0.48
	16	**0.74**	0.74	0.76	**0.82**	**0.49**
	8	0.66	**0.92**	0.39	0.80	0.38

3.4. Performance Outcomes of Our Proposed CNN Model in Classification Tasks

To enhance classification performance, we tested various CNN architectures and ultimately developed the proposed model. Our proposed CNN model has convolutional the 17 layers with 4, 8, and 16 filters, employing Batch Normalization, Leaky ReLU activation and 2x2 Average Pooling throughout. Outputs are flattened and fed into fully connected layers comprising 128 and 64 neurons with ReLU, followed by a sigmoid-activated binary classification layer, as detailed in Figure 7. Training was conducted over 30 epochs using the binary cross-entropy loss function and the Adam optimizer (learning rate: 0.001) to balance stability and convergence, with the dataset split into 80% for training and 20% for testing. Our proposed model was trained using batch sizes of 8, 16, 32, and 64 on promoter sequences converted into images via FCGR with k-mer values of 5 and 6.

Table 5 presents the accuracy (Acc), sensitivity (Sn), specificity (Sp), area under the curve (AUC), and Matthews correlation coefficient (MCC) values for different batch sizes and k-mer values classified by our proposed CNN model.

Table 5. Classification results of our proposed CNN model

k-mer	Batch Size	Acc	Sn	Sp	AUC	MCC
5	64	**0.83**	0.82	0.83	**0.90**	0.65
	32	0.82	0.77	0.87	0.89	0.65
	16	0.80	0.76	0.85	0.89	0.61
	8	0.81	0.74	**0.89**	0.89	0.64
6	64	**0.83**	0.80	0.86	0.88	**0.67**
	32	**0.83**	**0.84**	0.82	0.88	0.66
	16	0.81	0.77	0.84	0.87	0.62
	8	0.82	0.81	0.83	0.88	0.64

The following results were yielded by our image classification using our proposed CNN model: In the k-mer 5 and 6 representation, the highest accuracy (Acc) of 0.83, area under the curve (AUC) of 0.90, and Matthews correlation coefficient (MCC) of 0.67 were achieved with a batch size of 64. In the k-mer 6 representation, the highest sensitivity (Sn) value of 0.84 was observed in the model trained with a batch size of 32, while in the k-mer 5 representation the highest specificity (Sp) value of 0.89 was found in the model trained with a batch size of 8. Figure 13 presents the predicted values of our proposed model displayed on the confusion matrix. With k-mer 5 and a batch size of 64, our proposed CNN model was able to predict 82% of promoters and 83% of non-promoters, achieving an accuracy (Acc) of 0.83 and an area under the curve (AUC) of 0.90.

4. DISCUSSION

In this study, a CNN model was developed by fine-tuning pre-trained ResNet-50, VGG16, and GoogleNet models, as well as the LeNet-5 model, with different hyperparameter values for the prediction of promoter and non-promoter DNA sequences. When classifying the images using the ResNet-50 model with a k-mer size of 5 and a batch size of 8, the ResNet-50 model was able to predict 80% of promoters and 84% of non-promoters, achieving an accuracy (Acc) of 0.82 and an area under the curve (AUC) of 0.89. When classifying with the VGG16 model with a k-mer size of 5 and a batch size of 64, the VGG16 model was able to predict 78% of promoters and 83% of non-promoters, achieving an accuracy (Acc) of 0.80 and an area under the curve (AUC) of 0.88. When classifying with the GoogleNet model with a k-mer size of 6 and a batch size of 16, the GoogleNet model was able to predict 74% of promoters and 76% of non-promoters, achieving an accuracy (Acc) of 0.74 and an area under the curve (AUC) of 0.82. Upon examining the classification performance, it can be seen that the classification accuracy of promoter and non-promoter sequences is relatively low compared to existing literature.

Table 6. Performance comparison of predictors on benchmark dataset

Predictor	Sn	Sp	Acc	AUC	MCC
iPSW(2L)-PseKNC [11]	0.81	0.84	**0.83**	**0.90**	0.66
Le et al. [26]	**0.82**	0.88	0.85	/	0.70
iPSW (PseDNC-DL) [27]	0.83	0.86	0.85	0.92	0.70
BERT-Promoter [13]	0.84	0.86	0.85	**0.90**	/
iProL [15]	0.84	0.86	0.85	0.92	0.71
dPromoter-XGBoost [28]	0.85	0.81	**0.83**	/	**0.67**
iPromoter-CLA [29]	0.86	0.85	0.86	0.92	0.72
Ours	**0.82**	**0.83**	**0.83**	**0.90**	**0.67**

In this study, using the proposed 17-layer CNN model with a k-mer size of 5 and a batch size of 64, the model was able to predict 82% of promoters and 83% of non-promoters,

achieving an accuracy (Acc) of 0.83 and an area under the curve (AUC) of 0.90. When comparing this measure with existing studies in the literature, it can be seen that similar prediction results were obtained using the models proposed in the literature. The performance evaluation results of the studies in the literature are presented in a comparative manner in Table 6.

When examining the performance evaluation metrics presented in Table 6, it can be seen that the proposed model is more successful than the iPSW(2L)-PseKNC [11] method in promoter prediction and achieves the same prediction performance as Le et al. [26]. In non-promoter prediction, the proposed model outperforms the dPromoter-XGBoost [28] method. When comparing accuracy values, the results are the same as those of iPSW(2L)-PseKNC [11] and dPromoter-XGBoost [28]. Regarding the AUC value, the model achieves the same result as iPSW(2L)-PseKNC [11] and BERT-Promoter [13]. In terms of MCC, the proposed model performs better than iPSW(2L)-PseKNC [11] and achieves the same result as dPromoter-XGBoost [28]. It is important to note that in the above prediction models, sequence-to-image conversion and image classification were not employed. In contrast, promoter classification in the literature is often performed using natural language processing methods. This study, however, is the first to convert sequences into images and apply image classification for promoter prediction, marking a methodological innovation. The comparability of the obtained results with the literature indicates that this approach can be used effectively for promoter classification.

5. CONCLUSIONS

In this study, CNN models were utilized to classify color images representing promoter and non-promoter sequences from the E. coli K-12 genome. A total of 6764 images, consisting of 3382 promoter and 3382 non-promoter sequences, were divided into training and testing datasets with an 80:20 split ratio. Next, we utilized three pre-trained models - ResNet-50, VGG16, and GoogleNet - to classify the images. The ResNet-50 model achieved an AUC of 89%, the VGG16 model achieved 88%, and the GoogleNet model achieved 82%. Subsequently, we proposed our custom 17-layer CNN model, which yielded promising results, achieving an AUC of 90%. In a future study, we plan to expand our dataset, assess prediction performance in a different organism, develop and evaluate new models to enhance prediction accuracy, and apply additional pre-trained models for promoter sequence classification.

Author contributions: G.M.Ş; Writing – review & editing, Writing–original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.
Conflict of Interest: No conflict of interest was declared by the authors.
Financial Disclosure: The authors declared that this study has received no financial support.

REFERENCES

[1] F. Xu et al., “dbDEMC 3.0: Functional Exploration of Differentially Expressed miRNAs in Cancers of Human and Model Organisms,” Genomics. Proteomics Bioinformatics, vol. 20, no. 3, pp. 446–454, Jun. 2022, doi: 10.1016/j.gpb.2022.04.006.

[2] D. Castanotto and J. J. Rossi, “The promises and pitfalls of RNA-interference-based therapeutics,” Nature, vol. 457, no. 7228, pp. 426–433, Jan. 2009, doi: 10.1038/nature07758.

[3] A. L. Roy and D. S. Singer, “Core promoters in transcription: old problem, new insights,” Trends Biochem. Sci., vol. 40, no. 3, pp. 165–171, Mar. 2015, doi: 10.1016/j.tibs.2015.01.007.

[4] T. I. Lee and R. A. Young, “Transcriptional Regulation and Its Misregulation in Disease,” Cell, vol. 152, no. 6, pp. 1237–1251, Mar. 2013, doi: 10.1016/j.cell.2013.02.014.

[5] M. De Gobbi et al., “A Regulatory SNP Causes a Human Genetic Disease by Creating a New Transcriptional Promoter,” Science (80-.)., vol. 312, no. 5777, pp. 1215–1217, May 2006, doi: 10.1126/science.1126431.

[6] L. E. Montefiori et al., “A promoter interaction map for cardiovascular disease genetics,” Elife, vol. 7, Jul. 2018, doi: 10.7554/eLife.35788.

[7] R. J. Leeman-Neill et al., “Noncoding mutations cause super-enhancer retargeting resulting in protein synthesis dysregulation during B cell lymphoma progression,” Nat. Genet., vol. 55, no. 12, pp. 2160–2174, Dec. 2023, doi: 10.1038/s41588-023-01561-1.

[8] W. Suza and D. Lee, Genetics, agriculture, and biotechnology. Iowa State University, 2021.

[9] P. Gade and D. V. Kalvakolanu, “Chromatin Immunoprecipitation Assay as a Tool for Analyzing Transcription Factor Activity,” in Transcriptional Regulation: Methods and Protocols, 2012, pp. 85–104.

[10] C. B. Yildiz et al., “EphrinA5 regulates cell motility by modulating Snhg15/DNA triplex-dependent targeting of DNMT1 to the Ncam1 promoter,” Epigenetics Chromatin, vol. 16, no. 1, p. 42, Oct. 2023, doi: 10.1186/s13072-023-00516-4.

[11] X. Xiao, Z.-C. Xu, W.-R. Qiu, P. Wang, H.-T. Ge, and K.-C. Chou, “iPSW(2L)-PseKNC: A two-layer predictor for identifying promoters and their strength by hybrid features via pseudo-K-tuple nucleotide composition,” Genomics, vol. 111, no. 6, pp. 1785–1793, Dec. 2019, doi: 10.1016/j.ygeno.2018.12.001.

[12] M. Oubounyt, Z. Louadi, H. Tayara, and K. T. Chong, “DeePromoter: Robust Promoter Predictor Using Deep Learning,” Front. Genet., vol. 10, Apr. 2019, doi: 10.3389/fgene.2019.00286.

[13] N. Q. K. Le, Q.-T. Ho, V.-N. Nguyen, and J.-S. Chang, "BERT-Promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection," Comput. Biol. Chem., vol. 99, p. 107732, Aug. 2022, doi: 10.1016/j.compbiolchem.2022.107732.

[14] Y. Li et al., "msBERT-Promoter: a multi-scale ensemble predictor based on BERT pre-trained model for the two-stage prediction of DNA promoters and their strengths," BMC Biol., vol. 22, no. 1, p. 126, May 2024, doi: 10.1186/s12915-024-01923-z.

[15] B. Peng, G. Sun, and Y. Fan, "iProL: identifying DNA promoters from sequence information based on Longformer pre-trained model," BMC Bioinformatics, vol. 25, no. 1, p. 224, Jun. 2024, doi: 10.1186/s12859-024-05849-9.

[16] F. Ben Nasr Barber and A. Elloumi Oueslati, "Human exons and introns classification using pre-trained Resnet-50 and GoogleNet models and 13-layers CNN model," J. Genet. Eng. Biotechnol., vol. 22, no. 1, p. 100359, Mar. 2024, doi: 10.1016/j.jgeb.2024.100359.

[17] S. T. Sara, M. M. Hasan, A. Ahmad, and S. Shatabda, "Convolutional neural networks with image representation of amino acid sequences for protein function prediction," Comput. Biol. Chem., vol. 92, p. 107494, Jun. 2021, doi: 10.1016/j.compbiolchem.2021.107494.

[18] J. Shang, C. Peng, X. Tang, and Y. Sun, "PhaVIP: Phage VIrion Protein classification based on chaos game representation and Vision Transformer," Bioinformatics, vol. 39, no. Supplement_1, pp. i30–i39, Jun. 2023, doi: 10.1093/bioinformatics/btad229.

[19] S. Gama-Castro et al., "RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond," Nucleic Acids Res., vol. 44, no. D1, pp. D133–D143, Jan. 2016, doi: 10.1093/nar/gkv1156.

[20] M. F. Barnsley, Fractals Everywhere, 2nd ed. Academic Press, 2014.

[21] H. J. Jeffrey, "Chaos game representation of gene structure," Nucleic Acids Res., vol. 18, no. 8, pp. 2163–2170, 1990, doi: 10.1093/nar/18.8.2163.

[22] A. Halder, Piyush, B. Mathew, and D. Sengupta, "Improved Python Package for DNA Sequence Encoding using Frequency Chaos Game Representation." Apr. 18, 2024, doi: 10.1101/2024.04.14.589394.

[23] A. Shabbir et al., "Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning of ResNet50," Math. Probl. Eng., vol. 2021, pp. 1–18, Jul. 2021, doi: 10.1155/2021/5843816.

[24] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," Int. J. Sci. Res. Publ., vol. 9, no. 10, p. p9420, Oct. 2019, doi: 10.29322/IJSRP.9.10.2019.p9420.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[26] N. Q. K. Le, E. K. Y. Yapp, N. Nagasundaram, and H.-Y. Yeh, "Classifying Promoters by Interpreting the Hidden Information of DNA Sequences via Deep Learning and Combination of Continuous FastText N-Grams," Front. Bioeng. Biotechnol., vol. 7, Nov. 2019, doi: 10.3389/fbioe.2019.00305.

[27] H. Tayara, M. Tahir, and K. T. Chong, "Identification of prokaryotic promoters and their strength by integrating heterogeneous features," Genomics, vol. 112, no. 2, pp. 1396–1403, Mar. 2020, doi: 10.1016/j.ygeno.2019.08.009.

[28] H. Li et al., "dPromoter-XGBoost: Detecting promoters and strength by combining multiple descriptors and feature selection using XGBoost," Methods, vol. 204, pp. 215–222, Aug. 2022, doi: 10.1016/j.ymeth.2022.01.001.

[29] Z. Zhang, J. Zhao, P.-J. Wei, and C.-H. Zheng, "iPromoter-CLA: Identifying promoters and their strength by deep capsule networks with bidirectional long short-term memory," Comput. Methods Programs Biomed., vol. 226, p. 107087, Nov. 2022, doi: 10.1016/j.cmpb.2022.107087.