Horizontal Sequence Pooling Technique in Convolutional Neural Networks to Optimize Feature Extraction for DNA Sequence Classification

Lilibeth Coronel
Graduate Programs, Technological Institute of the Philippines, Mindanao State University at Naawan, Quezon City, Philippines
lilibeth.coronel@msunaawan.edu.ph

Arnel Fajardo
College of Computing Science, Information and Communication Technology, Isabela State University, Cauayan City, Philippines
acfajardo2011@gmail.com

Ruji Medina
Graduate Programs, Technological Institute of the Philippines, Quezon City, Philippines
ruji.medina@tip.edu.ph

Abstract: *The exact positioning of features within the sequence is important in Deoxyribonucleic Acid (DNA) sequence classification, as it encodes the unique genetic information of each organism. In Convolutional Neural Networks (CNNs), pooling techniques are vital for efficient feature extraction. However, traditional pooling techniques demonstrated some limitations in domain-specific pooling for sequence-based data analysis, specifically, lack of positional sensitivity, thereby, encountering information loss. To address these constraints, this study introduces Horizontal Sequence Pooling (HSP), a novel pooling technique that enhances feature extraction by applying positional pooling of sequences across the horizontal axis of the feature maps. The CNN model framework was optimized through data preprocessing and hyper-parameter tuning. The results validate that HSP significantly outperforms traditional pooling techniques across multiple metrics. It achieved a reduction in feature parameters by as high as 96% and validation loss by 19%. Furthermore, HSP attained the highest accuracy of 96%, a Matthews Correlation Coefficient (MCC) of 96%, and an Area-Under-the-Curve Precision and Recall (AUC-PR) score of 99%, indicating its superior ability to balance precision and recall. These results underscore HSP's efficiency in feature extraction and its capability to handle complex, imbalanced datasets, making it a highly effective method for DNA sequence classification in CNN architectures.*

Keywords: *Pooling technique, classification, convolutional neural network, feature extraction, genomic sequence.*

Received April 18, 2024; accepted August 26, 2024
https://doi.org/10.34028/iajit/21/5/6

1. Introduction

Deoxyribonucleic Acid (DNA) sequence classification is crucial in computational biology [10]. It consists of two strands, each comprising numerous nucleotides, and is structured into a characteristic double helix formation, represented by four bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) [20]. The positioning of nucleotides is vital for gene expression and most DNA-related processes, as it determines the unique genetic information [26]. Extracting this important genetic information remains challenging. Feature extraction identifies and extracts the most dominant features from the input data for the specific task, resulting in positional or rotational features of that position [12].

Convolutional Neural Networks (CNNs) have gained popularity for genomic sequence analysis, establishing state-of-the-art accuracy in various biological data analyses. It consists of nodes that identify local features from the input vector, minimize parameters through the pooling layer, and subsequent layers merge these features into a fully connected layer [24]. However, when a CNN uses a large max pooling window size in its first layer, it disrupts the spatial arrangement of segmented features, preventing subsequent layers from hierarchically combining these into complete feature representations [14].

Pooling is one of the important components of CNNs. The most common pooling methods are max pooling and average pooling. Despite their success, these pooling methods have significant limitations when applied to DNA sequence classification. On the other hand, pooling operations encounter a loss of information while extracting features which affects classification accuracy [1]. Max pooling can lead to a loss of critical spatial information, which is particularly problematic in genomic data where the position of nucleotides is essential for accurate classification. The approach may also amplify noise by selecting the maximum value within a pooling window, which might not represent the true underlying pattern [27, 33]. While average pooling [17] involves taking the average of each feature map and retaining distinct features, it can potentially decrease prediction accuracy [30] and tends to blur important features [2], reducing the sensitivity of the model to variations in the DNA sequence.

Other state-of-the-art pooling methods like mixed pooling and Horizontal max (Hmax) pooling are recognized in the field of deep learning. Mixed pooling [31] combines max and average pooling, either by randomly selecting them for each patch or by mixing their outputs. However, it introduced variability in how features are represented across different patches, which can make it harder for the model to learn robust representations. While Hmax pooling [21] reduces noise and detects features by acquiring the maximum value from cumulative horizontal pixels, the method is specifically applied to image datasets. Furthermore, existing pooling techniques, do not adequately address unique characteristics of sequence-based data, leading to suboptimal performance in genomic analysis.

Thus, this study proposed a novel technique called Horizontal Sequence Pooling (HSP), which modifies the Hmax pooling to be specifically tailored for DNA sequence data. While Hmax pooling applied max pooling to pixel values of the images horizontally, our proposed method utilized max pooling and average pooling to balance and optimize feature extraction of sequence-based data, leading to better performance in classification tasks. The innovation here is the method applies positional pooling across the horizontal axis of the feature maps, where nucleotides are paired. This approach uniquely preserves the feature position of nucleotides, minimizes the loss of information by positional max pooling, and reduces noise by averaging the max-pooled features, thereby addressing the critical gap in existing methods. Data preprocessing and hyper-parameter tuning are carried out in the CNN model architecture to fit DNA sequence classification. The study evaluates the effectiveness of the proposed pooling technique compared to traditional and state-of-the-art pooling techniques, emphasizing its capability to reduce feature parameters, reduce loss, and enhance classification accuracy. Additionally, a comparative analysis is performed to identify the most suited CNN model for imbalanced data, using metrics such as precision, recall (sensitivity), specificity, F1-score, Area-Under-the-Curve Precision and Recall (AUC-PR), and Matthews Correlation Coefficient (MCC).

2. Related Literature and Works

2.1. Pooling Techniques

In convolutional-based systems, pooling plays a crucial role in decreasing the dimensions of extracted features. These features are subsampled to produce multiple feature maps with reduced resolutions [32].

Max pooling [4] is the most widely used pooling approach in CNNs. This method selects the maximum value within a pooling window and has been widely adopted due to its simplicity and effectiveness in reducing dimensionality while retaining the most prominent features. The equation is given in (1), where *Rmax* is a pooling region and $\{\alpha_1,\ldots,\alpha_{|\mathrm{Rmax}|}\}$ is a set of activations [23].

$$f_{max} = \max_{i \in R_{max}} a_i \tag{1}$$

Average pooling [15] computes the statistical mean of a neuron cluster in the feature map. In the average pooling approach, the input image is segmented into several distinct rectangular regions. The mean of the pixel values within each of these rectangles is computed, and this average is used to form the output. Mathematically, average pooling is given in Equation (2), where vector *x* represents activations from a set of *N* permutations in a rectangular area of an image or channel.

$$f_{ave}(X) = \frac{1}{N}\sum_{i}^{N} = 1x_i \tag{2}$$

Mixed pooling [31] randomly selects max pooling and average pooling operations. The technique alters the pooling pattern in the stochastic method to mitigate the issues typically associated with max pooling and average pooling. The equation is given in (3), where λ is a random value either 0 or 1, indicating the selection of using the max pooling or average pooling.

$$y_{kij} = \lambda \cdot \max_{(p,q)\epsilon R_{ij}} + (1-\lambda) \cdot \frac{1}{|R_{ij}|} \sum_{(p,q)\epsilon R_{ij}} x_{kpq} \tag{3}$$

Recently, a new pooling technique called Horizontal Max pooling [21] has been proposed. It is a noise reduction and feature detection technique for images that focuses on obtaining the maximum value from the sum of horizontal pixels. This method enhances edge prominence more effectively than average pooling and reduces noise better than max pooling. The algorithm iterates through the pool array by adding the first and third pixel values of the image, dividing it by 2, and stored into variable *A* and the same with the second and fourth into variable *B*. It then compares *A* and *B*, appending the maximum to a new output array.

In summary, all the pooling functions are fundamentally rooted in max pooling and/or average pooling, with their variations depending on the specific operations performed. The limitation of traditional pooling techniques and the improvements in state-of-the-art pooling techniques directly motivate the development of the proposed HSP technique. While max pooling excels in feature prominence and average pooling in spatial retention, neither adequately addresses the unique requirements of DNA sequence classification. The critical need to preserve positional information in nucleotide sequences necessitates a novel approach that can integrate the strengths of both techniques while mitigating their weaknesses. The proposed technique not only preserves the hierarchical structure of DNA sequences but also enhances classification accuracy by minimizing loss during the pooling process.

2.2. Convolutional Neural Network

CNNs established a foundation in the domain of deep learning [16]. CNNs are adept at automatically learning features at various levels of abstraction from raw input data, which has transformed the field of computer vision and beyond [9]. CNNs are mainly effective in applications such as image classification with high accuracy and have become a powerful tool in bioinformatics, particularly for DNA sequence classification tasks. Pooling layers, which reduce the spatial dimensions of feature maps, are critical components in CNNs, helping to manage computational complexity while retaining important features. Recent studies have explored various pooling strategies to enhance the performance of CNNs in DNA sequence analysis, highlighting the evolving nature of this field.

Max pooling is a widely used technique in CNNs, where the maximum value from a pooling window is selected to downsample the feature map. This method has proven effective in several DNA sequence classification tasks. For instance, [25] employed a modified CNN model for DNA sequence classification, integrating max pooling layers and a novel downsampling method to enhance classification accuracy. This approach demonstrated a significant improvement in processing time and accuracy, particularly in the classification of 16S rRNA bacterial sequences. Similarly, [22] utilized a global-max-pooling layer in their study on cancer subtyping using single point mutations, The layer was crucial in addressing the challenges of processing genetic mutation data, resulting in improved performance in cancer subtyping tasks. The efficacy of max pooling in these contexts underscores its utility in reducing feature map dimensionality while preserving critical information necessary for accurate classification.

Recent studies have explored hybrid pooling strategies, combining max pooling with other pooling methods to capture both local and global features more effectively. A hybrid CNN model was developed by [6] that integrates both max and average pooling. This approach allowed the model to capture small-and-large-scale local features, enhancing the prediction of protein-protein interactions. The combination of pooling methods facilitated a more comprehensive feature extraction, leading to superior performance in classification tasks. A hybrid pooling layer design called AVG-MAX VPB was also proposed by [19], which combines average and max pooling to improve the accuracy of breast cancer classification from thermograms. By leveraging the strengths of both pooling methods, the AVG-MAX VPB design enabled the model to collect informative features more effectively, resulting in higher classification accuracy compared to traditional pooling methods.

Several studies have introduced novel pooling techniques tailored to specific genomic applications. A deep learning framework called PhosVarDeep was developed by [18] that employed a CNN architecture with pooling layers to predict phosphor-variants. This model significantly outperforms traditional machine learning methods, highlighting the importance of pooling layers in refining feature extraction and improving prediction accuracy in protein sequence analysis. While [7] proposed a novel CNN-based approach using the Hamming distance-based pooling technique to improve the classification of DNA sequences. This method enhanced the discriminative power of the model, enabling it to outperform existing state-of-the-art techniques.

Hybrid models that combine CNNs with other neural network architectures, such as Long Short-Term Memory (LSTM) and GRU, have also benefited from the integration of pooling layers. A study [29] used a combination of CNN, BiLSTM, and Bi-GRU with pooling layers to detect mutations in lung cancer DNA sequences. The inclusion of pooling layers contributed to the high accuracy achieved in mutation classification, demonstrating the effectiveness of this approach in handling sequential data. Likewise, [8] also integrated pooling layers in a hybrid deep learning model that combines CNNs with LSTM and bidirectional architectures for viral DNA sequence classification. The pooling layers played a vital role in achieving high classification accuracy, particularly in addressing the challenges of imbalanced datasets.

Pooling layers have also been applied in novel ways to address specific challenges in genomic studies. The use of CNNs alongside Graph Neural Networks (GNNs) for predicting Z-DNA regions was explored by [28]. Pooling layers were instrumental in refining feature extraction, enabling the model to achieve better accuracy in genomic functional element prediction.

The studies reviewed demonstrate the diverse and critical roles that pooling layers play in enhancing the performance of CNNs for DNA sequence classification tasks. From traditional pooling to advanced hybrid and novel pooling techniques, these methods have significantly contributed to the accuracy and efficiency of CNN models in various genomic applications.

3. Proposed Method

3.1. Datasets

Our dataset was curated using fish DNA/Genomic sequences from the public nucleotide sequence database: “Barcode Of Life Data (BOLD) system” (https://www.boldsystems.org/). The DNA sequence data is formatted as a Comma Separated Value (CSV) file. The original dataset is a collection of 53,503 DNA samples, linking scientific names to nucleotide sequences that are 648~655 bases in length. It includes 1,235 species names sourced from 163 countries.

3.2. Data Preprocessing and Augmentation

Preprocessing is a vital phase in many machine learning and deep learning algorithms, especially when dealing with numerical data rather than non-numeric types. In the context of the DNA dataset, the genomic sequences are considered non-numeric data. In this study, one-hot encoding is applied to convert sequences into numerical format, which retains the positional information of each nucleotide within the sequences, and padding is utilized to handle sequences of varying lengths ensuring consistent input size for the model. For any characters in sequence that do not match the known nucleotides, a '-' is used for padding and represented as [0, 0, 0, 0]. The species class ID and nucleotide sequences are transformed into a one-hot encoded format using vector representation. Each nucleotide (A, C, G, T) is represented as a distinct 4-dimensional vector as shown in Figure 1.

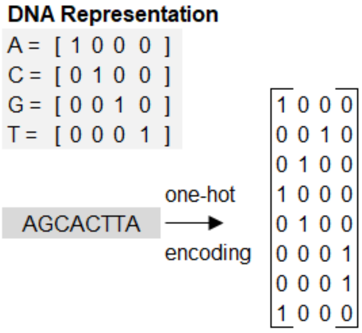
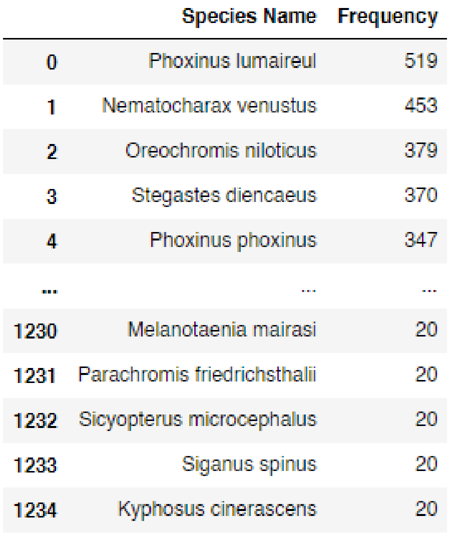


Figure 1. One-hot encoding of nucleotide DNA sequence.

The class frequency distribution of each species is shown in Figure 2. It is observed that there is an imbalanced dataset, thus a data augmentation technique called Synthetic Minority Oversampling Technique (SMOTE) [3, 5], is applied to handle this problem. Synthetic samples are created using the SMOTE algorithm to closely align the minority class with the majority class. The algorithm identifies minority classes based on a specified threshold which is set to 47 such that classes with occurrences less than the threshold are considered minority classes. SMOTE generates synthetic samples to balance the minority class with the majority class, thereby enhancing the model's ability to learn from underrepresented data points.

In the SMOTE process, first, one-hot encoded labels are converted into single labels to facilitate the identification of minority classes. Second, a threshold is set to determine the minority classes. Classes with occurrences below the threshold are considered minority classes. Third, the samples belonging to the minority classes are extracted for the SMOTE process. The feature data is flattened to fit the SMOTE requirements. The algorithm is applied to the minority data to generate synthetic samples. This helps in balancing the class distribution. Fourth, the synthetic samples generated from the minority classes are combined with the original majority class samples to form a balanced dataset. Finally, the resampled feature data is reshaped back to its original format, and the labels are one-hot encoded again to be used as input for the CNN model. By applying SMOTE, we ensure that the dataset is balanced, which improves the model's ability to learn and classify fish species across all classes, including those that were initially underrepresented.



	Species Name	Frequency
0	Phoxinus lumaireul	519
1	Nematocharax venustus	453
2	Oreochromis niloticus	379
3	Stegastes diencaeus	370
4	Phoxinus phoxinus	347
...	...	...
1230	Melanotaenia mairasi	20
1231	Parachromis friedrichsthalii	20
1232	Sicyopterus microcephalus	20
1233	Siganus spinus	20
1234	Kyphosus cinerascens	20

Figure 2. Frequency by species class.

3.3. Horizontal Sequence Pooling Technique

The HSP provides an advanced representation that captures the maximum presence of nucleotides in pairs and retains the most significant features within the window. This is achieved by applying the average operation; ensuring important features are not lost while reducing noise.

- Representation of the Sequence: each sequence is represented as a matrix *W* with dimensions 4 x *m*, where 4 corresponds to the one-hot encoding dimensions for nucleotides A, C, G, and T, and *m* is the window or column size. *R* represents the set of real numbers.

$$W \in R^{4\times m} \tag{4}$$

- Sliding window approach: for each *W*, slides over the sequence with a predefined pool size and stride, capturing local information at each step. Here *X* is the input tensor, *p* is the pool size, and *i* is the starting index of the window. The process ensures that the pooling operation is applied systematically across the entire sequence.

$$W_i = X[:, i: i + p] \tag{5}$$

- Feature extraction within the window: for each window, W_i, two features, $feature_a$, and $feature_b$, are extracted by calculating the maximum values at specific positions within the window. Here, $feature_a$ extracts maximum values from indices 0 and 2, while $feature_b$ extracts maximum values from indices 1 and 3. This operation ensures that important features are not lost.

$$feature_a = max(W_i[:,:,[0,2]], axis = 1) \tag{6}$$

$$feature_b = \max(W_i[:,:,[1,3]], axis = 1 \tag{7}$$

- Pooling operation: the pooling operation is computed by averaging the features *feature$_a$* and *feature$_b$*. This average operation ensures that noise is reduced and that the most significant features within the window are retained without losing important features.

$$HSP = ave(feature_a + feature_b) \tag{8}$$

The pooled features from all windows are stacked along the temporal dimension to form the final output tensor. This process ensures that the entire sequence is covered and relevant features are aggregated effectively. Figure 3 shows an example of how pooling operation works.

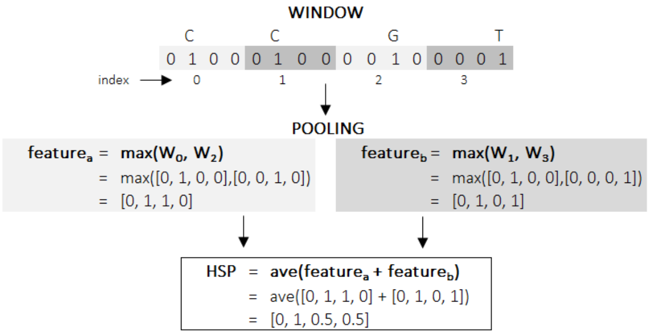

WINDOW
C C G T
0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1
index
0 1 2 3
POOLING
feature$_a$ = max(W$_0$, W$_2$)
= max([0, 1, 0, 0],[0, 0, 1, 0])
= [0, 1, 1, 0]
feature$_b$ = max(W$_1$, W$_3$)
= max([0, 1, 0, 0],[0, 0, 0, 1])
= [0, 1, 0, 1]
HSP = ave(feature$_a$ + feature$_b$)
= ave([0, 1, 1, 0] + [0, 1, 0, 1])
= [0, 1, 0.5, 0.5]

Figure 3. Example of HSP operation.

3.4. The Model

The proposed model design as presented in Figure 4, details the implementation of an enhanced CNN pooling layer using the HSP technique to optimize feature extraction for classification tasks. The choice of hyper-parameters and architecture components was driven by the need to effectively capture and process the intricate sequential nature of the input data.

The architecture of the proposed model starts with a one-dimensional convolutional layer designed to capture a diverse set of features from the input sequence data. This layer utilized 64 filters, which were chosen to effectively identify various patterns within the data. The kernel size was set to 3, striking a balance between capturing local dependencies and maintaining computational efficiency. The Rectified Linear Unit (ReLU) activation function was employed to mitigate the vanishing gradient problem and enhance convergence speed. The input shape for this layer was (655, 4), which corresponds to the sequence length and the one-hot encoded representation of the nucleotides (A, C, G, T).

Following the convolutional layer, a HSP was incorporated. Both the pool and stride size for this layer were set to 4. This configuration helps in reducing the dimensionality of the feature maps while retaining the most significant information within each window. By ensuring that key features are not lost, the HSP layer enhances the model's generalization capabilities.

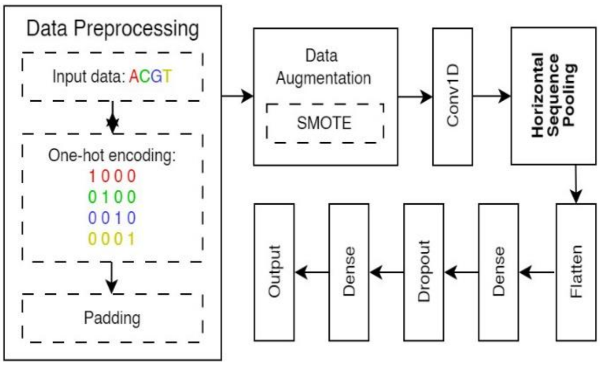


Figure 4. The overall model with HSP.

Next, the model includes a flatten layer, which converts the pooled feature map into a one-dimensional vector. This transformation is crucial for preparing the data for the subsequent fully connected layers. The first fully connected dense layer contains 256 neurons, providing a high capacity for learning complex patterns within the data. ReLU activation function is used again in this layer to ensure non-linearity and efficient training capabilities.

To prevent overfitting during the training process, a Dropout [11] layer was added with a dropout rate of 0.5. This technique randomly sets half of the input units to zero during training, encouraging the model to learn more robust features. Finally, the model concludes with an output dense layer that uses softmax activation. This layer was designed for the classification task, ensuring that the output probabilities across the classes sum to one, making it suitable for multi-class classification tasks.

3.5. Evaluation Metric

To assess the effectiveness of the proposed method for DNA sequence classification, five key performance metrics are employed: Accuracy, Precision, Recall (Sensitivity), Specificity, and F1-score. The AUC-PR and MCC functions of the TensorFlow are also utilized which summarizes the trade-off between precision and recall for different probability thresholds of the imbalanced data. The metrics provide a comprehensive evaluation of the method's accuracy and its ability to distinguish between classes accurately.

$$Accuracy = (TP + TN)/(Total\ no.of\ samples) \tag{9}$$

$$Precision = TP/(TP + FP) \tag{10}$$

$$Recall\ (Sensitivity) = TP/(TP + FN) \tag{11}$$

$$Specificity = TN/(TN + FP) \tag{12}$$

$$F1 - score = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{13}$$

4. Results and Discussion

The proposed model is experimented with using Python 3.9.12 on an Intel(R) Core i7 10870H CPU @ 2.20GHz with 32GB of random-access memory and 6GB NVIDIA GeForce RTX 3060 of graphics processing units. Following the application of data augmentation, the dataset expanded to 69,965, which are split into training, validation, and testing sets with respective proportions of 70% (48,975), 20% (13,993), and 10% (6,997). In the training phase, the categorical cross-entropy function of the TensorFlow is used as the loss function, the Adam optimizer [13] is set at a learning rate of 0.001, and the CNN model is trained to measure accuracy, loss, and AUC-PR which helps in evaluating the models' prediction correctness and quality. Table 1 shows the summary of the CNN model architecture with different pooling techniques. The total number of parameters of the model with HSP is substantially lower (13.45%) compared to Hmax pooling, max pooling, and average pooling, and (3.65%) compared to the mixed pooling technique. These differences highlight the efficiency of the HSP technique in terms of parameter utilization and overall model complexity.

Table 1. CNN model architecture with different pooling techniques.

Model	Total trainable parameters
CNN-max pooling	2,989,075
CNN-average pooling	2,989,075
CNN-mixed pooling	11,000,851
CNN-horizontal max pooling	2,989,075
CNN-HSP	401,939

Five different models are trained, each using a distinct pooling technique: HSP, Hmax pooling, mixed pooling, average pooling, and max pooling in 15 epochs with a 128-batch size of nucleotide sequences which provides a stable update during the training process. Figures 5 to 10 present comparative convergence plots depicting the training and validation loss, accuracy, and AUC-PR across different pooling techniques.

Figures 5 and 6 illustrate the decreasing trend in training and validation loss across the five pooling techniques, revealing significant differences in their performance. Max pooling exhibits the highest validation loss of 4.8726 and shows minimal improvement, reflecting its inefficiency in capturing the nuanced genetic features necessary for distinguishing between closely related species. This poor performance highlights its limitations in generalizing to unseen data. Average pooling shows a much better performance with a significantly lower validation loss of 0.2653. This indicates its moderate ability to extract and generalize key features from the DNA sequences. Mixed pooling and Hmax pooling show intermediate performance, with validation losses of 1.5035 and 1.981, respectively, suggesting they can capture more relevant features than max pooling but still do not fully exploit the complex patterns within the dataset.

HSP on the other hand, stands out by achieving the lowest validation loss of 0.1929 closely aligned with its training loss. This alignment indicates that HSP effectively captures the most informative features within fish DNA sequences, leading to a superior generalization of new data. The strong convergence and low validation loss suggest that HSP minimizes overfitting, making it the most efficient pooling technique among those tested.

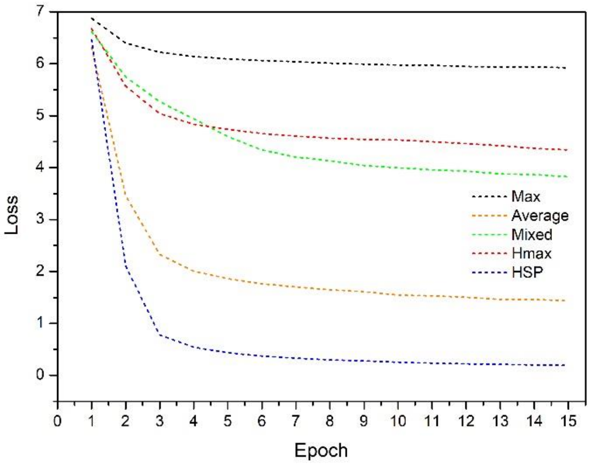


Figure 5. Convergence plot of training loss.

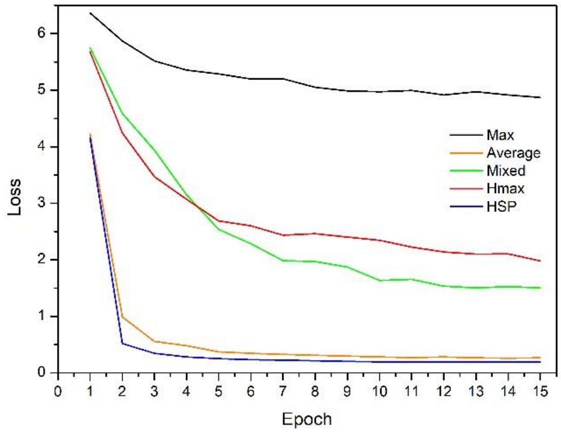


Figure 6. Convergence plot of validation loss.

As shown in Figures 7 and 8, while all five techniques demonstrate improvements in training and validation accuracy over time, there are clear disparities in their effectiveness, reflecting their ability to handle the intricacies of this diverse dataset. HSP emerges as the most effective technique, achieving the highest training accuracy of 0.9439 and validation accuracy of 0.9588. This superior performance indicates HSP's significant capability to capture complex patterns within DNA sequences and deliver robust classification performance. On the other hand, max pooling performs the worst, with 0.0258 and 0.1041 training and validation accuracies, underscoring its limitations in effectively extracting relevant features from the DNA sequences, leading to poor classification performance. Average pooling performs moderately, with 0.5996 training and 0.9398 validation. While average pooling can smooth out data and reduce noise, it may also miss some critical genetic variations essential for accurate species classification. Mixed pooling and Hmax pooling, with 0.1851 and 0.1195 training and validation accuracies of 0.7790 and 0.6748, respectively, perform better than max pooling. These results highlight the clear advantage of using HSP for tasks that require high accuracy in feature extraction and classification.

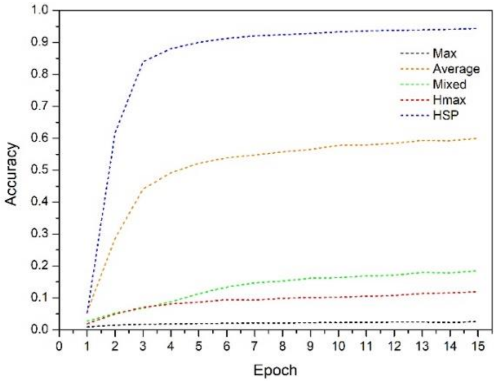


Figure 7. Convergence plot of training accuracy.

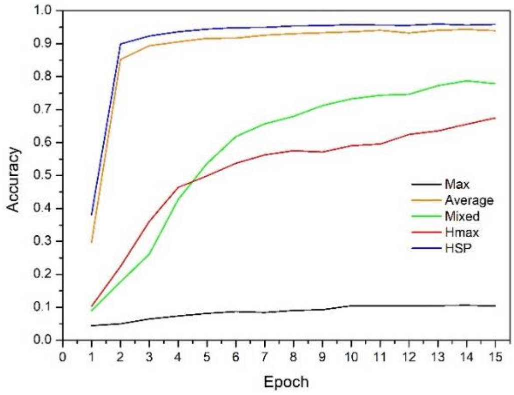


Figure 8. Convergence plot of validation accuracy.

Figures 9 and 10 illustrate the AUC-PR learning curves for training and validation across the five pooling techniques, revealing clear differences in their ability to balance positive predictions and capture all positive instances. HSP demonstrates exceptional performance, achieving the highest AUC-PR scores of 0.9822 in training and 0.9848 in validation. This superior performance indicates HSP's exceptional ability to accurately capture the essential features in the DNA sequences that distinguish one species from another while minimizing false positives. In contrast, max pooling shows the poorest performance, with training and validation AUC-PR scores of 0.0287 and 0.0936, respectively, indicating its inefficiency in managing imbalanced data and accurately capturing positive instances. Average pooling performs reasonably well, with AUC-PR scores of 0.6954 in training and 0.9782 in validation, but still falls short of HSP, suggesting that while it can maintain a good balance and capture intricate patterns, it does not match HSP precision. Mixed pooling and Hmax pooling, with training AUC-PR scores of 0.2154 and 0.1372, and validation scores of 0.7731 and 0.6293, respectively, underperform compared to HSP. These results emphasize the effectiveness of HSP in handling complex, imbalanced data, requiring high precision in identifying position instances.

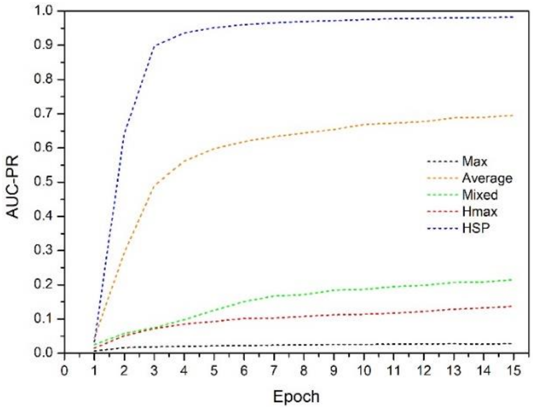


Figure 9. Convergence plot of training AUC-PR.

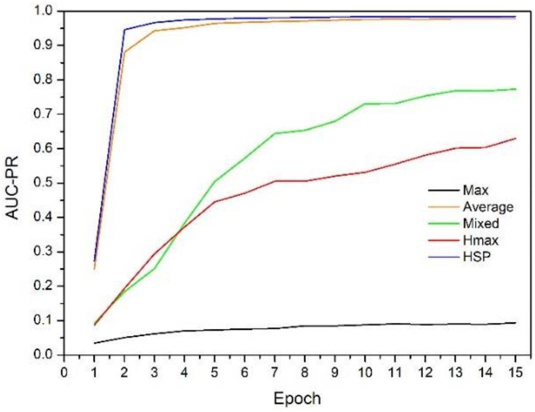


Figure 10. Convergence plot of validation AUC-PR.

Moreover, Table 2 provides a comprehensive overview of the performance of five different pooling techniques on the test dataset, evaluated across several key metrics. The results demonstrate that HSP significantly outperforms the other pooling techniques in every evaluated metric, highlighting its effectiveness in handling complex DNA sequence data.

Table 2. Comparative performance metrics across different pooling techniques.

Model	**Metric**						
	Accuracy	**AUC-PR**	**MCC**	**Precision**	**Recall (sensitivity)**	**Specificity**	**F1-score**
CNN-max pooling	0.1045	0.0940	0.1039	0.0082	0.0286	0.9992	0.0108
CNN-average pooling	0.9384	0.9752	0.9384	0.9337	0.9365	0.9999	0.9252
CNN-mixed pooling	0.7819	0.7729	0.7818	0.7537	0.7583	0.9998	0.9298
CNN-horizontal max pooling	0.6696	0.6272	0.6694	0.6101	0.6369	0.9997	0.5896
CNN-HSP	0.9594	0.9851	0.9594	0.9555	0.9581	0.9999	0.9507

HSP achieves a high accuracy of 0.9594, along with the highest AUC-PR score of 0.9851, which underscores its exceptional ability to balance precision and recall. This high AUC-PR indicates that HSP is highly effective at correctly identifying DNA sequences that belong to specific fish species while minimizing false positives. The model's high sensitivity score of 0.9581 further reinforces its ability to accurately detect true positives, meaning it is particularly adept at capturing the subtle genetic variations that distinguish one species from another. These results emphasize the critical role of advanced pooling techniques like HSP in enabling the model to extract the most relevant features from complex, imbalanced datasets like the fish DNA sequences used in this study.

5. Conclusions

The study introduced a novel HSP as a feature extraction technique tailored for DNA sequence classification within CNN architectures.

The results confirm that HSP significantly outperforms traditional pooling techniques by reducing the number of parameters and loss while improving accuracy. This implies that the HSP technique is more effective than conventional pooling techniques in feature extraction and handling imbalanced datasets, making it a dominant tool for DNA sequence analysis. Furthermore, the successful implementation of HSP lays the groundwork for advancements in various fields, including health care, aquaculture, local government, and education, where large volumes of DNA data require efficient extraction, analysis, and interpretation.

Future research will focus on exploring variations of HSP and fine-tuning strategies to further optimize pooling techniques for specific types of genomic data, potentially aiming to identify optimal configurations tailored for complex biological datasets.

Acknowledgment

The authors gratefully acknowledge the financial assistance provided by the Philippine Government, facilitated by the Commission on Higher Education (CHED), which was instrumental in the successful completion of the study.

References

[1] Agarwal K. and Dixit M., “Scrupulous SCGAN Framework for Recognition of Restored Images with Caffe based PCA Filtration,” *The International Arab Journal of Information Technology*, vol. 21, no. 1, pp. 107-116, 2024. https://doi.org/10.34028/iajit/21/1/10

[2] Bera S. and Shrivastava V., “Effect of Pooling Strategy on Convolutional Neural Network for Classification of Hyperspectral Remote Sensing Images,” *IET Image Process*, vol. 14, pp. 480-486, 2020. https://doi.org/10.1049/iet-ipr.2019.0561

[3] Blagus R. and Lusa L., “SMOTE for High-Dimensional Class-Imbalanced Data,” *BMC Bioinformatics*, vol. 14, no. 106, pp. 1-16, 2013. https://doi.org/10.1186/1471-2105-14-106

[4] Boureau Y., Ponce J., and LeCun Y., “A Theoretical Analysis of Feature Pooling in Visual Recognition,” *in Proceedings of the 27th International Conference on International Conference on Machine Learning*, Haifa, pp. 111-118, 2010. https://www.di.ens.fr/willow/pdfs/icml2010b.pdf

[5] Chawla N., Bowyer K., Hall L., and Kegelmeyer P., “SMOTE: Synthetic Minority Over-Sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, no.1, pp. 321-357, 2002.

[6] Dang T. and Vu T., “Sequence-based Protein-Protein Interaction Prediction Using Multi-Kernel Deep Convolutional Neural Networks with Protein Language Model,” *bioRxiv*, pp. 1-15, 2024. https://doi.org/10.1101/2023.10.03.560728

[7] Dong J., Jiang M., Hu L., and He Z., “Hamming Encoder: Mining Discriminative k-mers for Discrete Sequence Classification,” *arXiv Preprint*, vol. abs/2310.10321, pp. 1-13, 2023. https://doi.org/10.48550/arXiv.2310.10321

[8] El-Tohamy A., Maghwary H., and Badr N., “A Deep Learning Approach for Viral DNA Sequence Classification Using Genetic Algorithm,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 8, pp. 530-538, 2022. DOI:10.14569/IJACSA.2022.0130861

[9] Feng S., Zhou H., and Dong H., “Application of Deep Transfer Learning to Predicting Crystal Structures of Inorganic Substances,” *Computational Materials Science*, vol. 195, pp. 110476, 2021. https://doi.org/10.1016/j.commatsci.2021.110476

[10] Gunasekaran H., Ramalakshmi K., Arokiaraj A., Kanmani S., and Venkatesan C., “Analysis of DNA Sequence Classification Using CNN and Hybrid Models,” *Computational and Mathematical Methods in Medicine*, vol. 2021, no. 1835056, 2021. https://doi.org/10.1155/2021/1835056

[11] Hinton G., Srivastava N., Krizhevsky A., Sutskever I., and Salakhutdinov R., “Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors,” *arXiv Preprint*, vol. arXiv:1207.0580, pp. 1-18, 2012. https://doi.org/10.48550/arXiv.1207.0580

[12] Jena M., Mishra S., and Mishra D., “Empirical Analysis of Activation Functions and Pooling Layers in CNN for Classification of Diabetic Retinopathy,” *in Proceedings of the International Conference on Applied Machine Learning*, Bhubaneswar, pp. 34-39, 2019. DOI:10.1109/ICAML48257.2019.00014

[13] Kingma D. and Ba J., “Adam: A Method for Stochastic Optimization,” *arXiv Preprint*, vol. arXiv:1412.6980, pp. 1-15, 2014. https://doi.org/10.48550/arXiv.1412.6980

[14] Koo P. and Eddy S., “Representation Learning of Genomic Sequence Motifs with Convolutional Neural Networks,” *PLoS Computational Biology*, vol. 15, no. 12, pp. 1-17, 2019. https://doi.org/10.1371/journal.pcbi.1007560

[15] LeCun Y., Bottou L., Bengio Y., and Haffner P.,

"Gradient-Based Learning Applied to Document Recognition," *IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998. DOI:10.1109/5.726791

[16] LeCun Y., Bengio Y., and Hinton G., "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015. DOI:10.1038/nature14539

[17] Lin M., Chen Q., and Yan S., "Network in Network," *arXiv Preprint*, vol. arXiv:1312.4400, pp. 1-10, 2014. https://doi.org/10.48550/arXiv.1312.4400

[18] Liu X., Wang M., and Li A., "PhosVarDeep: Deep-Learning Based Prediction of Phospho-Variants Using Sequence Information," *PeerJ*, vol. 10, pp. 1-18, 2022. DOI:10.7717/peerj.12847

[19] Mohamed E., Gaber T., Karam O., and Rashed E., "A Novel CNN Pooling Layer for Breast Cancer Segmentation and Classification from Thermograms," *PLOS ONE*, vol. 17, no. 10, pp. 1-18, 2022. DOI:10.1371/journal.pone.0276523

[20] Mohammed K., Boyapati S., Kandimalla M., Kavati M., and Saleti S., "A Comparative Analysis of the Evolution of DNA Sequencing Techniques along with the Accuracy Prediction of a Sample DNA Sequence Dataset using Machine Learning," *in Proceeding of the 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing*, Nagpurpp, pp. 1-5, 2023. DOI:10.1109/PCEMS58491.2023.10136116

[21] More Y., Dumbre K., and Shiragapur B., "Horizontal Max Pooling a Novel Approach for Noise Reduction in Max Pooling for Better Feature Detect," *in Proceedings of the International Conference on Emerging Smart Computing and Informatics*, Pune, pp. 1-5, 2023. DOI:10.1109/ESCI56872.2023.10099648.

[22] Parhami P., Fateh M., Rezvani M., and Rokny H., "A Benchmarking of Deep Neural Network Models for Cancer Subtyping Using Single Point Mutations," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 1-14, 2022. https://doi.org/10.1101/2022.07.24.501264

[23] Passricha V., and Aggarwal R., "A Comparative Analysis of Pooling Strategies for Convolutional Neural Network Based Hindi ASR," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 675-691, 2020. https://doi.org/10.1007/s12652-019-01325-y

[24] Soffer S., Ben-Cohen A., Shimon O., Amitai M., and Greenspan H., "Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide," *Radiology*, vol. 290, no. 3, pp. 590-606, 2019. DOI: 10.1148/radiol.2018180547

[25] Soliman N., Abd-Alhalem S., Ismaiel N., and El-Samie F., "An Improved Convolutional Neural Network Model for DNA Classification," *Computers, Materials and Continua*, vol. 70, no. 3, pp. 5907-5927, 2022. DOI:10.32604/cmc.2022.018860

[26] Struhl K. and Segal E., "Determinants of Nucleosome Positioning," *Nature Structural and Molecular Biology*, vol. 20, no. 3, pp. 267-273, 2013. https://doi.org/10.1038/nsmb.2506

[27] Sun S., Hu B., Yu Z., and Song X., "A Stochastic Max Pooling Strategy for Convolutional Neural Network Trained by Noisy Samples," *International Journal of Computers, Communications and Control*, vol. 15, no. 1, 2020. DOI:10.15837/ijccc.2020.1.3712

[28] Voytetskiy A., Herbert A., and Poptsova M., "Graph Neural Networks for Z-DNA Prediction in Genomes," *in Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, Las Vegas, pp. 3173-3178, 2024. DOI:10.1109/BIBM55620

[29] Wisesty U., Purwarianti A., Pancoro A., Chattopadhyay A., Phan N., and Chuang E., "Join Classifier of Type and Index Mutation on Lung Cancer DNA Using Sequential Labeling Model," *IEEE Access*, vol. 10, pp. 9004-9021, 2022. DOI:10.1109/ACCESS.2022.3142925

[30] Yu C., Hung P., Hong J., and Chiang H., "Efficient Max Pooling Architecture with Zero-Padding for Convolutional Neural Networks," *in Proceedings of the IEEE 12th Global Conference on Consumer Electronics*, Nara, pp. 747-748, 2023. DOI:10.1109/GCCE59613

[31] Yu D., Wang H., Chen P., and Wei Z., "Mixed Pooling for Convolutional Neural Networks," *Rough Sets and Knowledge Technology*, vol. 8818, pp. 364-375, 2014. https://doi.org/10.1007/978-3-319-11740-9_34

[32] Zafar A., Aamir M., Mohd Nawi N., Arshad A., and Riaz S., "A Comparison of Pooling Methods for Convolutional Neural Networks," *Applied Sciences*, vol. 12, no. 17, pp. 8643, 2022. https://doi.org/10.3390/app12178643

[33] Zhao L. and Zhang Z., "A Improved Pooling Method for Convolutional Neural Networks," *Scientific Reports*, vol. 14, no. 1, pp. 1589, 2024. https://doi.org/10.1038/s41598-024-51258-6

Lilibeth Coronel is an Associate Professor in the Department of Information Technology at Mindanao State University at Naawan, Philippines. Currently pursuing her Doctor in Information Technology at the Technological Institute of the Philippines-Quezon city, Philippines under the supervision of Arnel Fajardo. Obtained her Master's in Information Technology at the University of Science and Technology in Southern Philippines last 2014 and her Bachelor's Degree in Computer Science at AMA Computer College, Philippines last 2001. Her research interest includes Image Processing, Deep Learning, and Bioinformatics.

Arnel Fajardo is a Professor in the College of Computing Science, Information and Communication Technology at Isabela State University, Philippines. Obtained his Ph.D. in Computer Engineering at Hanbat National University, Daejeon, Korea last 2011, Master's Degree in Computer Science at De La Salle University, Manila, Philippines last 1993, and Bachelor's Degree in Electrical Engineering at Mapua University, Manila, Philippines last 1986. His field of expertise and research areas include Artificial Intelligence, Speech Recognition, Image Processing, Smart Agriculture, and Engineering Education.

Ruji Medina is a Dean and Professor in the Graduate Programs at the Technological Institute of the Philippines-Quezon City, Philippines. Obtained his PhD in Environmental Engineering at the University of the Philippines Diliman, Quezon City, Philippines last 2015. Graduated Summa Cum Laude with his Master's Degree in Environmental Engineering at Mapua University, Manila, Philippines last 2002, and completed his Bachelor's Degree in Chemical Engineering at the University of the Philippines Diliman, Quezon City, Philippines last 1992.