COVID-19 vs influenza viruses: A cockroach optimized deep neural network classification approach

Mohamed A. El-dosuky1 | Mona Soliman2 | Aboul Ella Hassanien2

1Faculty of Computers and Info, Mansoura University, Mansoura, Egypt
2Faculty of Computer and Artificial intelligence, Cairo University, Cairo, Egypt

Correspondence
Aboul Ella Hassanien, Faculty of Computers and Information, Cairo University, Cairo, Egypt.
Email:

Abstract
Among Coronavirus, as with many other viruses, receptor interactions are an essential determinant of species specificity, virulence, and pathogenesis. The pathogenesis of the COVID-19 depends on the virus's ability to attach to and enter into a suitable human host cell. This paper presents a cockroach optimized deep neural network to detect COVID-19 and differentiate between COVID-19 and influenza types A, B, and C. The deep network architecture is inspired using a cockroach optimization algorithm to optimize the deep neural network hyper-parameters. COVID-19 sequences are obtained from repository 2019 Novel Coronavirus Resource, and influenza A, B, and C sub-dataset are obtained from other repositories. Five hundred ninety-four unique genomes sequences are used in the training and testing process with 99% overall accuracy for the classification model.

KEYWORDS
cockroach swarm optimization, convolutional neural networks, coronavirus, COVID-19, deep learning, influenza, SARS-CoV-2

1 | INTRODUCTION

At the end of the year 2019 in Wuhan in China, the first pneumonia cases of unknown etiology were identified. The virus causing this pneumonia is named SARS-CoV-2 because it has a phylogenetic affinity to Severe Acute Respiratory Syndrome coronavirus (SARS-CoV). The disease caused by SARS-CoV-2 is called by the World Health Organization (WHO) a corona-virus disease 2019 (COVID-19). Since then, the virus has spread exponentially in both rich and developing countries, leading to a pandemic health emergency. As the COVID-19 pandemic continues to spread, researchers attempt to grasp the virus's behavior better and curtail its growth and effect worldwide.

To control COVID-19, many suspected cases need to be tested so that they can be advised to remain socially distant and be treated correctly at the proper moment. There is a need for designing an efficient detection tool for COVID-19. Also, it is essential to distinguish between COVID-19 and other similar symptoms like influenza. The most popular research technique widely used to diagnose COVID-19 is a real-time reverse transcription-polymerase chain reaction (RT-PCR) with a sensitivity of 60% to 70%. Also, body radiological imaging, such as computed tomography (CT) and X-ray, has essential early diagnosis and therapy roles. Signs can be detected by examining patient radiological pictures. States that CT is a sensitive method for diagnosing COVID-19 pneumonia and may be used as an RT-PRC screening tool. Due to its reasonably recent appearance and not completely understood characteristics, there is still an unsatisfying detection rate for SARS-CoV-2.

In addition, SARS-CoV-2 is strikingly similar to other diseases like influenza and respiratory infection Coronavirus diseases, rendering detection much more
difficult. Therefore, enhancing current diagnostic methods for suppressing the spread is essential. More recent researches aim to deal with this problem using genome sequencing to detect SARS-CoV-2 and include a distinction between COVID-19 and other virus types through a classification process.

Classification processes are heavily dependent on the characteristics of the objects to be identified reflect. It is crucial to consider and quantify the essential aspects of the entity to create a reasonable representation. Still, in some situations, it is challenging to grasp the features use, and this affects the outputs of the classification model.\(^6\) Previously, neural profound learning systems or deep learning models have derived valuable functionality from input patterns. Deep learning technology is based on the artificial neural network system (ANNs). Such as ANNs are continually taking learning algorithms, and it is possible to increase the performance of the training processes by continuously growing the details. The accuracy depends on more significant volumes of data. The deep learning process entirely depends on two stages: the training phase and the inferring phase.\(^7\) Deep learning can be used to classify diseases from genome data.\(^8\) The genome data generally consists of deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins. We can extract viral RNA through blood samples of an infected human. Instead of looking at the signs, we will detect virus activity very early on.

The convolutional neural network (CNN) architecture design is heavily dependent on parametrization and frequently involves technical knowledge of intrinsic existence or several experiments. Experts decide which hyperparameters to change and how they can be tuned, so it may be challenging to find suitable settings for a non-expert. There is a need to remove constraints on deciding the number of coevolutionary layers and pooling layers and their shape, etc., from traditional architecture. To achieve this goal, the optimal configuration of the CNNs must be automatically created without any user interaction. State-of-the-art techniques for selecting hyperparameters range from the basic grid and random\(^9\) queries to more sophisticated techniques that combine solution space exploration, and exploitation nature-inspired meta-heuristic methods.\(^10\) Few research papers are proposed for using deep learning for COVID-19 detection and diagnosis using Viral Gene Sequences. COVIDier, a deep-learning solution, is introduced to predict coronavirus family name from the gene sequences.\(^11\) It can classify the different genomes of Alpha coronavirus, Beta coronavirus, MERS, SARS-CoV-1, SARS-CoV-2, and bronchitis-CoV based on the training 1925 genomes belonging to the three families of SARS retrieved from the NCBI Database. COVIDier core idea predicts, with accuracy 99%, the genome with statistical similarity from the training dataset to the given genome. Another deep learning-based solution is proposed to achieve 98.1% accuracy in identifying SARS-CoV-2 from gene sequences.\(^12\) Experiments on data from the Novel Coronavirus Resource (2019nCoVR). The proposed work in Reference 12 correctly classified SARS-CoV-2, distinguishing it from other coronavirus strains, such as MERS-CoV, HCoV-NL63, HCoV-OC43, HCoV-229E, HCoV-HKU1, and SARS-CoV, regardless of missing information and errors in sequencing (noise).

This paper aims to provide a multi-classifier approach between COVID-19 and other common respiratory viruses, such as influenza virus (Type A, Type B, and Type C). This work utilizes the use of CNN to be used in the multi-classification model. We tackle the automated hyper-parameter selection problem in CNNs using one of the meta-heuristic algorithms known as cockroach swarm optimization (CSO). The experiments were built using 594 unique sequences. COVID-19 samples are gained from the repository 2019 Novel Coronavirus Resource.\(^13\) Influenza A, B, and C sub-datasets are obtained from the repository.\(^14\) The proposed model proves its efficiency during the testing process by providing 99% as an overall accuracy for the classification process.

2 | PRELIMINARIES

This section provides a brief explanation of the basic framework of CSO and CNN architecture along with some of the fundamental concepts.

2.1 | Cockroach swarm optimization (CSO)

The cockroaches’ (CSO’s) original version simulates specific cockroach's basic biological habits, including chase-swarming, dispersing, aggressive behavior. It has survived more than 0.35 billion years, more than 0.3 billion years older than a dinosaur. The cockroach has poor eyesight and a great smell. Up until now, social living makes them exist. Entomologists are discovering that cockroaches’ culture is similar, distinct from many other social organisms such as ants and bees. Yet, they still have swarm intelligence. When a family member goes to look for food, other participants may want to obey.\(^15\) Originally, CSO is contributed by Zhaohui and Haiyan.\(^16\) CSO is described as follows.\(^17\)

2.1.1 | Chase-swarming behavior

In conducting this behavior, cockroaches with the strongest power carry the best local solution candidate (\(P_i\))
and travel to the optimal global level ($P_g$) within the scope of its vision. This assures that a cockroach suddenly becomes stronger by finding a best solution candidate.

$$X_{i,G+1} = \begin{cases} \ X_{i,G} + \text{step-rand1} \cdot (P_g - X_{i,G}), & P_g \neq X_{i,G} \\ \ X_{i,G} + \text{step-rand2} \cdot (P_g - X_{i,G}), & P_g = X_{i,G} \end{cases}$$

(1)

where $X_{i,G}$ is the cockroach position at $G$th generation, rand1 and rand2 are random numbers in [0,1] interval, and step is a constant.

$P_g$ is the global best position at the $G$th iteration. $P_i$ is the local best position, based on Equation (2):

$$P_i = \text{Optimal}(X_i, |X_i - X_j| \leq \text{visual})$$

(2)

where visual is the perception constant.

### 2.1.2 Dispersion behavior

This behavior is performed frequently to ensure diversity. It performs a random step of all cockroaches.

$$X_i = X_i + \text{random}(1,D)$$

(3)

where random(1, D) is the random position in a $D$-dimensional range.

### 2.1.3 Ruthless behavior

This behavior mimics the phenomenon of eating weaker cockroaches when food resources are limited. It is implemented as replacing random individuals by the best individual. For $N$ cockroaches,

$$X_{\text{rand}} = P_g$$

(4)

where rand is a random integer in [1, N] interval. $P_g$ is the global best position.

### 2.2 Convolutional neural network (CNN)

Convolutional neural network (eg, ConvNet or CNN) contemplates a commonly utilized supervised deep learning platform. The convolutional neural network is a process in deep learning, consisting of several layers. In physiology, the ConvNets is driven by the visual cortex. Deep learning allows for the feeding of raw data into learning methods without the use of feature selection or visualization of a task diagram. Deep learning algorithms focused on the concept of mastering an correct set of attributes can do far more to accomplish those features than using hand-coding. The central principle of ConvNets is to obtain local features from the input at the upper layer (usually an image) and integrate them into more complex modules at lower layers. Its multi-layered architectural nature is computationally inefficient, and training these networks on a massive database takes many days. Such deep networks are equipped using GPUs. Convolution networks are Neural Networks’ simplest mathematical learning techniques, such as back-propagation, regularization, decent gradient, etc. There were three fundamental principles in CNNs: common receptive fields, mutual weights, and pooling.

#### 2.2.1 Local receptive field

Convolution networks are Neural Networks’ simplest mathematical learning techniques, such as back-propagation, regularization, decent gradient, and so on. There were three fundamental principles in CNNs: common receptive fields, mutual weights, and pooling.

#### 2.2.2 Weight sharing

Weight sharing corresponds to all receptive layer fields using the same filter/weights. It identifies the same characteristics in the reference image from different positions. That is why the map between the input layer and the hidden layer named the feature map, and weights are called mutual weights and biases. Mutual biases shared weights, and partialities also characterize the Kernel or filter. In image recognition, to detect more than one feature, we need more than one feature map to compromise the coevolutionary layer from multiple feature maps.

#### 2.2.3 Subsampling (pooling)

Subsampling minimizes the spatial size obtained and thus restricts the model’s weights. There are other methods of sub-sampling, but max-pooling is the most common technique for sub-sampling.

The convolution-based first functional architecture was LeNet, which uses back-propagation for network testing. LeNet was designed to understand handwritten digits (MNIST), and the processing of vast volumes of handwritten checks was introduced in the United States.
Sadly, the solution gained no coverage because it did not grow much into bigger problems. Since 2012, Differentiated CCN architecture has been suggested with different architecture. Modern ConvNets models’ performance began to increase, and researchers started working on how to reduce the scale and complexity of the existing ConvNet architectures without infringing on precision.

### 2.3 Hyper-parameter selection for CNN

Adjusting the hyper-parameter manually is undoubtedly one of the essential stages in the process flow for machine learning. Hyper-parameter retrieval technique takes intense computing cost, so the reproduction of experiments is almost impossible.\(^2^0\) The challenge in determining the correct hyper-parameter combination lies in a dynamic relationship between the parameters. For each hyper-parameter, the traditional naïve search strategy sets an acceptable range and seeks different values for each. This is known as grid quest. This somewhat sidesteps the search space for each iteration. This is expected to jump between such cycles, though, which would undoubtedly yield high output values. To address the drawbacks of grid search, random search operates on proven continuous and non-discriminatory functions.\(^2^1\) Random search is more likely to find an equilibrium internationally relative to a matrix search. Therefore, the number of independents Variables will degrade model efficiency.

The selection of hyper-parameters can be viewed as an optimization problem where the aim is to find a value that minimizes the loss function \( \ell(T; M) \) for a \( M \) model on a training set \( T \), often under certain conditions. This \( M \) model is developed using \( T \) using a learning algorithm \( A \), which usually involves solving an optimization problem. The model can be parameterized by the \( \lambda \) hyper-parameters and given as \( M = A(T, \lambda) \).\(^2^2\) The aim of the hyper-parameter extraction is to find the \( \lambda^* \) parametrization that generates a desirable \( M^* \) model, thus minimizing \( \ell(V, M^*) \) where \( V \) is the validity range. It can be described in the following equation\(^2^0\):

\[
\lambda^* = \arg\min_\lambda \ell(T; M) = \arg\min_\lambda f(\lambda; A, T, V, I)
\] (5)

The \( f \) objective function takes the \( \lambda \) hyper-parameters and provides the appropriate loss factor. The \( T \) and \( V \) datasets (where \( T \cap V = \emptyset \) is given), and the \( A \) learning algorithm along with the \( I \) loss function are chosen in advance. The training model’s generalization potential is quantified using the test set.

### 3 The Proposed Viral Gene Sequences Classification Model

This paper proposes a multi-classification model using viral gene sequences of the coronavirus and CNN. This model is used to provide a multi-classification tool between COVID-19 and three types of Influenza viruses (A, B, and C). A hyper-parameter selection is performed for CNN using CSO. We demonstrate that CSO efficiently explores the solution space, allowing a minimal topology of CNN to obtain competitive classification performance over the genome dataset. In the following, we will provide more details about the used data and proposed CNN architecture.

Figure 1 explains in details the proposed classification model. For the classification process, the proposed model can classify between four classes. It can classify between COVID-19 and three types of influenza viruses.

#### 3.1 Genome sequencing data characteristics

Viruses are the most common biological species on Earth that significantly affect living beings by inducing disease that influences their immune responses. More than 0.01% of viruses are sequenced, given their ubiquity, and impact.\(^2^3\) Genome sequencing shows the nucleotide sequence in a genome, identical to alphabet letters in English. Nucleotides are organic molecules that provide the building block for the basic structure of nucleic acids, such as RNA or DNA.

Comparing nucleotide composition in one virus gene to nucleotide order in another virus gene may show differences between the two viruses.\(^2^4\) Genome sequencing is a mechanism that specifies the sequence or order of nucleotides (ie, A, C, G, and U) in any of the genes found in the virus’s genome. This acronym stands respectively for adenine, cytosine, guanine, and uracil. Total genome sequencing can show the approximately 13 500-letter sequence of all genes in the genome of the virus.

Both influenza viruses are comprised of single-stranded RNA and dual-stranded DNA. The influenza virus RNA genes are made up of nucleotide chains bound together and coded by letters A, C, G, and U.\(^2^5\) Influenza is the most dangerous virus that causes Acute Respiratory Infections (ARIs), the most commonly found in lung infections. The structure of the influenza virus is comparatively basic. This contains predominantly 8-segmented RNA and highly immunogenic surface proteins. Three forms are distinguishable—influenza A, B, and C, all of which belong to the genus Orthomyxovirus.
Influenza A is infecting humans, ducks, goats, dogs, horses, and more. Genetic recombination via a segmented genome is possible. The influenza-type B virus has comparable biological properties to the one of type A. They are nonetheless indistinguishable in size and shape through electron microscopy. The A-type was primarily blamed for 20th to 21st century pandemics. Influenza B infects mostly humans and occasionally other animals. The Antigenic drift happens less commonly than with A-type viruses. The influenza-type C virus infects humans typically but is observed less commonly, induces moderate pediatric infections, and occasionally affects adults. Via a shorter genome (one segment less) it varies from types A and B, and its main surface glycoprotein is hemagglutinin-esterase-fusion (HEF), acting as H and N, respectively.

Coronavirus is RNA virus, and the newly discovered SARS-CoV-2 virus has a single short RNA strand, around 27 to 32 letters in length. These letters can be read one at a time, using a decoding tool. Coronavirus belongs to the Coronaviridae family, including alpha, beta, delta, and gamma coronavirus. As the name suggests, when viewed under an electron microscope, the spherical outer spike protein shows a distinctive crown shape. It is understood that the virus infects several hosts, including humans, other primates, and birds.

For this work, the SARS-CoV-2 sub-dataset is obtained from the repository 2019 Novel Coronavirus Resource. Influenza A, B, and C sub-datasets are obtained from the repository 14 using filtration. Sequence Type is chosen to be GenBank, Nucleotide is chosen to be complete, and the host is chosen to be Homo sapiens (human). This results in 594 unique sequences. The naming of the classes is adopted from NCBI organism naming convention, as shown in Table 1. The width of any sample is 31,029 nucleotides. Shorter sequences are right-padded with N symbol. Then the dataset is encoded as follows: C = 0.25, T = 0.50, G = 0.75, A = 1, and missing entries = 0.

### Table 1

| Class label | Organism name | Number of samples |
|-------------|---------------|-------------------|
| 0           | SARS-CoV-2    | 65                |
| 1           | InfluenzaA    | 287               |
| 2           | InfluenzaB    | 235               |
| 3           | InfluenzaC    | 7                 |

#### 3.2 Cockroach optimized deep neural network architecture

CNN architecture mainly consists of several convolutions, pooling layers, and fully connected layers. Many interface parameters are essential to remember, such as network size, number of neurons, their form for each node, node connectivity, etc. Hence, state-of-the-art CNN architectures require profound expertise, indicating that many design parameters should be optimized to achieve the best results for a specific dataset. Therefore, the construction of suitable architectures for target dataset requires an expert or trial-and-error knowledge. Because of this scenario, adaptive programming approaches are particularly useful for CNN architectures.
This work utilizes a CSO algorithm to provide hyper-parameter selection, as shown in Algorithm 1.

**Algorithm 1**

Hyper parameter optimization using CSO

Result: Global optima

initializing parameters (step, visualScope, dimension, stoppingCriteria)

initialization of n cockroaches’ coordinates randomly

\(X_1, \ldots, X_n\)

while stoppingCriteria are not met do

  Search cockroaches for best local optimal within visualScope;
  Run hunt-swarming and update \(P_g\);

  if \(X_i\) is local optimum then

    \(X_i : = X_i + \text{step.randomNumber.}(P_g - X_i)\);

  else

    \(X_i : = X_i + \text{step.randomNumber.}(P_i - X_i)\);

  end

Run dispersion procedure and update \(P_g\);

Run ruthlessness procedure \((X_k = P_g \text{ or } X_k = 0)\);

end

Output Global optima;

The deep learning model used for the proposed model is a CNN with three convolutional layers and one fully-connected layer. The input is a vector of 31 029 elements, the maximum size of the genome sequences in the dataset. The optimizer used for the weights is Adaptive Moment Estimation (Adam), with learning rate \(lr = 10^{-5}\), and run for 1000 epochs. Table 2 summarizes the main structure of CNN used for the classification.

The classical CNN network structure was used for the classification process. The convolutional layer is mainly used for feature extraction. Pooling operations eliminate over-fitting by dimensional reduction of data. The convolution layer input was 31 029 dimensional and converted into a 198-dimensional feature vector using a full-connection network. Three convolutional layers are used, each followed by a pooling layer, a full-connection layer, and the output classification layer. Each convolutional layer has three parameters that are \(w_i, h_i,\) and \(wd_i\) to reflect the convolutional layer and pooling. Initially, we arrange the tensor as an image \((8 \times 594)\) 1 channel. This arranges the tensor into eight channels \((1 \times 594)\) 8 channels. Each three convolutional layer CN has input channels and output channels. The proposed CNN architecture using CSO as shown in Table 3 has the following layers: The Input layer with size \((1 \times 594)\) for 8 channels, Three convolutional layers \((i = 0,1,2)\) with the output to each layer \(w_i\) is the input to the second layer \(w_{i-1}\). The first convolutional layer input is eight channels from the input layer. Each layer \(i\) has \((1 \times wd_i)\) filter window size Convolution (input weights) moving 1 step each time with a relu. Finally, a max-pooling with a \(h_i\) width window size, moving \(h_i\) in width by step. These three convolutional layers are followed by the fully connected layer (FN) is \(w_3\), and finally, the Output layer with four classes for multi-classification.

4 | EXPERIMENTAL RESULTS AND DISCUSSIONS

This work demonstrates high accuracy with the proposed classification model using CNN with hyper-parameter selection using CSO. We indicate our proposed model’s performance using overall accuracy, precision, recall, F-score, and confusion metric. Up to our knowledge, no research investigates the classification between COVID-19 and different types of influenza, although similar symptoms, So no comparative analysis is introduced for the multi-classification model. This work is implemented using Python 3.7.1 on two devices. The first is Acer Extensa running Windows 8 pro, 32 bits.

**Table 3** Hyper-parameter selection for CNN using CSO

| Parameter Description | Value |
|-----------------------|-------|
| \(w_0\) | Output for first CN/input for second CN | 131 |
| \(w_1\) | Output for second CN/input for third CN | 205 |
| \(w_2\) | Output for third CN | 151 |
| \(h_0\) | Width window size for first CN | 147 |
| \(h_1\) | Width window size for second CN | 235 |
| \(h_2\) | Width window size for third CN | 80 |
| \(wd_0\) | Filter window size for first CN | 127 |
| \(wd_1\) | Filter window size for second CN | 107 |
| \(wd_2\) | Filter window size for third CN | 123 |
| \(w_3\) | Output of FC | 198 |
4.1 Parameter setting for CNN using CSO

The main parameters of the CSO algorithm are shown in Table 4. The step parameter is set to 2. A smaller value of step guarantees to effectively scan the solution space but takes considerable computational time. The larger significance of step makes cockroaches” jump.”. This may make the algorithm miss the optima. Visual − Scope affects finding the optima. The dispersion coefficient affects diversity. hunger − threshold and ruthless behavior mimics the phenomenon of eating weaker cockroaches when food resources are limited. It replaces random individuals with the best individual.

4.2 Performance evaluation metrics

The confusion matrix enables the output of an algorithm to be visualized. A method’s accuracy (ACC) determines how accurate the values expected are. Precision (PR) specifies whether the experiment is reproducible or how many assumptions are accurate. Sensitivity (SE) tests the proportion of desirable components that are detected correctly. Specificity (SP) tests the percentage of negatives considered correctly. The following calculations illustrate how to measure those values, where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative.

\[
\text{Accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}
\]

\[
\text{Precision (PR)} = \frac{TP}{TP + FP} \tag{7}
\]

\[
\text{Sensitivity (SE)} = \frac{TP}{TP + FN} \tag{8}
\]

| Parameter                  | Value      |
|----------------------------|------------|
| step                       | 2          |
| visualScope                | 5          |
| Population size (N)        | 100        |
| Dispersion coefficient     | 20         |
| Hunger−threshold           | 0.5        |
| Speed controller (c)       | 5          |
| Cockroach range            | [−50, 50]  |
| stoppingCriteria (maximum iteration) | 1000       |

4.3 Training process

As one of the most classical loss functions used in classification models, cross-entropy was used in this study. When the epoch number of training iterations increased to more than 500, the loss value did not decrease or increase, suggesting that the models converged well to a relative optimal state without distinct overfitting. The training curve of the loss value and the accuracy rate for the classification model are shown in Figure 2.

The accuracy per epochs curves over all cross validation experiments are shown in Figures 3 and 4. These learning curves are widely used in machine learning for algorithms that learn incrementally over time. As we see for all folds of our experiment, the classifier accuracy is incrementally increased. This is an indicator of the model ability to classify among different classes after a certain number of epochs.

4.4 Multi-classification model results

The Multi-classification model aims to classify between COVID-19 and three types of influenza (A, B, and C). Figure 5 gives the confusion matrix of the proposed model. More investigation about multi-classification results are shown in Table 5. Influenza type-C gives the best accuracy measures among the other three classes. From Table 5 the overall Accuracy (OA), Overall Sensitivity (OS), and Overall Specificity (OP) are estimated to be 99%.

This work’s main contribution is its success in using genome data in COVID-19 classification against similar symptom diseases like influenza viruses with very high accuracy measures. We can observe the supremacy of the proposed model in discriminating the COVID-19 class from other classes. The high accuracy measures and high discrimination capacity are achieved by selecting the best parameters for CNN training. Such selection is performed using an optimization algorithm (eg, CSO) with the main objective function of neural network loss minimization. The CSO successfully spans the solution space and produces reliable and high-quality outcomes across various experimental environments, easily surpassing human experience while refining an internal expert-designed CNN architecture. Overall accuracy for CNN without using any optimization is evaluated to be 98.5%. Using CSO as an optimization method for CNN hyper-parameter selection is used to raise this overall accuracy to 99%.
**FIGURE 2** Accuracy and loss rates for the proposed multi-classification model [Color figure can be viewed at wileyonlinelibrary.com]

**FIGURE 3** Classification accuracy per epochs for first class (Fold 0-5) [Color figure can be viewed at wileyonlinelibrary.com]
CONCLUSION AND FUTURE WORK

This paper proposes a cockroach optimized deep neural network identification and classification approach that effectively identify the COVID-19 from viral genome sequences and classifies it against three types of influenza viruses A, B, and C. Although weights are learned by training on the dataset, additional crucial parameters exist named hyper-parameters, which are not obtained explicitly from the training dataset. These hyper-parameters will take on various values and add difficulty in determining the optimum model and design. The proposed approach used viral gene sequences as the input samples. The particle population (each determined by the number of hyper-parameter values) is developed in search of hyper-parameters that produce CNN’s best classification results. To verify the proposed method’s utility, we carried out experiments using the dataset of COVID-19 and influenza A, B, and C. The new solution achieved more significant accuracy steps according to reported results. We used the suggested approach to aim for maximal CNN neural network hyper-parameters. An overall accuracy with values 99% is detected for the proposed model of the multi-classification.

In the future, we aim to use deep learning methods to identify COVID-19 in a more noisy environment. We believe that for noisy data samples, more complex CNN is required. CNN training and accuracy are mainly depending on the dataset used. For more complex data,
neural networks may need to go deeper to discover more features to classify correctly. Noisy and complex data may result from capturing the device itself. Standard training algorithms have a virtual space for development. Some of these popular ways are parameter initialization techniques, hyper-parameter selection, optimization methods, adaptive learning rates, and supervised pre-training.

**DATA AVAILABILITY STATEMENT**

Data openly available in a public repository that does not issue DOIs. The SARS-CoV-2 sub-dataset is obtained from the repository 2019 Novel Coronavirus Resource. Influenzas A, B, and C sub-datasets are obtained from the repository using filtration.

**ORCID**

Aboul Ella Hassanien https://orcid.org/0000-0002-9989-6681

**REFERENCES**

1. Guan WJ, Ni ZY, Hu Y, et al. China medical treatment expert Group for COVID-19. Clin Charac Coronavirus Dis. 2020;82:1708-1720. https://doi.org/10.1056/NEJMoa200232.

2. Zhu N, Zhang D, Wang W, China Novel Coronavirus Investigating and Research Team. A Novel Coronavirus from Patients with Pneumonia in China. N Engl J Med. 2020;382(8):727-733.

3. World Health Organization. WHO Director-General’s opening remarks at the media briefing on COVID-19, March 11, 2020.

4. Swati B, Jayashree R, Madhulika B. Predictive analytics to combat with COVID-19 using genome sequencing. SSRN Electron J. 2020. https://doi.org/10.2139/ssrn.3580692.

5. Ozturka M, Talob E, Yildirimc U, Baloglud O, Yildirime UA. Optimization of Machine Learning Algorithms. NIPS’12: 226-229. https://doi.org/10.1016/j.compbiomed.2020.103792.

6. Rizzo R, Fiannaca A, La Rosa M, Urso A. (2016) A deep learning approach to DNA sequence classification. In: Angelini C, Rancoita P, Rovetta S, eds. Computational Intelligence Methods for Bioinformatics and Biostatistics. CIBB 2015. Lecture Notes in Computer Science, Vol. 9874. Cham: Springer. https://doi.org/10.1007/978-3-319-44332-4_10.

7. Ongsulee P. Artificial Intelligence, Machine Learning and Deep Learning. Paper presented at: 2017 15th International Conference on ICT and Knowledge Engineering, Bangkok; 2017, pp. 1–6. https://doi.org/10.1109/ICTKE.2017.8259629.

8. Lyu B, Haque A. Deep learning based tumor type classification using gene expression data. Paper presented at: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, pp. 89–96; 2018.

9. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281-305.

10. Jasper S, Hugo L, Ryan A. In: Proc NIPS, Pereira F, Burges CJC, Bojou L, Weinberger KQ, eds. Practical Bayesian Optimization of Machine Learning Algorithms. NIPS12: Proceedings of the 25th International Conference on Neural Information Processing Systems - 2. USA: ACM; 2012:2951-2959.

11. Habib P, Alsamam AM, Saber-Ayad M, Hassan SE, Hamwieh A. COVIDier: a deep-learning tool for coronaviruses genome and virulence proteins classification. bioRxiv. 2020. https://doi.org/10.1101/2020.05.03.075549.

12. Rincon A, Tonda A, Mendoza-Maldonado L, Claassen E, Garssen J, Kranepool AD. Accurate identification of sars-cov-2 from viral genome sequences using deep learning. bioRxiv. 2020. https://doi.org/10.1101/2020.03.13.990242.

13. Beijing Institute of Genomics, Chinese Academy of Science, China National Center for Bioinformation & National Genomics Data Center. https://bigd.big.ac.cn/ncov/?lang=en (Accessed June 27, 2020).

14. https://www.ncbi.nlm.nih.gov/labs/virus/vssi/ (Accessed June 27, 2020).

15. Cheng L, Zh W, Yanhong S, Guo A. Cockroach Swarm optimization algorithm for TSP. Adv Eng Forum, ISSN: 2234-991X. 2011:1:226-229. https://doi.org/10.4028/www.scientific.net/AEF.1.226.

16. Obagbua IC, Adegumi AO. An improved cockroach swarm optimization. Sci World J. 2018;2014:1-13. https://doi.org/10.1155/2014/375358.

17. Cheng L, Song Y, Bian Y. Cockroach swarm optimization using a neighborhood-based strategy. Int Arab J Inf Technol. 2019;16(4):784-790.

18. Schmidhuber J. Deep learning in neural networks. Neural Netw. 2015;61:85-117.

19. Erivaldo F, Junior F, Yen G. Particle swarm optimization of deep neural networks architectures for image classification. Swarm Evol Comput J. 2019;49:62-74.

20. Lorenzo P, Nalepa J, Kawulok M. Particle swarm optimization for hyper-parameter selection in deep neural networks. Paper presented at: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 481–488. https://doi.org/10.1145/3071178.3071208; 2017.

21. Goodfellow I, Farley D, Mirza M, Courville A, Bengio Y. Proceedings of the 30th International Conference on Machine Learning, PMLR 2015;28(3):1319-1327.

22. Wu J, Chen XY, Zhang H, Xiong LD, Lei H, Deng S. Hyper-parameter optimization for machine learning models based on Bayesian optimization. J Electron Sci Technol. 2019;17(1):26-40.

23. Houldcroft C, Beale M, Breuer J. Clinical and biological insights from viral genome sequencing. Nat Rev Microbiol. 2017;15:183-192.

24. Korsun N, Angelova S, Viki Gregory V, Daniels R, Georgieva I, Kolocorska A. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med. 2020;121:103792. https://doi.org/10.1016/j.compbiomed.2020.103792.

25. Dziabowska K, Czaczyk E, Nidzworski D. Detection methods of human and animal influenza virus-current trends. Adv Eng Forum. 2011;1:226-229. https://doi.org/10.4028/www.scientific.net/AEF.1.226.

26. Song H, Qi J, Khedri Z, et al. An open receptor-binding cavity in SARS-CoV-2. bioRxiv. 2020;82:395. https://doi.org/10.1101/2020.05.03.075549.

27. Bouvier N, Palese P. The biology of influenza viruses. Vaccine. 2008;26:49-53. https://doi.org/10.1016/j.vaccine.2008.07.039.
28. Gavin PJ, Thomson RB. Review of rapid diagnostic tests for influenza. Clin Appl Immunol Rev. 2004;4:151-172.
29. Woo PC, Lau SK, Chu CM, et al. Characterization and complete genome sequence of a novel coronavirus, coronavirus HKU1, from patients with pneumonia. J Virol. 2005;79(2):884-895. https://doi.org/10.1128/JVI.79.2.884-895.2005.
30. Cui J, Li F, Shi ZL. Origin and evolution of pathogenic coronaviruses. Nat Rev Microbiol. 2019;17(3):181-192.
31. Kim JM, Chung YS, Jo HJ, et al. Identification of coronavirus isolated from a patient in Korea with COVID-19. Osong Pub Health Res Perspect. 2020;11(1):3-7. https://doi.org/10.24171/j.phrp.2020.11.1.02.
32. Mizrachi I. Genbank: the nucleotide sequence database, The NCBI Hand book [Internet], updated 22; 2007.
33. Ye F. Particle swarm optimization-based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data. PLoS One. 2017;12(12):e0188746. https://doi.org/10.1371/journal.pone.0188746.

How to cite this article: El-dosuky MA, Soliman M, Hassanien AE. COVID-19 vs influenza viruses: A cockroach optimized deep neural network classification approach. Int J Imaging Syst Technol. 2021;31:472–482. https://doi.org/10.1002/ima.22562