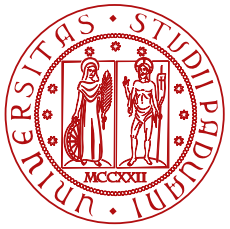

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in Control Systems Engineering

Deep Learning Barcoding for Improved DNA Sequence Classification

Master Candidate

Yazzed Hussein Younis Abdalla

Student ID 2043181

Supervisor

Prof. Nanni Loris

University of Padova

Co-supervisor

Academic Year
2023/2024

To the resilient people of Palestine and Sudan, who have faced countless challenges with unwavering strength and determination. Your perseverance and courage serve as an inspiration to us all. Your struggles will not be forgotten, and I stand in solidarity with you in the fight for justice and freedom.

To my brother, mother and family who have supported and encouraged me throughout this journey.

Abstract

The introduction of DNA barcoding has significantly transformed species identification, utilizing short DNA sequences from a uniform region of the genome. However, the conventional methods of DNA barcoding often depend on manual analysis and struggle with precise classification of species that are closely related. In our research, we introduce an innovative approach to DNA barcoding classification using deep learning techniques and sophisticated neural networks. This method leverages the capabilities of deep learning algorithms to improve the accuracy and efficiency of identifying species, especially in taxonomically complex groups. This approach has demonstrated more than 90 percent accuracy in classifying both simulated and real datasets. Despite the exceptional results deep learning brings to species classification using DNA sequences, implementing it remains challenging. The deep barcoding model we developed holds promise as a tool for species classification and can significantly contribute to understanding DNA barcode-based species identification processes.

Contents

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

DNA Deoxyribonucleic Acid

RNA Ribonuclecic Acid

COI cytochrome c oxidase subunit I

(rbcL ribulose-bisphosphate carboxylase

ITS Internal Transcribed Spacer

(IBOL International Barcode of Life

CBOL Consortium for the Barcode of Life

KNN k-nearest Neighbor

SVM Support Vector Machine

DT Decision Tree

NB Naïve Bayes

MLP Multilayer Perceptron

RF Random Forest

CNN Convolutional Neural Network

NGS Next-Generation Sequencing

WGS Whole Genome Sequencing

(WES whole exome sequencing

MEGA Molecular Evolutionary Genetics Analysis

RGB Red Green Blue

Adam Adaptive Moment Estimation

SGD Stochastic Gradient Descen

AdaGrad Adaptive Gradient

RMSProp Root Mean Square Propagation

ROC Receiver Operating Characteristic

TPR True Positive Rate

(FPR False Positive Rate

SNP singlenucleotide polymorphism

BLAST Basic Local Alignment Search Tool

1 Introduction

Biodiversity loss is a critical global environmental issue, and ecologists are continuously developing strategies to conserve biological diversity and protect natural resources. However, accessing comprehensive taxonomic information about the biosphere is a significant challenge due to the taxonomic impediment. This issue makes it difficult for researchers, especially those not specializing in taxonomy, to access and understand taxonomic data. To overcome this, the use of genetic information, particularly DNA sequences, has been proposed as a solution. [36].

The classification of DNA sequences is an essential aspect of bioinformatics, with a significant impact on scientific research and practical applications. This thesis explores the latest developments in the intersection of deep learning and DNA barcoding, which promises to revolutionize the way we comprehend genetic materials. Deep learning techniques have enabled us to interpret complex biological data more accurately and precisely than ever before. As a result, it has opened new avenues for scientific research in the field of bioinformatics.[16]

DNA barcoding was introduced in 2003 as a method to identify different species. It uses a segment of the mitochondrial cytochrome c oxidase subunit I (COI) gene as a DNA marker to provide enough information to categorize a specimen into a specific species. This method has proven to be effective in species classification and has expanded to include other genetic markers like the chloroplast ribulose-bisphosphate carboxylase gene (rbcL) and maturase K (matK) for plants, and internal transcribed spacers (ITSs) for fungi. The creation of a comprehensive reference sequence database makes DNA barcoding an efficient tool for accurately identifying species from unknown samples.. [35]

Traditional taxonomists typically rely on physical characteristics to identify species. However, in cases where this method proves difficult, DNA Barcoding can be a valu-

able solution. This technique allows for the identification of species, even in instances where specimens are incomplete, damaged, or underdeveloped. DNA Barcoding is accomplished by analyzing a short gene sequence from small tissue samples. This technique has been globally adopted for species identification due to its high variability, even among closely related species. Initiatives such as the International Barcode of Life project (IBOL) and the Consortium for the Barcode of Life (CBOL) have played a significant role in promoting DNA Barcoding as a standard. These initiatives aim to establish an online, freely accessible sequence database.[27]

Various methods for species identification using DNA barcodes have been developed. These include tree-based taxonomic methods such as neighbor-joining, similarity-based methods like BLAST, character-based methods such as BLOG, and machine learning-based methods, machine learning techniques such as Support Vector Machine (SVM), k-nearest Neighbor (KNN), Decision Tree (DT), Naïve Bayes (NB), Multilayer Perceptron (MLP), Random Forest (RF), and hierarchical supervised classifiers.

Accurately classifying DNA sequences is a significant challenge in various fields, from medical diagnostics to evolutionary biology. Traditional methods often struggle with limitations in accuracy, speed, and scalability. However, the application of deep learning algorithms has the potential to overcome these barriers, offering a more robust and efficient approach. This thesis investigates the use of deep learning techniques in the realm of DNA barcoding, a method employed to identify species based on genetic sequences. By utilizing advanced neural networks, this research aims to improve the precision and efficiency of DNA sequence classification.

This research holds great significance beyond academic curiosity. It has practical implications in various areas, such as disease control where rapid and accurate identification of pathogens is crucial. In environmental studies, it helps in monitoring biodiversity and preserving ecosystems. Additionally, in the medical field, it improves our understanding of genetic disorders and aids in personalized medicine.

This thesis is divided into several sections. First, it provides a detailed overview of the current status of DNA sequence classification, including its challenges and opportunities. Next, it explores the principles of deep learning and its relevance and applicability in bioinformatics. The main focus of the thesis is on a new approach to DNA barcoding that uses deep learning techniques. The methodology, experimental design, and data analysis are all detailed in this section. The results of this advanced method are compared with traditional classification approaches. Finally, the thesis concludes with a reflection on the findings and their implications for future research and practical applications.

With the advancements in bioinformatics, this research aims to explore the potential of deep learning in improving DNA sequence classification. By leveraging this technology,

we can further push the boundaries of genetic analysis, ultimately leading to a better understanding of genomics.

DNA barcode classification involves creating a reference library of DNA-barcoded specimens from known species. This reference library is then used to classify a query set of unknown species using DNA barcode sequences. The query set is transformed into a supervised learning dataset, with the reference set serving as the training set. The training set contains labeled specimens of known species, and the testing set includes specimens of unknown species for classification. The model is trained using the training set, and its performance is evaluated using the testing set to determine the classification accuracy.

Deep learning has achieved remarkable progress in machine learning, especially in various bioinformatics applications such as image processing, survival analysis, signal processing, and sequence analysis. Among different types of deep learning models, Convolutional Neural Network (CNN) has shown outstanding performance in prediction tasks, particularly in image and sequence processing. CNNs are designed to learn recognition tasks directly from data during training, eliminating the need for hand-crafted feature extraction or preprocessing by experts. They use features like local connectivity, parameter sharing, pooling, and multiple layers to effectively capture both local and global features from the training data.[2]

2

Related Work

This section critically examines the intersection of deep learning techniques with DNA barcoding in DNA sequence classification, highlighting the evolution from traditional molecular methods to advanced computational models. This shift signifies a major change in genomics research, driven by the emergence of big data from next-generation sequencing technologies.

Advances in Genomic Data Analysis

The advent of next-generation sequencing (NGS) technologies has led to an exponential increase in genomic data. This includes complex datasets like the human genome, which comprises over 3 billion base pairs, pushing genomics towards a data-driven science. Techniques such as whole genome sequencing (WGS), whole exome sequencing (WES), transcriptomic, and proteomic profiling have led to a rapid accumulation of omics data. This necessitates the use of bioinformatics and machine learning tools in areas like genotype-phenotype correlation, biomarker identification, gene function prediction, and mapping biomedically active genomic regions[1].

Deep Learning in DNA Barcoding

A pivotal study in this field proposed the 'DeepBarcoding' framework for species classification using DNA barcodes. This approach utilizes raw sequence data, represented as one-hot encoding in a one-dimensional image, and employs a deep convolutional neural network combined with a fully connected deep neural network for sequence analysis. Figure 1 shows a simple architecture of CNN. This study exemplifies the innovative use of deep learning in enhancing the accuracy and efficiency of DNA barcoding[25]. The classification of DNA sequences has been significantly improved by various machine-learning techniques. These methods are crucial in biomedical data analysis, particularly in identi-

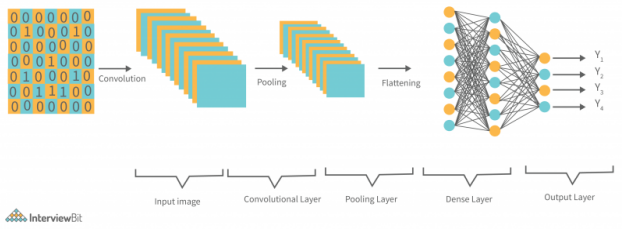

Figure 2.1: CNN Architecture

fying and classifying viruses, an essential task in preventing outbreaks like COVID-19.

Challenges in DNA Barcode Sequence Analysis

A specific application of DNA barcoding in fish classification using deep learning has been investigated, highlighting the challenges posed by the high dimensionality of DNA barcode sequences and the limitations in the number of fish species. These challenges make it difficult to analyze DNA sequences and accurately classify fish from different families[17].

Gathering DNA sequences from a variety of biological and genomic databases, including global repositories like GenBank and regional databases. This step often involves selecting sequences from diverse species, habitats, and biological conditions to create a comprehensive dataset.

- Preprocessing: Involves quality control measures like filtering out low-quality or ambiguous sequences, error correction algorithms, and sequence alignment. Sequence normalization, such as one-hot encoding of nucleotides, is employed to convert biological sequences into a format suitable for deep learning algorithms, ensuring consistency and comparability across the dataset.

- Identification of Relevant Features: Utilizing CNNs to detect patterns and signatures within the DNA sequences. Features can include sequence motifs, GC content, k-mer frequencies (the occurrence of k-length subsequences), and other genomic markers. Optimization of CNN Architecture: Select the appropriate architecture for CNN, including the number and size of convolutional filters, the depth of the network, and pooling layers to reduce dimensionality and activation functions. The architecture is tailored to capture the inherent complexity and variability of genomic sequences effectively[31].

- Model Training and Validation: Involves feeding the feature-rich DNA sequences into the CNN models. The training process employs algorithms like backpropagation and stochastic gradient descent to optimize the network weights. Hyperparameters such as learning rate, batch size, and the number of training epochs are fine-tuned to achieve the best model performance. Validation and Cross-Validation: Utilizing techniques like k-fold cross-validation to assess the model's performance on unseen data. This step is crucial in determining the model's ability to generalize beyond the training dataset and in preventing overfitting.

- DNA Sequence Classification: The trained models are then used to classify DNA sequences such as species, functional groups, or phylogenetic classifications. This

involves running new, unseen sequences through the model and interpreting the output.

- Evaluation Metrics and Analysis: The model's performance is evaluated using metrics like accuracy, precision, recall, and F1 score. Confusion matrices are also used to understand the model's classification behavior across different categories [26].

- Iterative Model Refinement and Adaptation: Based on the performance metrics, the models undergo iterative refinements, including retraining with augmented datasets, tuning hyperparameters, or adjusting the network architecture.

This review underscores the significant impact of deep learning and machine learning on DNA sequence classification, particularly in the context of DNA barcoding. The integration of these technologies marks a major advancement in genomic analysis. However, ongoing research is required to address existing challenges and fully exploit their potential in bioinformatics.

3 Background

3.1 Data

3.1.1 Real Datasets

Specific public empirical datasets, accessible via GenBank Nucleotide Database, were chosen based on set criteria. **(i) sequences exhibit a broad range of phylogenetic diversity.**
The inclusion of a wide range of phylogenetic diversity in DNA sequencing studies is essential to avoid knowledge gaps in our understanding of the genome across different populations.

(ii) they present a challenge in identification due to minimal inter-species sequence variation;
Analyzing minimal inter-species sequence variation is important because it provides insights into closely related species or strains, which are often difficult to distinguish. Full-length 16S rRNA gene sequencing, for example, can resolve subtle nucleotide substitutions between intragenomic copies of the gene, providing species and strain-level taxonomic resolution. This is significant for understanding bacterial communities and can impact areas such as microbiome analysis and infectious disease research[18].

(iii) they encompass various genomic regions.
Studying various genomic regions in DNA sequencing is important because it provides a more comprehensive understanding of the genetic makeup and medical needs of an individual. Sequencing genomes can reveal which stretches of DNA contain genes that code for important proteins or regulatory data, controlling gene activation. This holistic

view is crucial for advancing personalized medicine and disease prevention

These criteria guided the selection of six distinct datasets, detailed in Table 1.

1.Cypraeidae: This dataset includes 2,008 DNA barcode sequences, each 618 bases long, from 211 species of Cypraeidae, a well-researched group of marine gastropods. Notably, 112 species in this collection are represented by four or more sequences.[26]

2.Drosophila: Renowned for high within-species genetic divergence, this dataset contains 615 DNA barcode sequences from 19 Drosophila species. Each sequence is 663 bases long, and 15 species are represented by more than five sequences.[7]

3.Inga: Featuring species from the tropical leguminous tree genus Inga, this dataset includes 908 DNA barcodes, each 1,838 bases long, from 56 species. Notably, 35 species have over five sequences represented.[10]

4.Bats: Comprising 839 barcode sequences from 96 species, this dataset represents various species within the mammalian order Chiroptera.[29]

5.Fishes: This dataset includes 626 recent barcode sequences from 82 fish species, sourced from the GenBank Nucleotide Database and representing the broader group of fishes within the animal kingdom.[8]

6.Bird: With 1,623 barcode sequences from 150 species, each sequence ranging between 648 and 690 nucleotides, this dataset was provided by the CBOL at the 2007 Conference.[15]

Table 3.1: Summary of the Real Datasets

Dataset	#sequences	Seq. length	#species	Gene region(s)
Cypraeidae	2,008	614	211	COI
Drosophila	615	663	19	COI
Inga	908	1,838	56	trnD, ITS
Bats	839	657	96	COI
Fishes	626	719	82	COI
Birds	1,623	691	150	COI

Legend: #sequences = number of dataset sequences comprised in the dataset; Seq. length = length of the sequences; #species = number of species in the datasets; Gene Region(s) = gene region(s) used as Barcode for each dataset;

3.1.2 Simulation Data

"Authentic DNA Barcode data sets were sourced from the website of the Institute for System Analysis and Computer Science http://dmb.iasi.cnr.it/supbarcodes.php. Additionally, we utilized the Coalescent package in Mesquite software for the generation of simulated data[34]. This simulation process was grounded in the assessment of species

divergence times and the effective population size (Ne), defined as the count of individuals in a species population actively contributing genetic material to future generations.

Our methodology involved the initial step of simulating gene trees using the Yule coalescence model[34]. These simulations encompassed effective population sizes of 1,000, 10,000, and 50,000 individuals, thereby creating datasets representing 50 species, each comprising 20 individuals. For thoroughness and accuracy, each simulation was conducted 100 times.

As the effective population size expanded, the complexity of the dataset correspondingly increased. Subsequently, simulated DNA Barcode sequences on these generated gene trees. The length of these sequences was maintained at 650 bases, aligning with the standard length observed in real DNA Barcodes." The next table presents simulation data information.

Table 3.2: Summary of the Simulation Data.

Dataset	N_e	**Individual**[1]	**Seq. length**[2]	**Species**[3]
N_e1000	1000	20	650	50
N_e10000	10000	20	650	50
N_e50000	50000	20	650	50

Legend: [1]Individual indicates number of sequences for each species; [2]Seq. length is the length of the sequences; [3]Species represents a number of species (i.e., classes).

3.2 MEGA

MEGA (Molecular Evolutionary Genetics Analysis) version 11 is a comprehensive suite for analyzing DNA and protein sequence data from species and populations. It includes tools for alignment, testing evolutionary hypotheses, inferring phylogenetic trees, and estimating divergence times.

In the alignment process, MEGA11 offers an advanced and user-friendly approach. The software provides options for sequence alignment, which are essential for the subsequent analyses of sequence variation and phylogeny construction. Here are the key features related to the alignment and analysis process in MEGA11[33]:

1. Comprehensive Toolset:

   MEGA11 contains a broad collection of computational methods and tools for molecular evolutionary genetics analysis, allowing for building time trees using rapid relaxed-clock methods, and methods for estimating divergence times with probability densities for calibration.

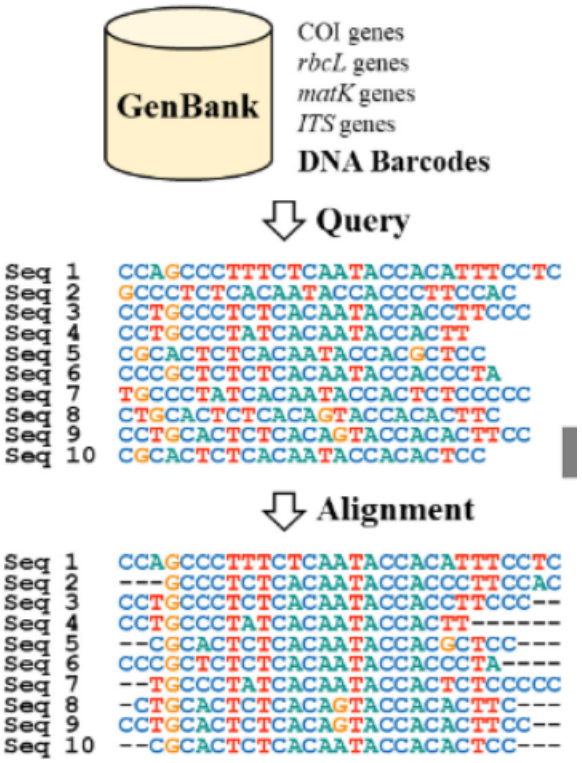

Figure 3.1: sequence alignment

2. Node Calibration:

   The Node Calibration Editor in MEGA's RelTime method facilitates the selection of calibration points on the phylogenetic tree, allowing users to choose individual node calibrations and probability densities.

3. Tip Dating Analysis:

   MEGA11's Tip Dating Wizard aids in setting up RTDT analysis, prompting users to specify outgroups and sample times, which is crucial for estimating branch lengths in the timetree.

4. Tree Exploration:

   The Tree Explorer feature is enriched with interactive exploration and customization tools. The new toolbar enhances accessibility, providing instant formatting and facilitating the visualization of evolutionary events.

Alignment of DNA sequences is a crucial step that needs to be done before classifying them using Convolutional Neural Networks (CNNs). This is important for several reasons. Firstly, alignment ensures that homologous sequences are compared at corresponding positions, which is necessary to achieve the consistent input that is required by CNNs. Secondly, since sequences obtained from different association numbers in GenBank can have varying lengths, alignment helps to standardize the lengths of these sequences, making it easier to compare them.[22]. see the alignment in the image[3.1].

Effective learning and feature extraction, such as conserved motifs and sequence variations, are possible when sequences are properly aligned. Aligning sequences helps to eliminate gaps and mismatches, reducing noise and improving CNN's classification accuracy. Our work begins with aligning the training data set by using muscle alignment. After selecting all the sequences, we insert our test data file and align it on our pre-aligned training data with the same training options to ensure that both files have the same sequence length.

3.3 Definition of the architecture of a neural network

[13] [20]Convolutional Neural Networks (CNNs) are a type of deep neural network that is primarily used for analyzing visual imagery. They are also referred to as shift-invariant or space-invariant artificial neural networks, which is due to their shared-weights architecture and translation invariance characteristics. The most significant advantage of CNNs is their ability to learn spatial hierarchies of features from input images automatically and adaptively. CNNs, or Convolutional Neural Networks, are a type of artificial neural network designed to minimize preprocessing requirements. They consist of neurons that

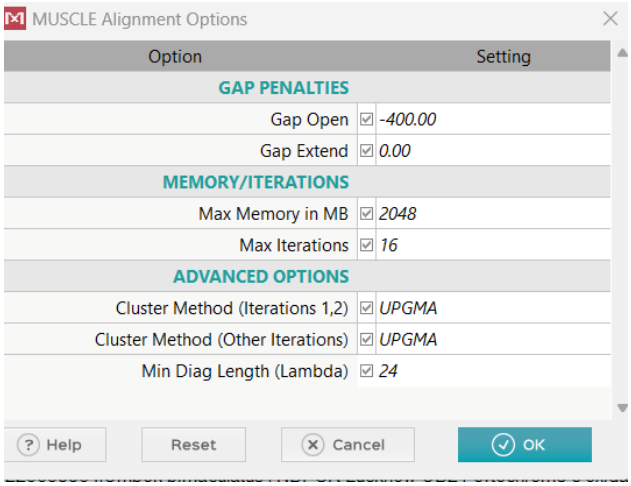

Figure 3.2: Mega Settings Options

have learnable biases and weights. Each neuron receives multiple inputs, computes a weighted sum, applies an activation function, and outputs a response.

The architecture of CNNs mimics the connectivity pattern of neurons in the human brain, specifically the visual cortex. The receptive field, which is the area in the visual field that a neuron responds to, is restricted for each cortical neuron. However, the receptive fields of different neurons partially overlap, thus covering the entire visual field.

3.3.1 Image Input Layer

A `imageInputLayeris` where we specify the size of the image, for instance, 28x28x1. These numbers correspond to the height, width, and size of the channel. The encrypted data consists of grayscale images, so the channel size (color channel) is 1. For a color image, the channel size is 3, corresponding to RGB values.

3.3.2 Convolutional Layer

In the convolutional layer, the first argument is `filterSize`, which is the height and width of the filters that the training function uses when scanning images. for example, the number 3 indicates that the filter size is 3x3. we can specify different sizes for the height and width of the filter. The second argument is the number of filters, `numFiltersi`.e. the number of neurons that connect to the same region of the input. This parameter determines the number of feature maps. Use the name-value pair `'Padding'` to add a fill to the input feature map. For a convolutional layer with a default stride of 1, padding `'same'` ensures that the output spatial dimension is the same as the input spatial

dimension. we can also define the learning pace and speeds for this level using the name-value pair arguments of `convolution2dLayer`.

3.3.3 Batch Normalization Layer

Batch normalization layers normalize the activations and gradients that propagate through a neural network, making neural network training a simpler optimization problem. Use batch normalization layers between convolutional layers and nonlinearities, such as ReLU layers, to speed up neural network training and reduce sensitivity to neural network initialization. `batchNormalizationLayer`to create a batch normalization layer.

3.3.4 Maximum Pooling Layer

Convolutional layers (with activation functions) are sometimes followed by a subsampling operation that reduces the spatial dimension of the feature map and removes redundant spatial information. Subsampling allows you to increase the number of filters in deeper convolutional layers without increasing the amount of computation required for each layer. One way to do subsampling is to use a maximum pooling, which you can create using `maxPooling2dLayer`. The maximum pooling level returns the maximum values of the input rectangular regions, specified by the first argument poolSize. In this example, the size of the rectangular region is [2,2]. The name-value pair argument 'Stride ' specifies the step size that the training function uses when scanning the input.

3.3.5 Fully Connected Layer

The convolutional and subsampling layers are followed by one or more fully connected layers. As the name suggests, a fully connected layer is a layer in which neurons connect to all neurons in the previous layer. This layer combines all the features learned from previous layers throughout the image to identify the largest patterns. The last fully connected layer combines features to classify images. Therefore, the parameter OutputSize in the last fully connected layer is equal to the number of classes in the target data.

3.3.6 Softmax Layer

The softmax activation function normalizes the output of the fully connected layer. The output of the softmax layer consists of positive numbers that sum to one and can then be used as classification probabilities by the classification layer.

3.3.7 Classification Level

The final level is the classification level. This layer uses the probabilities returned by the softmax activation function for each input to assign the input to one of the mutually exclusive classes and calculate the loss.

3.4 Specifying training options

The learning rate is a crucial hyperparameter in the training of neural networks. It controls the size of the steps taken during optimization. An initial value of 0.001 is used as it's generally a good starting point for convergence. As training progresses, reducing the learning rate can help the model converge more smoothly. the learning rate is halved (a factor of 0.5) at specific intervals. The learning rate is reduced every 50 epochs. This periodic reduction helps in fine-tuning the network by taking smaller steps as it learns more.

The Adam optimizer is employed for loss function optimization, 'Adam' refers to the Adaptive Moment Estimation (Adam) optimizer. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent (SGD) procedure to update network weights iteratively based on training data. Adam combines the advantages of two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp)[30][19][42].see Algorithm 1

Algorithm 1 Adam Optimization Algorithm

Input: Learning rate α, parameters β_1, β_2, small number ϵ

Initialize: Time step $t \leftarrow 0$, moment vectors $m \leftarrow 0$, $v \leftarrow 0$, parameter vector θ

while stopping criterion not met **do**

$t \leftarrow t + 1$

Compute gradient g w.r.t. loss function for θ

$m \leftarrow \beta_1 \cdot m + (1 - \beta_1) \cdot g$

$v \leftarrow \beta_2 \cdot v + (1 - \beta_2) \cdot g^2$

$\hat{m} \leftarrow m/(1 - \beta_1^t)$

$\hat{v} \leftarrow v/(1 - \beta_2^t)$

$\theta \leftarrow \theta - \alpha \cdot \hat{m}/(\sqrt{\hat{v}} + \epsilon)$

end while

Output: Optimized parameters θ

An epoch is one complete presentation of the data set to be learned to the learning machine. Setting it to 150 means the entire dataset will be passed forward and backward through the neural network 150 times. Mini-batch size is the number of sub-cases of the training dataset that the network will examine before the weights are updated. A size of 32 is a balance between the efficiency of batch learning and the robustness of stochastic learning. Every 10 epochs, the network will be evaluated against the validation set to monitor its performance on data it hasn't been trained on. set verbose to true, this will show detailed progress of the training in the command window, including information about the current epoch, mini-batch, and other statistics.

Plots, 'and training-progress', Provide a graphical representation of the training progress. This will generate plots showing metrics such as accuracy and loss over each epoch, providing a visual means to monitor the training process. piecewise indicates that the learning rate changes in a piecewise-constant manner, as defined by dropFactor and dropPeriod. 'Every-epoch' means the data will be shuffled before each training epoch, which helps in preventing cycles or patterns during training that could lead to biased learning.

3.5 Sequence Representation

We represented our sequences through two approaches as follows:

one-hot encoding is a common preprocessing step in bioinformatics and machine learning for handling DNA sequences. The one-hot encoded representation converts the categorical nucleotide data (A, C, G, T)into a numerical form that can be processed by various machine learning algorithms, especially in tasks like sequence classification, mutation analysis, and pattern recognition in genomic sequences.

the length of a specific DNA sequence is determined and is assumed to be a cell array where each cell contains a DNA sequence. The barcode is an index indicating a specific DNA sequence within this array. The length function calculates the number of nucleotides in the chosen sequence. All the variables are cleared to ensure that the `one_hot_encoded` sequence starts with no prior data, avoiding any data contamination or errors from previous runs. Then we iterate over each nucleotide in the DNA sequence and Check the nucleotide at the current position, then Assign a `one_hot_encoded` vector for each nucleotide. Each case corresponds to a nucleotide. `one_hot_map` is a predefined matrix where each row is a one-hot encoded vector representing a nucleotide. For instance, `one_hot_map(1, :)` could represent 'A', `one_hot_map(2, :)` could represent 'C', and so on and then assigns a default encoding for any character not recognized as a standard nucleotide. and finally, if a DNA sequence is shorter than maxLen, the encoded sequence

Figure 3.3: 2-MER

is padded with zeros at the end to reach the desired length.

In this approach the loop iterates through the DNA sequence, considering pairs of nucleotides (bases). Since the encoding involves pairs of bases, the loop goes up to `sequence_length - 1`. then we create a pair of consecutive nucleotides from the sequence. For example, if seq is 'AGCT', the first pairDNA would be AG. `WhatPair` is set to the index of the matching pair from `TwoMerName`. For example, if pairDNA is AG, and AG is the 5th element in `TwoMerName`, `WhatPair` becomes 5. After that, we initialize a 16-dimensional vector. This vector will represent the one-hot encoding for the nucleotide pair and then set the value at the index corresponding to the nucleotide pair to its index. This creates a one-hot encoded vector, where all elements are 0 except for the one representing the current nucleotide pair. finally stores the one-hot encoded vector in the `one_hot_encoded_seq` matrix. Each row of this matrix corresponds to a pair of nucleotides from the original sequence.

4 Analysis

4.1 Performance Indicators

4.1.1 Accuracy

Accuracy measures the proportion of correctly predicted instances against the total instances in the dataset.[28][11], In the context of deep barcoding, it assesses how well the deep learning model correctly identifies species or genetic variations.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{4.1}$$

In deep barcoding, which involves identifying species or variations based on genetic sequences, high accuracy means the model reliably distinguishes between different genetic markers. It's a straightforward metric that gives a quick snapshot of model performance. However, it may not always be the best indicator, especially in imbalanced datasets where one class significantly outnumbers others.

4.1.2 Error under the ROC Curve

The ROC (Receiver Operating Characteristic) curve is a graphical representation of a model's diagnostic ability. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The 'error under the ROC curve' refers to the

area outside the ROC curve, which represents the model's misclassification rate[14][21].

$$\text{Error under the ROC Curve} = 1 - \text{Area Under ROC} \tag{4.2}$$

This metric is crucial for evaluating the trade-off between sensitivity (true positive rate) and specificity (`1 - false positive rate`). It gives a more nuanced view of model performance across different thresholds, which is essential in deep barcoding where differentiating between closely related species or genetic variations is critical.

4.1.3 Performance on Real Data

We conducted several tests on our real datasets. For tables 1, 2, and 3, we used non-aligned data (raw data). Table 1 shows the EUC for the One-Hot and Tow-Mer methods, it is the fusion (sum rule) among 20 CNNs (simply re-iterating the training)the error under the ROC curve for deep One-hot were `0.101,0.125,0.255,0,0.123,0.050` and we got `0.103,0.158,0.276,0,0.118,0.043` for Two-Mer method. Table 2 shows the mean and the standard deviation of the 20 nets (one-hot method) used in the ensemble. Table 3 presents the accuracy of our model and Table 4 shows our accuracy with aligned data through muscle alignment which is slightly better for fishes, birds inga datasets.

Table 4.1: EUC:error under the ROC curve

Dataset	One-Hot	Two-Mer
Cypraeidae	0.101	0.103
Drosophila	0.125	0.158
Inga	0.255	0.276
Bats	0	0
Fishes	0.123	0.118
Birds	0.050	0.043

4.1.4 Performance on simulated Data

Simulated DNA Barcode sequence dataset have been analyzed for the classification according to our nets with Ne equal to 50000 and the performance summarized in Table 5

Table 4.2: MEAN / STD OF 20 NETS

Dataset	EUC MEAN	STD
Cypraeidae	0.168	0.048
Drosophila	0.162	0.43
Inga	0.599	0.252
Bats	0	0
Fishes	0.349	0.446
Birds	0.796	0.288

Legend:STD, standard deviation ; EUC , :error under the ROC curve

Table 4.3: Accuracy (One-hot/Raw Data)

Dataset	accuracy
Cypraeidae	96.88
Drosophila	99.14
Inga	93.39
Bats	100
Fishes	95.50
Birds	95.58

Legend: Raw Data without alignment process

4.1.5 Comparative Analysis

In this section, we will compare our methods with previous works that used the same datasets. As shown in Table 5, the average classification accuracy for One-hot encoding with the aligned sequence is 97.98, which is higher than those obtained for the other methods. One-Hot encoding with aligned data, Deep barcoding, RF, and SVM achieved 100 percent classification accuracy on Fish datasets, indicating that the training data was completely fit for training models to test data. However, the NB and KNN classifiers provided adverse results for the Bird dataset, indicating that not all supervised learning classifiers are suitable for DNA barcode classification. Our methods showed the highest performance for Ne(50000) in both the real and simulation datasets, as shown in Table 6.

In this research, the Convolutional Neural Network (CNN) framework was utilized, incorporating four : locally connected layers, layers with shared weights, convolution and pooling layers, and multiple layers [23]. CNN represents a nonlinear, multilayered method for transforming data, capable of autonomously identifying essential features from datasets, thereby eliminating the need for manual feature identification by experts. Numerous studies have confirmed CNN's superior efficacy in tasks like image classifica-

Table 4.4: Accuracy (One-hot/Aligned Data)

Dataset	accuracy
Cypraeidae	96.88
Drosophila	99.14
Inga	93.39
Bats	100
Fishes	100
Birds	97.16

Legend: muscle alignment for DNA sequence

Table 4.5: Performane on Simulated Data

Dataset	accuracy	F1-Score
Ne(50000)	94.53	94.68

tion and recognition [12], [6]. CNN models have shown outstanding results in various sequence analysis tasks, including the analysis of noncoding RNA sequences [5], the impact of noncoding sequence variations on DNA methylation [41], the identification of DNA-binding and RNA-binding proteins through next-generation sequencing techniques [3], as well as the examination of protein [4] and genome sequences [32], [24], [37]. In light of these capabilities, the study employed CNN with deep barcoding for species categorization using DNA barcodes, achieving superior results compared to other classification methods.

During the process of training deep learning models, several issues such as overfitting and high computational complexity often arise. To overcome these challenges, various new regularization methods have been introduced to establish robust and dependable models. Our Model incorporates several of these methods, including dropout, early stopping, batch normalization, cross-validation, and data augmentation.

DNA barcoding has been developed as an effective means for species identification, utilizing short DNA sequences (such as the COI and rbcL genes) to create cost-effective, intuitive, and straightforward barcodes. These efficient short DNA barcodes have led to reduced storage requirements in databases and have enhanced the identification of species with significant economic potential [16]. Furthermore, the introduction of single-nucleotide polymorphism barcodes (SNP tags) [40] has offered easier accessibility and quicker storage capabilities [38], progressively decreasing storage costs [9]. DNA barcoding thus accelerates the discovery of new or previously unknown species [39]. Despite these advancements, achieving precise and rapid taxonomy identification and classifica-

Table 4.6: Performance Comparison on Real Data

Data		MLP	NB	DT	KNN	SVM	RF	DB	1H/R	1H/A
Cypraeidae	ACC	94.89	91.48	93.75	95.17	95.17	96.02	96.31	**96.88**	**96.59**
Drosophila	ACC	99.14	99.14	97.41	99.14	99.14	99.14	99.14	**99.14**	**99.14**
Inga	ACC	81.97	73.77	86.89	89.34	93.44	92.62	93.44	**93.39**	**95.04**
Bats	ACC	99.31	97.92	95.83	95.14	99.31	99.31	99.31	**100.00**	**100.00**
Fishes	ACC	98.20	97.30	96.40	97.30	100.00	100.00	100.00	**95.50**	**100.00**
Birds	ACC	90.54	53.63	91.02	69.72	94.64	90.54	97.48	**95.58**	**97.16**
Average		94.01	85.54	93.68	90.97	96.95	96.16	97.61	**96.74**	**97.98**

Legend: DeepBarcoding, proposed method; RF, random forest; SVM, support vector machine; KNN, k-nearest neighbor; DT, decision tree; NB, naïve Bayes; MLP, multi-layer perceptron; DB,DeepBarcoding ; 1H/R , One-Hot using raw data ; 1H/A , One-Hot using aligned data , The best performances on each dataset are in bold.

Table 4.7: Performance Comparison on Simulated Data

Data		MLP	NB	DT	KNN	SVM	RF	DB	One-Hot
Ne(50000)	ACC	86.89	88.15	91.24	90.91	93.60	93.25	94.21	**94.53**
	F1-Score	86.31	87.64	91.01	90.45	93.22	93.12	93.89	**94.68**

Legend: DeepBarcoding, proposed method; RF, random forest; SVM, support vector machine; KNN, k-nearest neighbor; DT, decision tree; NB, naïve Bayes; MLP, multi-layer perceptron; DB,DeepBarcoding ; The best performances on each dataset are in bold.

tion using DNA barcodes is still a substantial challenge. To address this, both simulated and real data have been applied in species classification using a deep barcoding model, which has demonstrated remarkable effectiveness. With these benefits, deep barcoding is poised to become a highly promising tool for future DNA barcoding applications.

5

Conclusions and Future Works

Deep barcoding is a method of supervised learning that relies on the availability and quality of well-known DNA sequences in the training set. This presents a limitation in extending the model to new species. However, this thesis presents a comprehensive method for classifying an unknown specimen into a known species category by analyzing its DNA Barcode. The method uses a deep barcoding model that processes both raw and aligned sequences. The classification outcomes of the model were benchmarked against traditional methods like MLP, SVM, etc. The analysis reveals that our model is a strong contender in successfully managing DNA Barcode species classification, achieving outstanding classification results.

In real data scenarios, the model's performance was on par with traditional DNA Barcode classification methods, whereas it showed superior results with simulated data. Despite these advancements, deep learning in DNA barcode analysis faces several challenges. Approaches like deep barcoding are pivotal for researchers to further explore DNA barcode, gradually clarifying the process of DNA barcode-based species identification.

Finally, employing a pre-trained model may be an effective solution when dealing with complex and numerous DNA sequence datasets.

References

[1] Wardah S Alharbi and Mamoon Rashid. "A review of deep learning applications in human genomics using next-generation sequencing data". In: *Human Genomics* 16.1 (2022), pp. 1–20.

[2] Babak Alipanahi et al. "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning". In: *Nature biotechnology* 33.8 (Aug. 2015), pp. 831–838. ISSN: 1087-0156. DOI: 10.1038/nbt.3300. URL: https://www.nature.com/articles/nbt.3300.pdf.

[3] Babak Alipanahi et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning". In: *Nature biotechnology* 33.8 (2015), pp. 831–838.

[4] José Juan Almagro Armenteros et al. "DeepLoc: prediction of protein subcellular localization using deep learning". In: *Bioinformatics* 33.21 (2017), pp. 3387–3395.

[5] Genta Aoki and Yasubumi Sakakibara. "Convolutional neural networks for classification of alignments of non-coding RNA sequences". In: *Bioinformatics* 34.13 (2018), pp. i237–i244.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[7] Johannes Bergsten et al. "The Effect of Geographical Scale of Sampling on DNA Barcoding". In: *Systematic Biology* 61.5 (2012), pp. 851–869. ISSN: 10635157, 1076836X. URL: http://www.jstor.org/stable/41677983 (visited on 11/23/2023).

[8] Paola Bertolazzi, Giovanni Felici, and Emanuel Weitschek. "Learning to classify species with barcodes". In: *BMC Bioinform.* 10.S-14 (2009), p. 7. DOI: 10.1186/1471-2105-10-S14-S7. URL: https://doi.org/10.1186/1471-2105-10-S14-S7.

[9] Li-Yeh Chuang et al. "A comparative analysis of chaotic particle swarm optimizations for detecting single nucleotide polymorphism barcodes". In: *Artificial Intelligence in Medicine* 73 (2016), pp. 23–33.

[10] Kyle G. Dexter, John W. Terborgh, and Clifford W. Cunningham. "Historical effects on beta diversity and community assembly in Amazonian trees". In: *Proceedings of the National Academy of Sciences* 109.20 (2012), pp. 7787–7792. DOI: 10.1073/pnas.1203523109. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1203523109. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1203523109.

[11] Warren John Ewens, Gregory R Grant, et al. *Statistical methods in bioinformatics: an introduction*. Vol. 2. Springer, 2005.

[12] Zubair Md. Fadlullah et al. "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2432–2455. DOI: 10.1109/COMST.2017.2707140.

[13] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016.

[14] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[15] Paul D N Hebert et al. "Identification of birds through DNA barcodes". In: *PLoS biology* 2.10 (2004), e312.

[16] Paul DN Hebert et al. "Biological identifications through DNA barcodes". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1512 (2003), pp. 313–321.

[17] Lina Jin et al. "Fish classification using DNA barcode sequences through deep learning method". In: *Symmetry* 13.9 (2021), p. 1599.

[18] Jethro S Johnson et al. "Evaluation of 16S rRNA gene sequencing for species and strain-level microbiome analysis". In: *Nature communications* 10.1 (2019), p. 5029.

[19] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[21] WJ Krzanowski and DJ Hand. "ROC Curves for Continuous Data Introduction". In: *ROC CURVES FOR CONTINUOUS DATA*. CRC PRESS-TAYLOR FRANCIS GROUP, 2009, pp. 1–+. URL: https://www.webofscience.com/api/gateway?GWVersion=2&SrcApp=PARTNER_APP&SrcAuth=LinksAMR&KeyUT=WOS:000267733100001&DestLinkType=FullRecord&DestApp=ALL_WOS&UsrCustomerID=a2bf6146997ec60c407a63945d4

[22] Aryan Lall and Siddharth Tallur. "Deep reinforcement learning-based pairwise DNA sequence alignment method compatible with embedded edge devices". In: *Scientific Reports* 13.1 (2023), p. 2773.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[24] Qiao Liu et al. "Chromatin accessibility prediction via a hybrid deep convolutional neural network". In: *Bioinformatics* 34.5 (2018), pp. 732–738.

[25] Giosuè Lo Bosco and Mattia Antonino Di Gangi. "Deep learning architectures for DNA sequence classification". In: *Fuzzy Logic and Soft Computing Applications: 11th International Workshop, WILF 2016, Naples, Italy, December 19–21, 2016, Revised Selected Papers 11*. Springer. 2017, pp. 162–171.

[26] Christopher P Meyer and Gustav Paulay. "DNA barcoding: error rates based on comprehensive sampling". In: *PLoS biology* 3.12 (2005), e422.

[27] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics". In: *Briefings in Bioinformatics* 18.5 (July 2016), pp. 851–869. ISSN: 1467-5463. DOI: 10.1093/bib/bbw068. eprint: https://academic.oup.com/bib/article-pdf/18/5/851/25581102/bbw068.pdf. URL: https://doi.org/10.1093/bib/bbw068.

[28] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.

[29] Takeru Nakazato and Utsugi Jinbo. "Cross-sectional use of barcode of life data system and GenBank as DNA barcoding databases for the advancement of museomics". In: *Frontiers in Ecology and Evolution* 10 (2022). ISSN: 2296-701X. DOI: 10.3389/fevo.2022.966605. URL: https://www.frontiersin.org/articles/10.3389/fevo.2022.966605.

[30] Mohamed Reyad, Amany M Sarhan, and M Arafa. "A modified Adam algorithm for deep neural network optimization". In: *Neural Computing and Applications* (2023), pp. 1–18.

[31] Riccardo Rizzo et al. "A deep learning approach to DNA sequence classification". In: *Computational Intelligence Methods for Bioinformatics and Biostatistics: 12th International Meeting, CIBB 2015, Naples, Italy, September 10-12, 2015, Revised Selected Papers 12*. Springer. 2016, pp. 129–140.

[32] Ritambhara Singh et al. "DeepChrome: deep-learning for predicting gene expression from histone modifications". In: *Bioinformatics* 32.17 (2016), pp. i639–i648.

[33] Koichiro Tamura, Glen Stecher, and Sudhir Kumar. "MEGA11: molecular evolutionary genetics analysis version 11". In: *Molecular biology and evolution* 38.7 (2021), pp. 3022–3027.

[34] Robin van Velzen et al. "DNA barcoding of recently diverged species: relative performance of matching methods". In: *PloS one* 7.1 (2012), e30490.

[35] Emanuel Weitschek, Giulia Fiscon, and Giovanni Felici. "Supervised DNA Barcodes species classification: analysis, comparisons and results". In: *BioData mining* 7 (2014), pp. 1–18.

[36] Edward O Wilson. *The diversity of life*. WW Norton & Company, 1999.

[37] Bite Yang et al. "BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone". In: *Bioinformatics* 33.13 (Feb. 2017), pp. 1930–1936. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx105. eprint: https://academic.oup.com/bioinformatics/article-pdf/33/13/1930/49040412/bioinformatics_33_13_1930.pdf. URL: https://doi.org/10.1093/bioinformatics/btx105.

[38] Cheng-Hong Yang et al. "Analysis of high-order SNP barcodes in mitochondrial D-loop for chronic dialysis susceptibility". In: *Journal of Biomedical Informatics* 63 (2016), pp. 112–119. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2016.08.009. URL: https://www.sciencedirect.com/science/article/pii/S1532046416300867.

[39] Cheng-Hong Yang et al. "Decision Tree Algorithm–Generated Single-Nucleotide Polymorphism Barcodes of rbcL Genes for 38 Brassicaceae Species Tagging". In: *Evolutionary Bioinformatics* 14 (2018). PMID: 29551885, p. 1176934318760856. DOI: 10.1177/1176934318760856. eprint: https://doi.org/10.1177/1176934318760856. URL: https://doi.org/10.1177/1176934318760856.

[40] Cheng-Hong Yang et al. "Evaluation of breast cancer susceptibility using improved genetic algorithms to generate genotype SNP barcodes". In: *IEEE/ACM transactions on computational biology and bioinformatics* 10.2 (2013), pp. 361–371.

[41] Haoyang Zeng and David K Gifford. "Predicting the impact of non-coding variants on DNA methylation". In: *Nucleic acids research* 45.11 (2017), e99–e99.

[42] Zijun Zhang. "Improved adam optimizer for deep neural networks". In: *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*. Ieee. 2018, pp. 1–2.

Acknowledgments

I would like to express my deepest gratitude to my thesis advisor, `Loris Nanni`, for their invaluable guidance, patience, and expertise. Their insights and feedback have been crucial in shaping this research.
Finally, I would like to extend my gratitude to all those who have directly or indirectly contributed to this research and to my personal and professional growth during my master's program.