

J. Inf. Commun. Converg. Eng. 21(2): 159-166, Jun. 2023

Regular paper

Classification in Different Genera by Cytochrome Oxidase Subunit I Gene Using CNN-LSTM Hybrid Model

Meijing Li¹ and Dongkeun Kim²*

¹Department of Artificial Intelligence and Data Engineering, Sangmyung University, Seoul 03016, Republic of Korea ²Department of Intelligent Engineering Informatics for Human, College of Convergence Engineering, Sangmyung University, Seoul 03016, Republic of Korea

Abstract

The COI gene is a sequence of approximately 650 bp at the 5' terminal of the mitochondrial Cytochrome c Oxidase subunit I (COI) gene. As an effective DexyriboNucleic Acid (DNA) barcode, it is widely used for the taxonomic identification and evolutionary analysis of species. We created a CNN-LSTM hybrid model by combining the gene features partially extracted by the Long Short-Term Memory (LSTM) network with the feature maps obtained by the CNN. Compared to K-Means Clustering, Support Vector Machines (SVM), and a single CNN classification model, after training 278 samples in a training set that included 15 genera from two orders, the CNN-LSTM hybrid model achieved 94% accuracy in the test set, which contained 118 samples. We augmented the training set samples and four genera into four orders, and the classification accuracy of the test set reached 100%. This study also proposes calculating the cosine similarity between the training and test sets to initially assess the reliability of the predicted results and discover new species.

Index Terms: COI gene, DNA barcode, CNN-LSTM hybrid, Species classification

I. INTRODUCTION

A. Cytochrome oxidase subunit I (COI) gene

Mitochondrial DeoxyriboNucleic Acid (DNA) is the genetic structure of mitochondria and is an important organelle that produces energy (adenosine triphosphate) for cells. Because mitochondria mainly pass through egg cells, they have strong maternal genetic characteristics and enhance the genetic specificity of the species. As shown in Fig. 1, the Cytochrome c Oxidase subunit I (COI) gene is a fragment of about 650 bp (a base pair is a basic unit of double-stranded nucleic acids consisting of two nucleobases bound to each other by hydrogen bonds) at the 5' terminal of the COI gene in mitochondrial Deoxyribonucleic Acid (DNA). The evolutionary rate of the COI gene was high, and the variation between species was generally obvious. However, within the species, the variation was relatively conserved.

Hebert conducted a series of confirmatory studies [1,2,3]; the first experiment used the COI gene to classify several species into their phyla and orders, and to classify several Lepidoptera insects into their own species; the second experiment selected about 2200 species from 11 animal phyla. After partial sequence comparison between the COI genes in intraspecific and closely related species, more than 90% of the species had significantly greater interspecific differences than intraspecific differences. The third experiment was performed on North American birds with better taxonomic studies. Most species can be distinguished by comparing their COI gene sequences.

Received 23 February 2023, Revised 21 April 2023, Accepted 8 May 2023

*Corresponding Author Dongkeun Kim (E-mail: dkim@smu.ac.kr)

Open Access https://doi.org/10.56977/jicce.2023.21.2.159

print ISSN: 2234-8255 online ISSN: 2234-8883

[©]This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

Department of Intelligent Engineering Informatics for Human, College of Convergence Engineering, Sangmyung University, Seoul, 03016, Republic of Korea

B. Related Work

Traditional species identification requires a familiarity with the morphological characteristics of multiple groups. Therefore, manual classification requires large investments in resources and time. With the development of next-generation sequencing technology, acquisition of the COI gene has become faster and easier. The COI gene is widely used as an effective DNA barcode taxonomic identification. It can greatly reduce manpower, and at the same time, it will have better performance [4] for identifying species that are difficult to distinguish, such as small insects, or a period of inconspicuous morphological features, such as larval stages. This approach will facilitate the development of species identification methods. Many related research projects have been launched, including the AII Leps Barcode of Life and Fish Barcode of Life Initiative.

The statistical method of constructing a phylogenetic tree by genetic comparison can be used to understand the evolutionary history of organisms and distinguish between species. The neighbor-joining method can determine the adjacent taxa that have the closest genetic distance [5]. The maximum likelihood method was used to select a phylogenetic tree with the most significant likelihood value. These methods require extensive computation to establish differentiation systems; therefore, they are only suitable for a limited amount of data analysis.

With the development of artificial neural networks, classification processes have become faster and more efficient. Tampuu et al. developed a ViraMiner model containing two branches based on a Convolutional Neural Network (CNN)

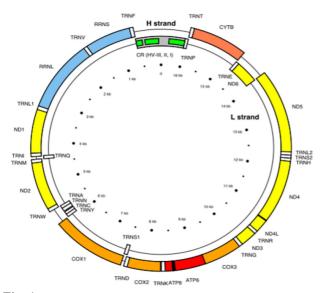


Fig. 1. Location of the *MT*- *COI* gene in the human mitochondrial genome. *MT*- *COI* is one of the three cytochrome c oxidase subunit mitochondrial genes and it is also called COX1 (https://en.wikipedia.org/wiki/Cytochrome_c_oxidase_subunit_]).

to predict the likelihood that an input DNA sequence is a virus [6]. Singh et al. utilized deep bidirectional Long Short-Term Memory (LSTM) to predict the origin of replication sequences in organisms [7]. Gunasekaran et al. used a hybrid model of CNN-LSTM for nine types of viruses: COVID, SARS, MERS, dengue, hepatitis, and influenza; the model achieved a high accuracy of 93.13% [8]. These models demonstrated that artificial neural networks perform well in the field of biological genetic information.

II. SYSTEM MODEL AND METHODS

A. Data collection and data pre-processing

We used the GenBank nucleic acid sequence database in the National Center for Biotechnology Information (NCBI) to retrieve relevant genetic information in two orders, *Rodentia* and *Lagomorpha*, and the COI gene sequences of 396 animals were randomly obtained. They contained 15 different genera of animals: *Rattus, Maxomys, Niviventer, Graomys, Eligmodontia, Phyllotis, Abrothrix, Akodon, Euneomys, Calomys, Tamias* and *Ochotona* belonging to *Rodentia* order, and *Sylvilagus, Oryctolagus, Lepus* belonging to *Lagomorpha* order. Finally, 278 and 118 samples were randomly selected as the training and test sets, respectively [9].

The one-hot encoding method can be used to encode nucleotides [10,11], so we used four types of vectors to represent *Adenosine (A)*, *Thymine (T)*, *Cytosine (C)*, *Guanine (G)*: [1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1], and [0,0,0,0] The values in the vector were considered the probabilities of the four bases at each position in the DNA sequence. We performed an operation aligned sequences of the same length (729 bp). The input vector of the CNN was a $27 \times 27 \times 4$ matrix, and that of the LSTM was a 4×729 matrix.

B. Classifier models

The K-means algorithm is a classic partition-based clustering method. The basic steps of the algorithm are as follows: (1) clustering is performed with k points in the space as centroids, (2) objects are classified in the nearest order, and (3) the value of the centroid of each cluster is updated iteratively until the best clustering result is obtained. However, clustering does not perform well when the data are unbalanced.

The Support Vector Machine (SVM) method has a positive effect on solving binary classification problems by creating a decision boundary that is the maximum-margin hyperplane. SVM parameters, such as the kernel and penalty parameters, have a significant influence on the complexity and performance of the prediction models [12]. SVM can perform nonlinear classification using the kernel method.

A CNN is a multilayer artificial neural network that uses

weight-sharing and gradient back-propagation algorithms to train the model [13]. The CNN mainly consists of input layers, a convolutional layer for kernel computation to extract features, a Rectified Linear Unit layer, a pooling layer for dimensionality reduction, a fully connected layer for combining local features for classification, and an output layer to obtain confidence scores for predicting different categories using the softmax activation function.

The LSTM network [14] can memorize values for an indefinite length of time using four unique gates, as shown in Fig. 2(a): As shown in formula (1), the forget gate limits the impact of the previous state from the present state; as shown in formulas (2) and (3), the input gate for introducing inputs, as shown in formula (4), the cell state can be updated; and as shown in formulas (5) and (6), the output gate determines the output value of this unit. In the formula (1~6), x_t is the input at time t; b_f , b_i , b_c , and b_o are the bias respectively in the forget gate, input gate, cell state update, and output gate; w_f, w_i, w_c, and w_o respectively are the network weights in forget gate, input gate, cell state update, and output gate; ft, it, and ot respectively are the results of forget gate, input gate, and output gate at time t; c_{t-1} , c_t , and \tilde{c}_t respectively are the cell state at time t-1 and time t, and the candidate cell state at time t; h_{t-1} and h_t respectively are the output at time t-1 and time t; σ is the logistic sigmoid function and tanh is the tanh function. The LSTM network retains important features through various gate functions, which can effectively slow down the gradient disappearance or explosion that may occur in long-sequence problems, and has better performance in long-sequence problems.

$$f_t = \sigma \left(w_f \cdot [h_{t-1}, x_t] + b_f \right) \tag{1}$$

$$i_t = \sigma \left(\mathbf{w}_i \cdot [h_{t-1}, x_t] + b_i \right) \tag{2}$$

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$\mathbf{c}_t = f_t * \mathbf{c}_{t-1} + i_t * \tilde{\mathbf{c}}_t \tag{4}$$

$$\boldsymbol{o}_{t} = \sigma \left(\boldsymbol{w}_{o} \cdot \left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_{t} \right] + \boldsymbol{b}_{o} \right) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

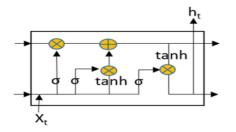


Fig. 2. Schematic diagram of cell state in LSTM. x_t is the input at time t; h_t are the output at time t; σ is the logistic sigmoid function and tanh is the tanh function; + for sum operation and [×] for multiplication operation.

C. CNN-LSTM Hybrid Models

We referred to other studies on gene classification and found that CNN are highly efficient classifiers. As described in [15], a conventional three-layer CNN model was developed to predict the effects of non-coding variants from genomic sequences only. Gene classification models do not require complex convolutional structures. Based on our experimental data, we found that the input vector of our CNN was only a $27 \times 27 \times 4$ matrix; therefore, we decided to use CNN as our gene classification selector. We further optimized the performance of the CNN by adjusting its hyperparameters and achieved an accuracy of 91% on the test set. However, CNN convolutions typically require large amounts of data for feature learning. Given the limited amount of available COI gene data, enhancing the feature-extraction ability of the classification model is critical. As we all know, gene expression at the microscopic level determines the morphology of organisms at the macroscopic level. Organisms of the same species often have similar forms, resulting in differences in the probability of gene sequence arrangements at the microscopic level. Therefore, to take advantage of this characteristic, we chose to use the LSTM network, which performs well in long-series continuous prediction. We concatenate the feature maps of the CNN and LSTM networks and feed them into a CNN for classification prediction. A high accuracy of 94% was achieved for the same test set. Our model differs from traditional statistical methods because it is highly trainable and computationally efficient. In addition, our CNN-LSTM hybrid model achieved better classification performance than the CNN alone, even with a small amount of data, without increasing the number of training samples. From a biological perspective, we also explained that the mutability of genes could cause CNN networks to suffer from performance suppression, whereas the CNN-LSTM network improved the extraction of gene features by utilizing the differences in the probability of nucleotide arrangement in the genes, thus improving the performance of the classifier.

III. RESULTS

In the K-means algorithm model, 209 samples from the training set were classified correctly and 69 were classified incorrectly. The results indicated that the classification of the training set was not effective. Although inter-genera differences in COI genes are generally greater than intra-genera differences, there is still a certain degree of conserved sequences in the genes of the different genera, at the same time, there is a certain rate of variation in the genes within

the genera. The inter- and intra-genera differences both had a significant impact on the results of this model. In the following section, we calculate the genetic distance of genes to discuss the reasons for this in depth.

Within the SVM algorithm model, which uses the linear kernel method and shows the best performance, 51 samples were correctly classified and 67 were incorrectly classified in the test set, with an accuracy rate of 43%. Owing to the uneven number of samples from various classes in the training set, overfitting the training set rendered the predictions less effective.

We compared the accuracy of the single CNN model with different hyperparameters, as listed in Tabel 1. The CNN model performed better than the other models, with 91% accuracy. For the experimental result that is genera *Graomys* and *Phyllotis* were be misclassified as *Tamias*, and *Akodon* was misclassified as *Sylvilagus*, we compute and compare *Graomys Phyllotis* and *Phyllotis* genera genetic distances in the testing set by Mega11, as shown in Fig. 3. Genetic distance refers to the degree of genetic difference between different species, in general, the genetic distance within a specie is relatively small. And as we analyzed, the genetic difference within the species can affect the accuracy of the model classification.

The average distances within Graomys genera is 0.0076, Phyllotis is 0.0075, and Tamia is 9.9188. There are large differences within the Tamia genera. The average distance between Graomys and Tamia is 6.1711. The mean inter-genus distance between Graomys and Tamia was smaller than the mean intra-genus distance of Tamia, increasing the possibility that Graomys was misclassified as Tamia during the classification process. The mean intergeneric distance between Phyllotis and Tamia was 20.6192. The distance between Phyllotis and Tamia was only twice that between Phyllotis and Tamia. In the case of incomplete feature extraction, the possibility of misclassifying Tamia could increase. The average genetic distance within Sylvilagu was 2.3636. However, the intergeneric distance between Sylvilagu and Akodon was only 1.6861. This increased the probability of Akodon being misclassified as Sylvilagu. Therefore, in the case of large intra-genera variations, the prediction results are likely to be affected, and the accuracy rate will be reduced.

Another factor that may affect the accuracy is likely caused by the CNN model. During the downsampling process, the extracted features are most likely to lose details. The encoded gene sequences were not similar to the image matrix and exhibited a strong correlation at the pixel level. As shown in Fig. 4, similar compositions of A+T and C+G bases in *Graomys* and *Tamias* or *Akodon* and *Sylvilagus* also increase the probability of misclassification.

The total number of permutations in the triplets was 64, a value well exceeding the number of amino acids (20). This indicates that many amino acids are specified by more than one codon, a phenomenon called degeneracy [16]. At the same time, morphologically close genera will produce closer gene expression, so we believe that the combination between bases is not completely random--the combination probability between different bases in triplets is different inter-genera. To take advantage of this characteristic, we used an LSTM network, which has good performance in long-series continuous prediction, such as text learning, which improves the classification ability of the network by extracting long-term sequential features. We built a CNN-LSTM hybrid model, as shown in Fig. 5. The CNN and LSTM networks were trained individually, and the feature map of the LSTM was merged with that of the CNN. Finally, the combined features are passed through the dense layer in the CNN to predict the genera. The results showed that 7 samples of Calomys were misclassified as Euneomys or Graomys, and the accuracy of the hybrid model was 94%.

We then added 17 *Calomys* samples to the training set. The number of samples in the training set was increased to 295, but the number of other genera in the training or test set remained unchanged. In addition, we retrained the model, and the prediction results of the test set were completely correct. To demonstrate the generality of the model, we used four genera: *Parambassis (Actinopteri* order), *Hygrobates* (*Trombidiformes* order), *Nephrops (Decapoda*order), *Bombus* (*Hymenoptera* order). The training set was increased to 315 samples and the test set was increased to 126. The test set was completely classified using the CNN-LSTM hybrid model.

We also performed non-feature combining; thus, the feature was first obtained through the CNN model and then passed into the LSTM network [17]. The experimental results were very similar to those obtained using a single CNN model. The LSTM network does not play an effective role. We did not increase the number of layers in the network further

Table 1. Hyperparameters of the CNN and accuracy of the test set. Contents in the table: the numbers on the left indicate the variables of the parameters, and the numbers on the right is the accuracy of the CNN network on the test set. The parameters marked in red are the final optimization parameters of the CNN

	Hyper-parameters and accuracy										
Number of kernels	32, 78%	64, 88%	128, 83%	256, 83%							
Kernel size of Convolution layer	2, 88%	3, 80%	4,78%	5, 78%							
Kernel size of Max-pooling layer	2,88%	3, 89%	4, 89%	5, 89%							
Number of Convolution layer	1,88%	2, 89%	3, 78%	4, 78%							
Coefficient of dropout	0.6, 84%	0.5, 90%	0.4, 91%	0.3, 89%							

13083 13073 1339 1339 1339 1339 1339 1339 1339 13	News News News News News News 2002 2107 22107 22107 2005 11044 1558 2375 2408 2375 2007 2375 2007 2375 2007 2375 2007 2375 2007 2375 2007 2005 2005 2005 2005 2005 2005 20	13.1196	0.2723	8	0		J0599822	KM/537944	J0599948	J0599941	10599947	JF457162	J0501447	JF457163	J0350512	J0350515	JF457164	JF457141	JF457139	JF457133	JF457138	JF457134	JF457140	GUSTOTOZ	H2966965	M2060234	M2060336	M2060248	M2060353	M2060352	M2060235	M2060279	M2060326	M2050308	M2060364	6
13.0678 13.3130	14.896		8	0.2775	0.2775	0.2712	0.2712	0.2743	0.2743	0.2743	0.2743	07220 17348 17.444 17.148 17.148 17.148 17.148 17.148 17.482 17.944 17.852 17.994 18.007 10.175 14.985 17.488 17.489 17.619 17.521 18.200 17.471 10.475 17.724 18.004 18.310 17.787 17.785	0.3105	0.3105	12785 12785 12995 12995 12995 12995 12091 13080 13080 13080 13075 18420 0.188 12783 12880 12775 11888 12775 23088 13823	12378 12378 12588 12588 12546 12541 13250 13250 13250 13270 18.184 0.166 12.016 12.003 12.403 11.201 12.403 27771 13.134	0.3202	14.9267 14.9267 15.0868 15.1076 15.1445 15.0868 13.1624 13.1641 13.1223 13.1457 11.1939 18.1274 13.4200 13.1652 13.4200 13.2863 13.4200	0.2839	173137 173137 174971 174971 174901 17459 0.2009 0.2009 0.3158 0.2009 17.9100 11.7576 18.8655 184522 18.8555	0.2839	0.2798	0.2806	132229 132229 13.5026 13.4575 13.4597 13.5252 12.9794 12.9794 13.0206 13.0019 15.0154	16.2619 16.2839 16.0885 16.0466 16.0466 16.0466 17.8811 17.6852 17.6869 17.7407	2 668 2 668 2 661 2 661 2 8429 2 8119	27296 27296 225942 225942 22852	2761 2761 2975 2975 2999 2998	27502 227502 22.9541 22.9541 22.9342 22.933	0.0124	0.0077	0.0015	0.0046	0.0015		200400 JUCKIN JOHN JOHN JOHN JOHN JUCKIN JUC
3.3130	3	13,1420 1	0.2755	0.2807	0.2807	0.2743	0.2743	0.2775	0.2775	0.2775	0.2775	17.3438 1	0.3139	0.3139	12,7695 1	12.3788 1	0.3237	14.9267 1	0.2873	17.3137 1	0.2873	0.2832	0.2839	13.2329 1	16.2839 1	2,6568 2	22,7296 2	22.7761 2	22,7502 2	0.0140	0.0093	0.0031	0.0061			N2050308, M
=	4.7509 24	3.3733 13	0.2741 (0.2793 (0.2793 (0.2730 (0.2730 (0.2761 (0.2761 (0.2761 (0.2761 (7.1414 17	0.2989 (0.2989 (2,9905 12	2,5808 12	0.3150 (5.0866 15	0.2858 (7.4971 17	0.2858	0.2817 (0.2824	3.5026 13	6.0685 16	2.8631 Z	2.9342 Z	2.9795 Z	29541 22	0.0108 (0.0093 (0.0031				ZD 5002 6 MG
3.3130 13	1.7292 24	3.3733 13	1.2678 0	0.2730 0	0.2730 0	0.2667 0	0.2667 0	0.2698 0	0.2698 0	0.2698 0	0.2698 0	7.1196 17	0.3055 0	0.3055 0	2,9905 12	2,5808 12	0.3150 0	5.1076 15	0.2791 0	7.4971 17	0.2791 0	0.2749 0	0.2759 0	3,4575 13	0.0466 16	2,8631 22	2,9342 22	2,9795 22	29541 22	0.0108 0	0.0062					2050279 MZ
13130 13	6857 24	13965 13	12574 0	12625 0	12625 0	12564 0	12564 0	0.2595 0	0.2595 0	0.2595 0	0.2595 0	.1196 17	13170 0	13170 0	9496 13	5448 12	13268 0	1445 15	0.2837 0	4601 17	0.2837 0	0.2795 0	0.2804 0	(4597 13	10466 16	8429 22	9142 22	9697 22	3342 22	0.0140						050295 MZ
.3654 23	.7292 22	.3733 15	2007 24.	2718 24	2718 24	2656 24	2656 24	2687 24	.2687 24.	2687 24	2687 24	.1632 17.	.3009 23.	.3009 23	.0111 13	6411 13	.3170 24	.0866 13.	2824 23	.4539 0.	2824 23	0.2783 23.3779 23.4030 23.3887 23.2891 16.5617 13.0257	02791 23.4921 23.5165 23.4735 23.4056 16.3994 13.3899	.5252 12	.0466 17.	.8119 0.	.8832 0.		9031							050052 MZ
0340 23	7037 22.	2314 15.:	5786 24.0	5786 24.0	5786 24.0	5568 24.	5568 24.	5786 24.	5786 24.0	5786 24.0	5786 24.0	9749 17.	9 <u>4</u> 23	944 23.	0660 13.	2502 13.1	0239 24.	1624 13	4921 23.	3209 0.3	4921 23.	3779 23.	4921 23.	9794 12	6811 17.	0.0109 0.	0.0046 0.	0.0015								50053 MZD
0696 23.0	1037 22.7	2101 15.2	5005 24.5	9005 24.5	9005 24.5	5786 24.5	5786 24.5	9005 24.5	9005 24.5	5005 24.5	9005 24.5	9532 17.9	9676 23.9	9676 23.9	0660 13.0	2502 13.2	0469 24.0	1841 13.1	5166 23.4	3209 0.3	5166 23.4	4030 23.3	5166 23.4	9794 13.0	8592 17.6	0.0125 0.0093	0.0062									50248, MZ08
290 23.1	037 22.6	606 15.2	637 24.5	637 24.5	637 24.5	418 24.4	418 24.4	637 24.5	637 24.5	637 24.5	637 24.5	904 18.0	276 23.8	276 23.8	660 13.0	502 13.2	1073 23.9	223 13.1	735 23.4	158 0.3	735 23.4	687 23.2	735 23.4	206 13.0	969 17.7	8										0336 MZ06
005 18.2	805 11.0	259 18.1	049 17.3	049 17.3	049 17.3	829 17.3	829 17.3	049 17.3	049 17.3	049 17.3	049 17.3	637 0.1	642 16.7	642 16.7	775 18.4	701 18.1	444 16.7	437 11.1	066 16.3	209 17.9	066 16.3	891 16.5	056 16.3	019 15.0	407											0234, HQ98
588 16.19	474 17.51	984 23.75	322 12.53	104 12.53	104 12.53	542 12.50	542 12.50	323 12.52	323 12.52	323 12.52	323 12.52	751 14.90	570 13.63	870 13.68	290 0.16	904 0.16	129 13.63	909 18.12	774 13.36	100 11.76	774 13.36	517 13.02	994 13.33	至						-						3965, GUS70
08 12.697	26 23.372	13199 (3 MQ) 13.373 (3.373 (3.375 (3.375 (3.274 (5.274 (5.274 (5.274 (5.274 (5.274 (5.274 (3.275 (3.275 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274 (3.274))))))))))))))))))))))))))))))))))))	02678 02574 02567 245766 24505 245567 24594 77.322 12534 0.172 0.175 0.1066 22.453 0.1066 13.3567 0.1689 11713 12.448 0.1717 0.1717 18.2577 0.0046 0.0046 0.0046 0.0046 0.0046 0.0051 0.0051 0.0051	0270 0225 0278 24576 24605 24507 24504 77304 12594 0776 0799 0180 22453 0180 13374 0184 11773 12488 0772 0172 18229 0005 0005 0005 0005 0005	0222 02718 24.578 24.605 24.527 24.594 17.514 12.529 0.778 0.759 0.160 22.4153 0.160 13.374 0.1684 11.773 12.048 0.172 0.772 18.229 0.005 0.005 0.005 0.005 0.005	0254 0255 24558 24578 24578 24578 24574 2452 4754 0472 0472 0473 2336 0473 13374 0463 11836 12057 0468 0468 18274 0405 0405 0405 0405	0254 0255 24558 24578 24578 24578 24579 24594 0729 0772 0772 0773 2336 0773 13374 0769 11584 12057 07680 07680 18274 0015 0005	0257 24576 24605 24557 24594 77322 12524 0173 0172 01776 12453 01776 13374 01662 11709 12022 01689 01689 182511 0000 0000	02207 24506 24605 24507 24504 17322 12524 0175 0172 0178 2445 01776 13174 01652 11709 12022 01699 01689 18251 0000	02207 245768 24605 24557 24594 77.323 125204 0.1735 0.1726 0.1776 12.4453 0.1776 13.3174 0.1622 11.7039 12.0222 0.1639 0.1639 18.2511 0.0000	02507 245766 246005 245507 245949 17.3223 12.5204 0.1753 0.1778 0.1776 12.3413 0.1776 13.3174 0.1652 11.7009 12.0222 0.1639 0.1639 18.2511	55 17.456	0370 0309 23944 239576 239276 23862 167870 13888 01886 01897 01872 226891 0.1672 12487 00125 131926 131494	03770 03099 23944 239576 239276 239542 1637870 136689 0.1685 0.1687 0.1672 226881 0.1672 124897 0.0125 131926 13.145	69 12793	43 12.421	03268 03170 24.0239 24.0469 24.0073 23.9444 16.7429 13.6549 0.1686 0.1617 0.1642 22.7640 0.1642 12.4740	74 13.420	02224 23.4921 23.5165 23.4735 23.4056 16.3774 13.3888 0.0015 0.0031 0.0000 18.6055	76 18.605	0.2224 23.4921 23.5165 23.4735 23.4056 16.3774 13.3688 0.0015	57 0.0016	æ													702 JF45714
11 12423	25 23,403	22 13,444	2 0.175	76 0.175	76 0.175	29 0.170	29 0.170	53 0.17Z	S3 0.172	53 0.17Z	S3 0.172	99 17.615	36 0.184	36 0.184	33 12.358	16 12.092	0.181	00 13.164	15 0.003	55 18.439	15 0.0031	6														40. 545 71
5 12676	8 23.397	11 13205	5 0.180	0.180	0.180	2 0.175	0.175	6 0.177	6 0.177	36 0.177	6 0.177	0 17.437	0.187	0.187	0 12772	12402	17 0.184	12 13.420	000	18.605	<u></u>															34.1 5-45713
3 18 1847	3 22 6017	2 13.6583	6 23.4153	0 23.4153	0 23.4153	3 23.3906	3 23.3906	6 23.4153	6 23.4153	6 23.4153	6 23.4153	1 18.290	2 22.6691	2 22,6691	5 11.688	3 11.2301	2 22.7640	0 13.2903	0 18.6055	5																8.1545713
12.6763	7 23,3973	3 13.2052	0.1806	0.1800	0.1800	0.1753	0.1753	0.1778	3 0.1776	3 0.1776	3 0.1776	17.4371	0.1872	0.1872	127725	12,4023	0.1842	3 13.4200	0.																	315457133
22.6738	13.2326	22,4989	13.3367	13.3174	13.3174	13.3174	13.3174	13.3174	13.3174	13.3174	13.3174	10.4715	12,4097	12,4097	23.0686	227171	12,4740																			5457141
13.1484	23.4961	13.1008	0.1689	0.1684	0.1684	0.1639	0.1639	0.1662	0.1662	0.1662	0.1662	17.7524	0.0125	0.0125	13.5623	13.1314																				J457164
16.4899	18.2316	23.2822	11.7173	11.7173	11.7173	11.6845	11.0845	11.7009	11.7009	11.7009	11.7009	18.0674	13.1926	13.1926	0.0015																					J0350515.
15.9787	18.5365	23.6225	12,0408	12,0408	12,0408	12.0057	12,0057	12.0232	12.0232	12.0232	12.0232	18.3160	13.1464	13,1464																						JQ350512
13.1674	23.5152	13.1041	0.1717	0.1712	0.1712	0.1686	0.1686	0.1689	0.1689	0.1689	0.1689	17.7957	0.0000																							F457163.
13.1674 2	23.5152 1	13.1041 1	0.1717 1	0.1712 1	0.1712 1	0.1666 1	0.1686 1	0.1689 1	0.1689 1	0.1689 1	0.1689 1	17.7957																								Q801447: J
26341	0.7138 2	8.6108 1	8.2577	8.2299	8,2299	8.2724	8.2724	8.2511	8.2511	8.2511	8.2511																									F457162.1.U
26414 12	28306 22	3.3138 13	0.0046	0.0015 (0.0015 (0.0015 (0.0015 (0.0000	0.0000	0.0000																										599847. JO
16414 12	28306 22	13138 13	1.0046 0	1.0015 0	1.0015 0	1.0015 0	1.0015 0		0.0000																											599041: JOS
6414 12	8306 22	.3138 13.	.0046 0.	.0015 0.	.0015 0.	.0015 0.		0.000												_																99848. KUR
6414 126	806 228	3138 13.5	0046 0.0	0015 0.(0015 0.(0.0015		-		-									-																37944 J056
206 126	306 228	3138 13.5	0061 0.0	0031 0.0031	0031 0.0031	0.0000																														19822_059
206 12.6	006 228	138 13.3	1061 0.0	0031 0.0000	1031																															9020. J259
414 12.6	563 22.8	138 13.3	061 0.0061	8																																9037. JD999
114 12.61	58 Z28	138 13.33	16 1						-																											X05. J0599
170 13.17	522 11.96	8			-															-																1818. KU527
69 17.60	3																																			905. 103556
78																																				265. JOD435

Fig. 3. There is the genetic distance of the 36 different testing samples, and different samples are recorded by Genbank ID. The data are used to analyze the Inter-genera and intra-genera genetic distance.

because our network is very simple and performs well in classification. This will also be more convenient for finetuning. In contrast to previous studies, we used the original DNA sequence directly instead of the K-Mer method for encoding [18]. This met method is very effective for small

DNA sequences, such as the COI gene, which ensures the transmission of genetic information and reduces complicated pre-processing process.

As stated by Alexandrov et al. [19], the overall separation

of a set of mutational signatures can be evaluated by examining the distribution of cosine similarities between signatures. Similarly, to understand the model's genus prediction results, we also referenced the cosine similarity values in the model. The formula is shown in formula (7): A_i and B_i are the components of vectors, and n is the size of the vector. We used the average of the feature vectors from the same genera in the training set shown in Formula (8), where m is the number of training samples. It can be used to measure

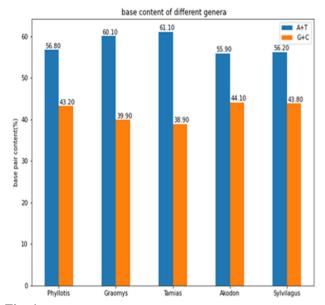


Fig. 4. Base content of the genera *Phyllotis, Graomys, Tamia, Akodon,* and *Sylvilagu* among misclassified samples in the test set. The blue column indicates the proportion of *Adenosine (A)-Thymine (T)* base pairs, and the orange column indicates the proportion of *Cytosine (C)-Guanine (G)* base pairs.

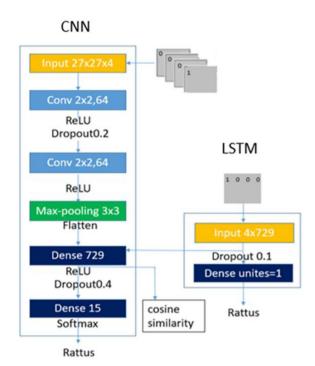


Fig. 5. CNN-LSTM hybrid model. The structure of CNN network: Input layer; the first Convolution layer: kernel size is (2,2), the number of kernels is 64, the activation function is 'ReLU', dropout coefficient is 0.2; the second Convolution layer: kernel size is (2,2), the number of kernels is 64, the activation function is 'ReLU'; Max-pooling layer: kernel size is (3,3); Flatten layer; the first Dense layer: the number of kernels is 729, the activation function is 'ReLU'; dropout coefficient is 0.4; the second Dense layer: the number of kernels is 15, the activation is 'softmax'. The structure of LSTM network: Input layer; the number of unites is 729, dropout coefficient is 0.1; Dense layer: the number of unites is 1.

the similarity between the test and training sets. This value can be used to measure the similarity between the test samples and training set. The closer the value is to 1, the greater the similarity between the two vectors. The values range from 0.733 to 0.999, as shown in Fig. 6. This indicates that the test sequence was highly similar to the training set.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$
(7)

$$\mathbf{A} = \frac{\sum_{j=1}^{m}}{m} \tag{8}$$

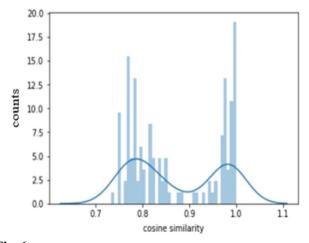


Fig. 6. Statistics of cosine-similarity value in the test set samples.

IV. DISCUSSION AND CONCLUSIONS

A. Results and Discussion

In contrast to previous studies that calculated the K-Mer frequency [20], the bases of the COI gene were converted to a vector matrix by one-hot encoding and could work directly on the sequence to make the model more direct and convenient. The feature extraction ability of the model can be improved with 94% accuracy by combining two features that are separate from the CNN and LSTM networks. When 17 *Calomys* samples or four genera, the model performed significantly with an accuracy of 100%. This implies that the proposed model is trainable and applicable. Compared with the K-means and SVM algorithm models, our hybrid model is more concise and efficient. There is no need to rebuild the model when expanding the amount of data, but only to fine-tune it to optimize the model.

We referred to the cosine similarity value to understand the results and initially assess the reliability of the predictions. The test set maintained a high degree of similarity with the training set, with values between 0.733 and 0.999. We calculated the cosine similarity value of the *Rattus* sample compared to that of the *Akodon* sample, which was only 0.559. Therefore, classification results with lower similarity values indicate that they are likely to be misclassified or that a new species exists.

B. Future Research

We constructed a CNN-LSTM hybrid model because the genetic information in the database is unbalanced in quantity, and there is a large deviation in the number of randomly selected samples. Although good classification results have been achieved, in future research, we need to develop the model using a larger dataset or for species classification. Mitochondria may not have a sufficient and stable mutation rate if the species formation time is very short or if mitochondrial gene outflow is present in closely related species. This makes it difficult to classify COI genes. If we set a different cosine similarity threshold for every category, it could help quantitatively evaluate the prediction results and improve the function of the model.

REFERENCES

- P. D. Hebert, A. Cywinska, S. L. Ball, and J. R. DeWaard, "Biological identifications through DNA barcodes," *in Proceedings* of the Royal Society of London. Series B: Biological Sciences, vol. 270, no. 1512, pp. 313-321, Feb. 2003. DOI: 10.1098/rspb.2002. 2218.
- [2] P. D. Hebert, S. Ratnasingham, and J. R. De Waard, "Barcoding animal life: cytochrome c oxidase subunit 1 divergences among closely related species," *in Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. suppl_1, pp. S96-S99, Aug. 2003. DOI: 10.1098/rsb1.2003.0025.
- [3] P. D. N. Hebert, M. Y. Stoeckle, T. S. Zemlak, and C. M. Francis, "Identification of birds through DNA barcodes," *PLoS Biology*, vol. 2, no. 10, p. e312, Sep. 2004. DOI: 10.1371/journal.pbio.0020312.
- [4] S. M. Guan, and B. Q. Gao, "COI sequence, the DNA barcode affecting animal taxonomy and ecology [J]," *Chinese Journal of Ecology*, vol. 27, no. 8, pp. 1406-1412, 2008. [Online] Available: http://www.cje.net.cn/CN/abstract/abstract15065.shtml.
- [5] Y. F. Tan, and R. C. Jin, "The efficient algorithm for reconstructing phylogenetic tree based on neibor-joining method," *Computer Engineering and Applications*, vol. 40, no. 21, pp. 84-85, 2004.
 [Online] Available: http://caod.oriprobe.com/articles/8475946/The_Efficient_Algorithm_for_Reconstructing_Phylogenetic_Tree_Based_on_Neibor_joining_Method.htm.
- [6] A. Tampuu, Z. Bzhalava, J. Dillner, and R. Vicente, "ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples," *PloS One*, vol. 14, no. 9, pp. e0222271, Sep. 2019. DOI: 10.1371/journal.pone.0222271.

- [7] U. Singh, S. Chauhan, A. Krishnamachari, and L. Vig, "Ensemble of deep long short term memory networks for labelling origin of replication sequences," *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1-7, Oct. 2015. DOI: 10.1109/DSAA.2015.7344871.
- [8] H. Gunasekaran, K. Ramalakshmi, A. Rex Macedo Arokiaraj, S. Deepa Kanmani, C. Venkatesan, and C. Suresh Gnana Dhas, "Analysis of DNA sequence classification using CNN and hybrid models," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1-12, Jul. 2021. DOI: 10.1155/2021/1835056.
- [9] GenBank nucleic acid sequence database in NCBI. [Online] Available: https://www.ncbi.nlm.nih.gov/nuccore/?term=.
- [10] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, no. 8, pp. 831-838, Aug. 2015. DOI: 10.1038/nbt.3300.
- [11] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome Research*, vol. 26, no. 7, pp. 990-999, 2016. DOI: 10.1101/gr.200535.115.
- [12] A. Tharwat, "Parameter investigation of support vector machine classifier with kernel functions," *Knowledge and Information Systems*, vol. 61, no. 3, pp. 1269-1302, Dec. 2019. DOI: 10.1007/ s10115-019-01335-4.
- [13] A. Ghosh, Anirudha, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," *Recent trends and advances in artificial intelligence and Internet of Things*, pp. 519-567, 2020. DOI: 10.1007/978-3-030-32644-9_36.
- [14] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235-1270, Jul. 2019. DOI: 10.1162/neco_a_ 01199.
- [15] J. Zhou, and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature Methods*, vol. 12, no. 10, pp. 931-934, Oct. 2015. DOI: 10.1038/nmeth.3547.
- [16] J. D. Watson, T. A. Baker, A. Gann, S. P. Bell, M. Levine, and R. M. Losick, Molecular Biology of the Gene, 7th ed. San Francisco: Pearson, 2013.
- [17] L. Deng, H. Wu, X. Liu, and H. Liu, "DeepD2V: a novel deep learning-based framework for predicting transcription factor binding sites from combined DNA sequence," *International Journal of Molecular Sciences*, vol. 22, no. 11, p. 5521, May 2021. DOI: 10.3390/ijms22115521.
- [18] Y. Zhang, S. Qiao, S. Ji, and Y. Li, "DeepSite: bidirectional LSTM and CNN models for predicting DNA-protein binding," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 4, pp. 841-851, Apr. 2020. DOI: 10.1007/s13042-019-00990-x.
- [19] L. B. Alexandrov, J. Kim, N. J. Haradhvala, M. N. Huang, A. W. Tian Ng, Y. Wu, A. Boot, K. R. Covington, D. A. Gordenin, E. N. Bergstrom, and S. A. Islam, "The repertoire of mutational signatures in human cancer," *Nature*, vol. 578, no. 7793, pp. 94-101, Feb. 2020. DOI: 10.1038/s41586-020-1943-3.
- [20] H. Vinje, K. H. Liland, T. Almøy, and L. Snipen, "Comparing K-mer based methods for improved classification of 16S sequences," *BMC Bioinformatics*, vol. 16, no. 1, pp. 1-13, Dec. 2015. DOI: 10.1186/ s12859-015-0647-4.



Meijing Li

is a M.S. student in the Department of Artificial Intelligence and Data Engineering at Sangmyung University. She received her B.S. degree in the Department of Biopharmaceutical from JILIN University in 2014. Her research areas include bioinformatics and data mining.



Dongkeun Kim

2003 Yonsei University Master's Degree in Medical Informatics 2008 Yonsei University Ph.D. in Biomedical Engineering 2009-present Professor at the Department of Human Intelligence and Information Engineering, Sangmyung University 2017-present Director of the Intelligent Information Technology Research Institute, Sangmyung University 2021-present Head of the Bio-Health Innovation Sharing University Unit, Sangmyung University Areas of interest: Bio-Health, Biomedical Engineering, Data Mining, and Demand Forecasting