Int. j. adv. multidisc. res. stud. 2025; 5(4):1159-1162

Received: 16-06-2025
Accepted: 26-07-2025

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Optimization of Deep Learning Convolutional Neural Network for Genomic Sequence Classification

[1] Akram Muhammad Zurgham, [2] Majeed Mehwish

[1, 2] Master's Student, MS Biomedical Engineering, Northeastern University, Shenyang, China

DOI: https://doi.org/10.62225/2583049X.2025.5.4.4740

Corresponding Author: **Akram Muhammad Zurgham**

Abstract

Genomic sequence classification plays a key role in genomics by enabling the categorization of different DNA regions, including promoters, enhancers, coding, and non-coding sequences. The ability to accurately classify these regions is crucial for understanding gene regulation, genome organization, and molecular disease mechanisms. In this study, we optimized a deep learning-based classifier, a convolutional neural network (CNN), improving upon a baseline CNN model from prior research. Using benchmark datasets for genomic sequence classification, our enhanced model demonstrated superior performance in both accuracy and F1-score, validating its effectiveness for high-throughput, sequence-based genomic analysis.

Keywords: Convolutional Neural Network (CNN), Genomic Sequence Classification, Deep Learning, Optimization, Bioinformatics

Introduction

The fast-growing discipline of genomics has completely changed how we perceive biological systems, mechanisms of diseases, and evolutionary processes. The important tasks in genomics are the annotation and classification of DNA sequences, incorporating their functional roles [1, 2]. The four bases found in genomic sequences are adenine (A), cytosine (C), guanine (G), and thymine (T), arranged in specific combinations to achieve a certain biological activity [3, 4]. Amongst these huge expanses of DNA, some parts are promoters, others enhancers, coding or non-coding exons, non-coding introns, or regulatory motifs, which regulate the expression of the gene and its contribution to the life of the cell [5]. Proper identification of these sequences forms the basis of understanding gene regulation, the occurrence of mutations associated with disease, and the interpretation of genome-wide association studies (GWAS) [6].

Conventional methods in the classification of computationally classified genomic sequences depended on alignment-based or heuristic algorithms, which despite being highly applicable in small-scale datasets, were not scalable or malleable to the changing demands of high-throughput modern data via next-generation sequencing (NGS) [7]. Such techniques were not able to identify weak sequence patterns or position-specific interactions, or higher-order interactions, and more complex ways needed to be found [8]. Over the past few years, deep learning (DL) has become a revolutionary tool in computational biology. Particularly, Convolutional Neural Networks (CNNs) have been shown to be effective in large-scale problems involving learning sequential data, like genomics sequences, due to their power to intrinsically learn spatially localized patterns and dependencies.

Gunasekaran *et al*. utilized CNNs and hybrid models for the classification of DNA sequences [9]. In a related contribution, a CNN-based architecture tailored for genome sequence classification, illustrating its adaptability to biomedical applications, including COVID-19-related genomic analysis [10]. Helaly *et al*. conducted a comparative study of CNN-based approaches for biological sequence taxonomic classification [11]. Du *et al*. developed a hybrid DL structure that classified chromosomal DNA. They combine CNNs with another neural network called recurrent neural networks (RNNs), which considers both spatial and sequential characteristics to classify the chromosomal DNA [12]. Soliman *et al*. improved CNN architectural layers that led to accurate and efficient execution of DNA classification tasks [13]. In the meantime, Shujaat *et al*. designed pcPromoter-CNN, a tailored CNN to identify and classify promoter sites in genomic data, and guided its accuracy toward selecting regulatory sequences along with it [14].

In this project, we focused on the task of genomic sequence classification using deep learning-based methods. Building upon

the work of Grešová *et al*. [15], who introduced a collection of benchmark datasets for genomic sequence classification along with a baseline CNN model, while the baseline model provided a foundation, there remained significant potential to improve the predictive accuracy of sequence classification models. Thus, the goal of our study was to provide and evaluate an optimized CNN-based architecture, aimed at outperforming the baseline model in classifying benchmark genomic sequences and performance evaluation using robust classification metrics.

Materials and Methods:

Datasets

In this study publicly, available datasets provided by Grešová *et al*. [15] through the Genomic Benchmark package were utilized. This collection provided a curated assortment of genomic sequence datasets, each containing DNA sequences labeled according to their biological function. Datasets had varied sizes, numbers of sequence classes, and sequence length distributions. Each dataset included thousands of sequences, with class distributions ranging from balanced to moderately imbalanced cases. The sequences themselves were encoded as strings of nucleotide bases. A comprehensive description of all datasets is presented in Table 1. For computational purposes, these character sequences were converted to integer-encoded vectors suitable for deep learning frameworks.

Table 1: Description of datasets in the genomic benchmark package

Datasets	No. of sequences	No. of classes	Class ratio	Median length	Standard deviation
dummy_mouse_enhancers_ensemble	1210	2	1	2381	984.4
demo_coding_vs_intergenomic_seqs	100000	2	1	200	0
demo_human_or_worm	100000	2	1	200	0
drosophila_enhancers_stark	6914	2	1	2142	285.5
human_enhancers_cohn	27791	2	1	500	0
human_enhancers_ensemble	154842	2	1	269	122.6
human_nontata_promoters	36131	2	1.2	251	0

Model architecture and implementation

Optimized CNN architecture specifically tailored for the classification of genomic sequence data (Fig 1). Firstly, an input layer that receives integer-encoded DNA sequences, where each nucleotide base is converted into a numerical index. Then, these indices were then sent to an embedding layer, which is an operation designed to convert a discrete value input to dense, continuous vectors. This will give the model the ability to capture meaningful patterns and contextual relationships in the sequences.

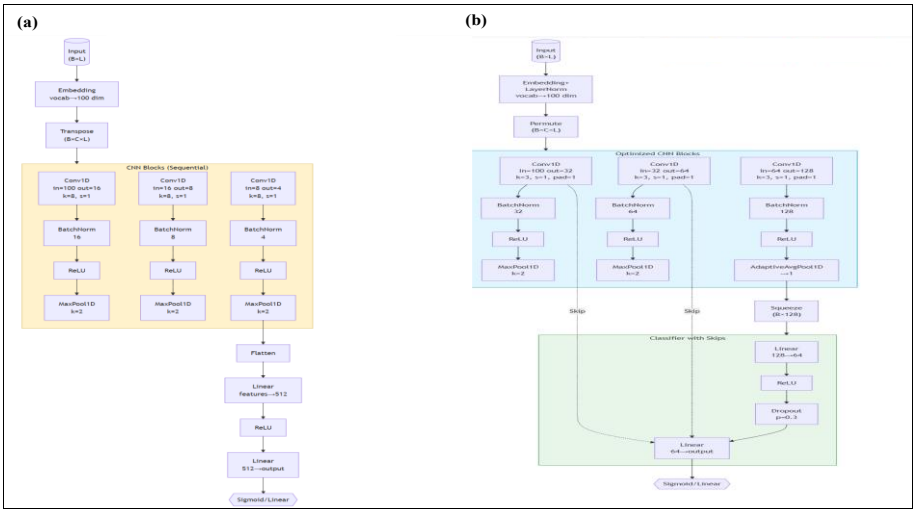

Fig 1: (a) Baseline CNN (b) Optimized CNN: Depthwise separable convolutions, global pooling instead of flattening information, skip connections preserve early features, adaptive pooling maintains key patterns, optimized classifier: bottleneck architecture (128 → 64 → output) & strategic dropout

The embedded representations are then processed through a series of three one-dimensional convolutional blocks, each designed to extract increasingly complex sequence features. The first convolutional block comprises a Conv1D layer that has 32 filters and a kernel size of 3, followed by a batch normalization layer to stabilize activations, a Rectified Linear Unit (ReLU) activation function introducing non-linearity, and a max-pooling layer that aids in decreasing the spatial characteristics of the feature maps.

Table 2: Model architecture refinements: From baseline CNN to optimized version

Feature	Original CNN	FastAccurateCNN	Importance
Convolution Type	Standard 1D Convs	Depthwise Separable Convs	3-4× faster with similar accuracy
Channel Progression	100→16→8→4	100→32→64→128	Increasing channels preserve more features as depth grows
Kernel Size	Large (k=8)	Optimal (k=3)	Better local feature capture
Skip Connections	None	Cross-block feature reuse	Prevents information loss
Pooling	Fixed MaxPooling	AdaptiveAvgPool+ Maxpool	Handles variable-length inputs better
Regularization	Only BatchNorm	LayerNorm + BatchNorm	LayerNorm handles variable-length sequences better & BatchNorm stabilizes training.
Dropout	None	p=0.3 in classifier	Prevents overfitting

The second block had the same structure with an increased number of filters set to 64, while the third block further increases the filter count to 128 and replaces max-pooling with an adaptive average pooling layer. This adaptive pooling operation reduces the feature map to a fixed-size vector, irreventive of sequence length of the input, thus being compatible with the subsequent classifier stage. Following feature extraction, the model includes a fully connected classifier composed of two linear layers. The first linear layer utilizes the ReLU activation function to minimize the dimensionality from 128 to 64, maintaining the non-linearity and dropout layer with a dropout probability of 0.3 to reduce overfitting. The final linear layer maps the 64-dimensional vector to the number of output classes, using a sigmoid activation function for binary classification tasks or leaving the raw logits for multi-class problems. Table 2 shows the model overall architecture refinements.

Table 3: Key training modifications from baseline to optimized model

Change	Basline	Optimized	Benefit
Optimizer	Adam	AdamW	Better Weight decay handling
LR Scheduler	None	OneCycleR	Faster convergence
Loss Scaling	Basic	Gradient Clipping	More stable training

The training procedure involved dividing 80% of each dataset for training and 20% for testing, ensuring reproducibility through a random split using the train_test_split() function. The model was trained in mini-batches, performing forward passes to generate predictions, computing the loss using either binary cross-entropy with logits loss or cross-entropy loss depending on the task, and backpropagating gradients to update the model parameters. The key training modifications are shown in Table 3.

Furthermore, the AdamW optimizer was applied for its robust convergence properties, complemented by a OneCycleLR learning rate scheduler that dynamically adjusted learning rates within each epoch for improved training stability and speed. Training performance was tracked in terms of cumulative loss and accuracy, with metrics recorded at the end of each epoch. After training completion, the model's generalization performance was assessed on the test set, and final accuracy and F1-score metrics were computed, with the latter providing an especially important measure in cases of class imbalance.

Results

Our results showed significant improvements in classification performance compared to the baseline CNN. In Fig 2a, the accuracy scores of both models across multiple genomic datasets were demonstrated. Optimized CNN consistently outperformed the baseline CNN in every dataset evaluated. The enhancement in accuracy was between small increases in balanced sets of data and significant steps in sets of data with increased sequence diversity and complexity. This indicated that this optimization is better at generalising sequence patterns, even in challenging classification tasks.

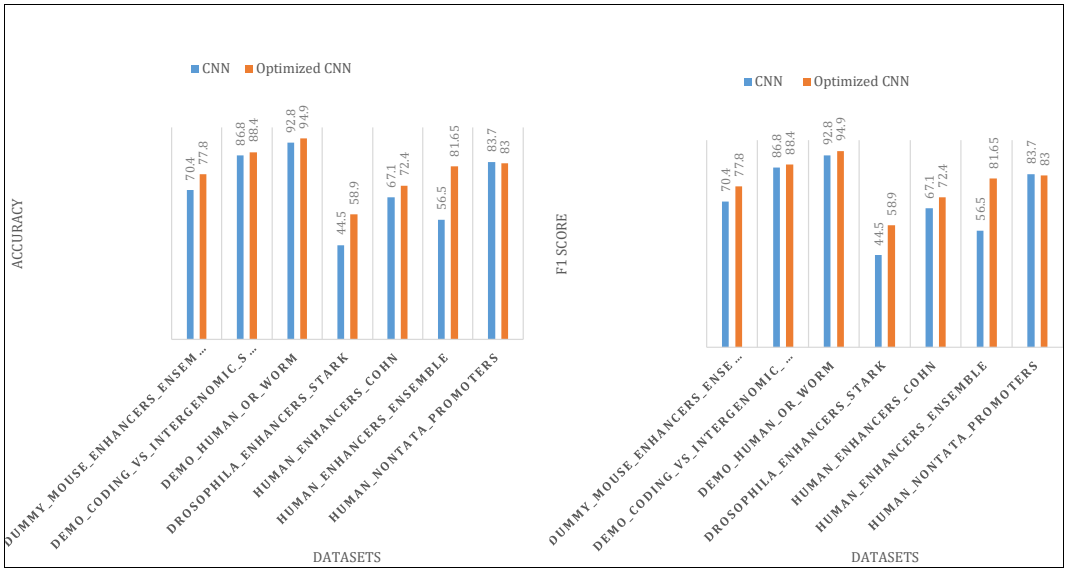

Fig 2: (a) Shows model accuracy across datasets, whereas (b) shows model F1 scores across datasets

F1-score metrics evaluation is presented in Fig 2b; datasets exhibited moderate class imbalance; the F1 score provided a more sensitive measure of model effectiveness by taking into consideration both false positives and false negatives. Our model achieved demonstrates higher F1-scores across all datasets, which shows its superior ability to accurately predict minority classes while maintaining high precision and recall.

Table 4 shows performance scores for both models across several genomic sequence datasets. Datasets with more complex sequence patterns or imbalanced class distributions showed more improvement, where the enhanced architecture's capacity to extract richer sequence features and its regularization techniques (like dropout and adaptive pooling) resulted in better generalization on unseen test data. This improvement is credited to architectural advancement which includes the use of depthwise separable convolutions, the use of batch normalization to guarantee stability of activation, adaptive average pooling to handle the various length of sequences, and dropout regularization to overcome overfitting.

Table 4: Comparison of classification performance between baseline CNN and optimized CNN across datasets

Datasets	Baseline CNN		Optimized CNN	
	Accuracy	F1 score	Accuracy	F1 score
dummy_mouse_enhancers_ensemble	69	70.4	79.3	77.8
demo_coding_vs_intergenomic_seqs	87.6	86.8	88.8	88.4
demo_human_or_worm	93	92.8	95	94.9
drosophila_enhancers_stark	58.6	44.5	68.6	58.9
human_enhancers_cohn	69.5	67.1	71.8	72.4
human_enhancers_ensemble	68.9	56.5	82.1	81.65
human_nontata_promoters	84.6	83.7	84.7	83

These results demonstrate that careful refinements in architecture and training procedure protocols can lead to deep learning models with significantly better predictive ability in the field of genomic sequence classification.

Conclusion

This study has shown that deep learning-driven genomic sequence classification can be enhanced considerably by the optimization of CNN architectures and training procedures. The given proposed optimized model was superior to a baseline CNN in its accuracy and F1-scores among several genome benchmark datasets. The improvements were mainly due to key additions such as multiple convolutional layers, batch normalizations, adaptive pooling, and AdamW optimizer with OneCycleLR scheduler. For future work, extending the model to multi-label sequence classification and integrating attention mechanisms could enhance its capacity to capture long-range dependencies in DNA sequences. Additionally, to extend the applicability of the model by testing it on real-world genomic data, and even with epigenetic annotations. This paper reports on the potential of optimization in DL in the field of computational genomics, as well as a practical and scalable framework that can be applied in the future to genome annotation and precision medicine.

Acknowledgement:
Not applicable.

References

1. Mock F, Kretschmer F, Kriese A, Böcker S, Marz M. Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. Proc Natl Acad Sci. 2022; 119(35):e2122636119.
2. Maire CL, Fuh MM, Kaulich K, Fita KD, Stevic I, Heiland DH, *et al*. Genome-wide methylation profiling of glioblastoma cell-derived extracellular vesicle DNA allows tumor classification. Neuro-Oncology. 2021; 23(7):1087-1099.
3. Bailey J. Nucleosides, nucleotides, polynucleotides (RNA and DNA) and the genetic code. In: Inventive Geniuses Who Changed the World: Fifty-Three Great British Scientists and Engineers and Five Centuries of Innovation. Springer, 2021, 313-340.
4. Tong H, Wang H, Wang X, Liu N, Li G, Wu D, *et al*. Development of deaminase-free T-to-S base editor and C-to-G base editor by engineered human uracil DNA glycosylase. Nat Commun. 2024; 15(1):4897.
5. Ntini E, Marsico A. Functional impacts of non-coding RNA processing on enhancer activity and target gene expression. J Mol Cell Biol. 2019; 11(10):868-879.
6. Duncan LE, Ostacher M, Ballon J. How genome-wide association studies (GWAS) made traditional candidate gene studies obsolete. Neuropsychopharmacology. 2019; 44(9):1518-1523.
7. Kucherov G. Evolution of biosequence search algorithms: A brief survey. Bioinformatics. 2019; 35(19):3547-3552.
8. Ao C, Jiao S, Wang Y, Yu L, Zou Q. Biological sequence classification: A review on data and general methods. Res, 2022, 0011.
9. Gunasekaran H, Ramalakshmi K, Rex Macedo Arokiaraj A, Deepa Kanmani S, Venkatesan C, Suresh Gnana Dhas C. Analysis of DNA sequence classification using CNN and hybrid models. Comput Math Methods Med. 2021; 1:1835056.
10. Gunasekaran H, Ramalakshmi K, Ramanathan S, Venkatesan R. A deep learning CNN model for genome sequence classification. In: Intelligent Computing Applications for COVID-19. CRC Press, 2021, 169-185.
11. Helaly MA, Rady S, Aref MM. Convolutional neural networks for biological sequence taxonomic classification: A comparative study. In: Int Conf Adv Intell Syst Informatics. Springer, 2019, 523-533.
12. Du Z, Xiao X, Uversky VN. Classification of chromosomal DNA sequences using hybrid deep learning architectures. Curr Bioinform. 2020; 15(10):1130-1136.
13. Soliman N, Abd-Alhalem S, El-Shafai W, Abdulrahman S, Abd El-Samie F. An improved convolutional neural network model for DNA classification. Comput Mater Continua. 2022; 70(3):5907-5927.
14. Shujaat M, Wahab A, Tayara H, Chong KT. pcPromoter-CNN: A CNN-based prediction and classification of promoters. Genes. 2020; 11(12):1529.
15. Grešová K, Martinek V, Čechák D, Šimeček P, Alexiou P. Genomic benchmarks: A collection of datasets for genomic sequence classification. BMC Genomic Data. 2023; 24(1):25.