Original Research Article

Development of a COVID-19 Patients' Fatality Prediction System Using Swarm Intelligent Convolution Neural Network

ABSTRACT

Aims: This work aims to develop a system that can be used to accurately and timely predict the fatality of a positively tested COVID-19 patient through the use of a deep learning technique – a swarm intelligent convolutional neural network.
Methodology: The dataset used in this study was acquired from the Kaggle repository database. The dataset contains the Lung Chest X-Ray images of COVID-19 patients. The images were pre-processed to obtain the desired image quality for further processing. This was followed by segmenting the pre-processed images. An Enhanced Firefly Algorithm (EFA) was formulated by applying the roulette wheel selection procedure to model the movement process of the firefly as a deterministic process to assist the standard Firefly Algorithm (FA) and application of Chaotic Sinusoidal Map Function to model the attractive process of the firefly which establishes a balance between exploration and exploitation in FA. The EFA was applied to optimize Convolution Neural Network (CNN) hyperparameters (number of layers, number of filters per layer, filter size and batch size). The segmented result was subsequently presented to EFA-CNN feature extraction and prediction of COVID-19 patient fatality cases. The formulated deep learning models (EFA-CNN and CNN) were implemented using Matrix Laboratory 2020a software. The implemented models were evaluated using specificity, sensitivity, false positive rate, accuracy, and recognition time/rate to determine the performance of the developed models.
Results: The findings revealed that the EFA-CNN model performs better in the prediction of COVID-19 patients' fatality compared to the CNN model. It was also discovered that the formulated EFA applied to select optimal values of the hyper-parameters for the CNN architecture accounted for improved recognition accuracy and reduced recognition time of the developed COVID-19 Patients' Fatality Prediction System.
Conclusion: The system developed will assist both the government and healthcare workers in providing the needed computational capability for the prediction of the fatality level of a positively tested COVID-19 patient.

Keywords: COVID-19, Deep Learning, Feature Extraction, Fatality Level, Prediction

1. INTRODUCTION

The novel Coronavirus disease 2019 (COVID-19) is a virus of the Coronavirus family which has clinical characteristics similar to the SARS-CoV and is also a source of a respiratory ailment plague throughout the world that originated in Wuhan, China and has swiftly spread to almost every region of the world [1]. The disease is caused by a new and severe kind of Coronavirus known as Severe Acute Respiratory Syndrome Coronavirus 2 (SARSCoV-2) with its primary symptoms including fever and cough, while gastrointestinal symptoms are uncommon. In COVID-19-infected patients, the absence of fever is more incessant than in patients tainted by comparable infections, such as the MERS Corona

Virus and the SARS Corona Virus [2]; thus, there is a chance of non-febrile patients being missed by an observation instrument with an essential spotlight on distinguishing fever [3]

The Coronavirus (SARS-CoV-2 virus) has caused damaging effects on humanity since its beginning in late 2019. In the months since the virus has advanced to become a prevalent global pandemic causing significant morbidity and mortality [4]. As stated in [5], this pandemic has been declared a worldwide well-being crisis and is spreading at an alarming rate. As of July 5th, 2020, over two hundred countries have been afflicted by the virus, amounting to almost a total of 530,000 deaths worldwide and have accounted for over 4.7 million confirmed cases internationally [6,7,8].

This pandemic continues to inflict severe public health and socio-economic burden in many parts of the world, including Nigeria. Not just has this infection seriously influenced people who have gotten the disease, but it has in addition affected medical care representatives and even patients with ailments disconnected to COVID-19. Currently, there is a safe and effective vaccine or antiviral for use against the pandemic in humans, as stated in [9], the control and mitigation efforts against the pandemic are focused on implementing non-pharmaceutical interventions (NPIs), such as social (physical)-distancing, community lockdown, contact tracing, quarantine of suspected cases, isolation of confirmed cases and the use of face masks in public.

Owing to the severity that some COVID-19 cases progress to, hospitalization is required, and these cases may advance to ICU admission. This inflicts huge stress on healthcare workforces as hospitals are working at full capacity and at times lack adequate equipment. The incidence of hospitals frequently reaching full capacity has become a devastating and disturbing issue, this results in extensive physician exhaustion [10] which can be detrimental to physician-patient interactions. With the pressure on medical facilities, it is indispensable for governments and the healthcare sector to detect and treat cases that are most likely to survive, by so doing, astutely utilizing the limited stock of medical resources and medications.

Many studies have investigated various Artificial Intelligence (AI) – based models (machine learning models with deep learning techniques) for understanding the transmission dynamics, control, prediction, and classification of potential treatments of COVID-19 infection [9,11,12]. Deep Learning (DL) or more commonly known as deep structured learning or hierarchical learning is a division of Machine Learning (ML) which is based on a set of algorithms that attempt to model high-level abstractions in data [13,14]. Such algorithms develop a layered, hierarchical architecture of learning and representing data. This hierarchical learning architecture is inspired by artificial intelligence emulating the deep, layered learning process of the primary sensorial areas of the neocortex in the human brain, which automatically extracts features and abstractions from underlying data [15,16,17]. Based on [18,19]), DL algorithms are useful when it comes to dealing with large amounts of unsupervised data and naturally learn data representations in a greedy layer-wise method.

Hussain et al. [20] developed an AI imaging analysis tool to classify COVID-19 lung infection in patients based on portable CXRs from other conditions, using five supervised ML AI algorithms and multi-class datasets. Texture and morphological Features extracted. Two-class and multi-class classifications were performed. Statistical analysis was carried out using unpaired two-tailed t-tests with unequal variance between groups. The receiver-operating characteristic (ROC) curve analysis was used to evaluate the performance of the developed classification models. It was established that AI classification of texture and morphological features of portable CXRs accurately distinguishes COVID-19 lung infection

in patients in multi-class datasets, and Deep-learning methods have the potential to improve diagnostic efficiency and accuracy for portable CXRs.

Khalifa et al. [11] developed a classification model based on the DL model (Deep Convolutional Neural Network (DCNN)) and classical ML algorithms (support vector machine, decision tree, and ensemble). The model was used to classify potential coronavirus treatments on a single human cell, using the subset of the dataset obtained from RxRx.ai. The dataset was grouped into the treatment type and the treatment concentration. The numerical features from the original dataset were transformed into the image domain and then fed into a DCNN model. The DCNN model consisted of three convolutional layers, three ReLU layers, three pooling layers, and two fully connected layers. It was found that the DCNN achieved a high-quality result when compared with ML algorithms in terms of testing accuracy for the treatment classification, while the classical ML algorithms (decision tree and support vector machine) achieved a similar result with the DCNN for the treatment concentration level prediction.

Ardabili et al. [21] presented a comparative analysis of ML and soft computing models (in terms of generalization ability and accuracy) in predicting the COVID-19 outbreak as an alternative to epidemiological models. The data used was collected for over 30 days in five countries (Italy, Germany, Iran, USA, and China) from the Worldometers website. Equations (Logistic, linear, logarithmic, quadratic, cubic, compound, power, and exponential) were employed to develop the desired model for the estimation of the time-series data. To choose the best model for the comparative analysis, Parameter tuning of these models was performed using Evolutionary Algorithms (Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO). Multi-Layered Perceptron (MLP) and Adaptive Network-based Fuzzy Inference System (ANFIS) were used for the prediction of the outbreak in the five countries. The developed models were evaluated in terms of performance and accuracy, using the Root Mean Square Error (RMSE) and Correlation Coefficient. It was observed that the ML models are effective in modelling the time series of the outbreak and that high-quality outbreak prediction can be obtained by integrating ML and SEIR models since ML models are useful in handling the shortcomings associated with SEIR models for COVID-19 prediction.

In the work of Narin et al. [22] Convolutional Neural Networks (CNNs) that predicted novel coronavirus with X-ray images were developed. The DL technique was exploited for the automatic prediction of 2019-nCoV patients. The dataset (chest X-ray images) and pre-trained models consisting of InceptionV3, ResNet50 and Inception ResNetV2 were trained and tested on the dataset. The evaluation of the CNNs models indicated that the RestNet50 pre-trained model gave the highest accuracy (98%) among the three models and that the model can aid health employees to make high-accuracy decisions in clinical practice and also detecting 2019-nCoV in the early stages of infection.

In the work of Yang et al. [23] a modified susceptible-exposed-infectious-removed (SEIR) Model and ML Model for the prediction of the trend of the 2019-nCoV pandemic in China were evolved. Population migration data before and after 23rd January 2020 and updated 2019-nCoV epidemiological data were integrated into the SEIR Model to derive the pandemic curve. The ML approach was trained on 2003 SARS data to predict the pandemic. It was observed that the developed models were effective in predicting the pandemic peaks and size

In the work of Ayyoubzadeh et al. [24] data mining and a deep learning model were used to predict 2019-nCoV incidence through levering Google trend data in Iran. Long Short-Term Memory and Linear Regression Models were used to guess the number of 2019-

nCoV positive cases. The evaluation of the models was done with Root Mean Square Error (RMSE) metric and 10 folds cross-validation techniques, respectively. The RMSE of long short-term memory and linear regression models were 27.187 and 7.562, respectively. It was observed that both data mining and a deep learning model predicted the trend of the 2019-nCoV outbreak. Such predictions can support healthcare managers and policymakers with planning, allocating and deploying healthcare resources effectively.

Muhammad et al. [12] developed Supervised ML models for COVID-19 infection detection with some learning algorithms (logistic regression, decision tree, support vector machine, naive Bayes, and artificial neural network) using epidemiology labelled dataset for positive and negative COVID-19 cases of Mexico. The correlation coefficient analysis between various dependent and independent features was carried out to determine a strong relationship between each dependent feature and independent feature of the dataset before the development of the models. 80% of the training dataset was used for training the models while the remaining 20% was used for testing the models. The efficiency and quality of the models were evaluated using accuracy, sensitivity and specificity performance metrics. The results indicated that the Decision Tree model has the highest accuracy of 94.99% while the Support Vector Machine model has the highest sensitivity of 93.34% and the Naïve Bayes model has the highest specificity of 94.30%. It was observed that the supervised ML models can be used as retrospective evaluation techniques or tools to validate COVID-19 infection cases

Dianbo et al.[25] developed a machine learning-based approach which combines disease estimates from mechanistic models with digital traces, through interpretable machine-learning methodologies, to consistently forecast COVID-19 activity in Chinese provinces in real-time. The machine-learning approach used a clustering method that allows the exploitation of geospatial synchronicities of COVID-19 activity across Chinese provinces, and a data augmentation method to deal with the small number of historical disease activity observations, characteristic of emerging outbreaks. The approach used as inputs (a) official health reports from the Chinese Center for Disease for Control and Prevention (China CDC), (b) COVID-19-related internet search activity from Baidu, (c) news media activity reported by Media Cloud, and (d) daily forecasts of COVID-19 activity from GLEAM, an agent-based mechanistic model. It was established that the model produced steady and accurate forecasts 2 days in advance of the current time, and the model's predictive power outperformed a collection of baseline models in 27 out of the 32 Chinese provinces and could be easily extended to other geographies presently affected by the COVID-19 outbreak to help decision makers.

In the work of Edison et al. [26], the existing coronavirus vaccine development status was surveyed, and the Vaxign and Vaxign-ML RV approaches were used to predict COVID-19 protein candidates for vaccine development. Six possible adhesins were identified, including the structural S protein and five other non-structural proteins, and three of them (S, nsp3, and nsp8 proteins) were predicted to induce high protective immunity. The S protein was predicted to have the highest protective antigenicity score, and it has been extensively studied as the target of coronavirus vaccines by other researchers. The sequence conservation and immunogenicity of the multi-domain nsp3 protein, which was predicted to have the second-highest protective antigenicity score yet, was further analyzed in this study. It was observed that predicted vaccine targets have the potential for effective and safe COVID-19 vaccine development. The authors proposed that a "Structural Protein(s) (SP) / a Non-Structural Protein(s) (NSP) cocktail vaccine" containing an SP and an NSP would stimulate effective complementary immune responses.

Malik [27] presented a data-driven ML approach for the analysis of the COVID-19 pandemic from its early infection transmission dynamics especially the inflation counter over time, using US data starting from the first case on 21st January 2020. The actionable public health insight was extracted which includes infectious force, rate of the mild infection becoming serious, estimates for asymptomatic infections and prediction of new infections over time. The approach revealed a very significant number of cases of asymptomatic infections during the COVID-19 pandemic, a lag of about ten days. It was quantitatively confirmed that the infectious force of the virus is strong, with about 0.14% transition from mild to serious infection. It was observed that the approach was e□cient, robust and general, being agnostic to the specific virus and applicable to di□erent populations or cohorts.

A review of the above-related works with literature analysis revealed that although ML models and other DL techniques have played significant roles in the prediction, diagnosis and containment of the COVID-19 pandemic, however, an effective method to discover and treat COVID-19 cases that are most likely to survive depends possibly on quality classification and accurate prediction. The question is how quality is the classification and how accurate is the prediction associated with these deep learning techniques. Fregoso et al. [28] indicated that optimization of the hyper-parameters for the CNN architecture can improve the recognition accuracy and reduce the recognition time of the CNN model.

Therefore, this research focused on the use of deep learning techniques to develop a COVID-19 Patients' Fatality Prediction System Using a Swarm Intelligent Convolution Neural Network. As stated by Odeniyi et al [29], the advantage of using a machine learning model is mainly based on the accuracy of predictions, which in turn helps to redirect resources more accurately. Also, the deep Learning technique's important characteristics are its ability to handle relatively large amounts of unlabeled data efficiently [30], the ability to eliminate the need to extract essential features manually, and make a better analysis.

The approach used was based on an ensemble model that makes use of an Enhanced Firefly Algorithm-based Convolutional Neural Network (EFA-CNN). An Enhanced Firefly Algorithm (EFA) was formulated from standard FA and used to select optimal values for the hyper-parameters for the CNN architecture for feature extraction and prediction of COVID-19 patient fatality cases. The formulated deep learning models (EFA-CNN and CNN) were implemented using Matrix Laboratory 2020a software. The implemented models were evaluated using accuracy, specificity, sensitivity, false positive rate, and recognition time/rate and comparative analyses were carried out to ascertain the best model between (EFA-CNN and CNN) to predict COVID-19 patient fatality cases.

2. METHODOLOGY

In this work, an ensemble model that makes use of an Enhanced Firefly Algorithm-based Convolutional Neural Network (EFA-CNN) was used to implement the COVID-19 patients' fatality prediction system. As shown in Figure 1, the lung chest X-ray images of positive and negative COVID-19 patients were first acquired from the Kaggle repository database. The positive COVID-19 cases (lung chest X-ray images) were later categorized into severe, mild and moderate via the average value of individual features extracted. Then, the lung chest X-ray images were pre-processed to obtain the desired image quality for further processing. This was followed by segmenting the pre-processed images. An Enhanced Firefly Algorithm (EFA) was formulated by applying a roulette wheel selection procedure to model the attraction process as a deterministic process to assist the standard FA and application of the Chaotic Sinusoidal Map Function to establish a balance between

exploration and exploitation in standard FA. The EFA was applied to optimize CNN hyperparameters (number of layers, number of filters per layer, filter size and batch size). The segmented result was subsequently presented to EFA-CNN feature extraction and prediction of COVID-19 patients' fatality cases. The formulated deep learning models (EFA-CNN and CNN) were implemented using Matrix Laboratory 2020a software in Windows 10 Professional 64-bit operating system environment deployed on Hewlett-Packard G56 Branded computer system (Laptop), Intel® Core ™ i5 Duo with 2.7GHz speed, 6GB Random Access Memory (RAM) and 1 TB hard disk drive. The implemented models were evaluated using specificity, sensitivity, false positive rate, accuracy, and recognition time/rate to determine the performance of the developed model.

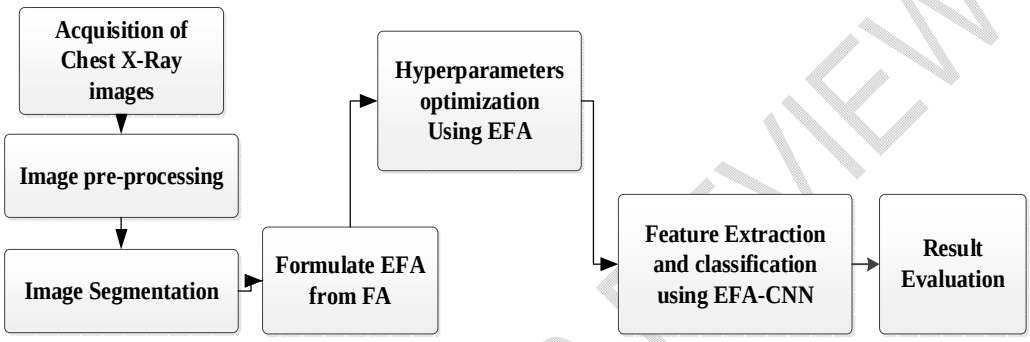


Figure 1: **Methodology to build deep learning models for Covid-19 Patients' Fatality Prediction system**

2.1 Image Acquisition

The dataset used in this study was acquired from the Kaggle repository database. The dataset contains the Lung Chest X-Ray images of positive and negative COVID-19 fatality patients. 3550 lung Chest X-Ray image datasets were used in the study; 2130 datasets were used for training while 1420 datasets were used for testing. The test data comprises 709 positive datasets and 711 negative datasets.

2.2 Image Pre-processing

The acquired images of COVID-19 were pre-processed. The Pre-processing techniques applied were image resizing, data augmentation and image normalization. The Lung Chest X-Ray images were resized to 224 X 224 pixels to make them suitable for the VGG-16 model and were subjected to data augmentation to increase training data and to reduce over-fitting problems. The augmentation techniques applied include rotation, vertical flipping, and horizontal flipping. The Lung Chest X-Ray images were enhanced through image normalization.

The histogram equalization method was applied for the Lung Chest X-Ray image normalization in which the intensity values of the images were distributed using the cumulative distribution function. This function finds out the transformation that maps input image grayscale values to transform image grayscale values. Histogram equalization was used for enhancement contrast that ensures that the input pixel intensity, x is transformed to a new intensity value x′ by T as shown in Equation (1). The transformed function (T) is

the product of a cumulative histogram and a scale factor. The scale factor was needed to fit the new intensity value within the range of the intensity values [31].

$$x' = T(x) = \sum_{i=0}^{N} n_i \left(\frac{\max\ (i)}{N}\right) \quad (1)$$

where n_i is the number of pixels at the intensity I, N is the total number of pixels in the image and max (i) is the maximum intensity i.

2.3 Image Segmentation

Segmentation of the Lung Chest X-Ray images was achieved using the Sobel-edge detection algorithm. Edge detection is more popular for identifying discontinuities in grey level than detecting isolated points and thin lines. The edge is the boundary between two regions with relatively distinct grey-level properties. The transitions between the two regions were determined based on the grey-level discontinuities. The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. In the input grayscale image, the approximate gradient magnitude was also identified at each point by the edge detector. The operator consists of a pair of 3x3 convolution kernels which are rotated by 90 degrees. The convolution masks of the Sobel detector are given in Equation (2) and Equation (3) [32].

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3)$$

Every point in the image used these two kernels G_x and $G_y$to do convolution. One of the two kernels has a maximum response to the vertical edge and the other has a maximum response to the level edge. The maximum value of the two convolutions was used as the output bit of the point, and the result was an image of edge amplitude.

The procedure employed to achieve Sobel Edge Detection is as follows [32]

Input: A Sample Image.
Output: Detected Edges.
Step 1: Accept the input image.
Step 2: Apply mask Gx, Gy to the input image.
Step 3: Apply the Sobel edge detection algorithm and the gradient.
Step 4: Masks manipulation of Gx, and Gy separately on the input image.
Step 5: Results combined to find the absolute magnitude of the gradient.
Step 6: The absolute magnitude is the output edges.

2.4 Formulation of Enhanced Firefly Algorithm

In the standard Firefly Algorithm (FA), the procedure starts from an initial population of randomly generated individuals. The quality of each individual is calculated using Equation

(4) and the best solution among them is selected. In FA, the form of attractiveness function of a firefly is depicted by Equation (4):

$$\beta(r) = \beta_0 \exp(-\gamma r^2) \quad (4)$$

where,

r = The distance between any two fireflies

β_0 = The initial attractiveness at $r = 0$

γ = An absorption coefficient which controls the decrease of the light intensity

The distance that exists in-between any two fireflies I and *j,* at a particular position x_i and *xj,* can be defined respectively as a Cartesian or Euclidean distance as shown in Equation (5):

$$r_{ij} = ||x_i - x_i|| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \quad (5)$$

where,

d is the dimensionality of the given problem.

The pattern of movement of a particular firefly I that is attracted by another brighter firefly *j* can be represented by the following Equation (6) and Equation (7):

$$x_i = x_i + \beta_0 \exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha * (rand - \frac{1}{2}) \quad (6)$$

$$x_i = x_i + \alpha * (rand - \frac{1}{2}) \quad (7)$$

In equation (6), the term x_i which is the first term is the present position of a firefly, and the term $\beta_0 \exp(-\gamma r_{ij}^2) * (x_j - x_i)$ which is the second term is meant for movement of the firefly towards the most attractive of the firefly by the intensity of light and the third term is meant to cater for the random movement of a firefly (random part) when it lacks the brighter ones. The α coefficient is a parameter for randomization, its value depends on the problem that is to be solved, while 'rand' is consistently distributed in the space (0, 1) as it is a random number generator. In equation 7, the movement of the best candidate is done randomly.

The Enhanced Firefly Algorithm (EFA) was formulated using Equation (8) to model the movement process of the firefly as a deterministic process instead of the random process in the existing firefly. The pattern of movement of a particular firefly I that is attracted by another brighter firefly *j* was enhanced by roulette wheel selection ($\boldsymbol{p_i}$) is expressed in Equation (8).

$$\boldsymbol{p_i} = \boldsymbol{rand} \leq \frac{f(x_i^t)}{\sum_{i=1}^{N} f(x_i^t)} \quad (8)$$

Where $\boldsymbol{f(x_i^t)}$ **is the** objective function value of firefly.

The enhanced pattern of movement of the firefly is shown in Equation (9) and Equation (10)

$$x_i = x_i + \beta_0 \exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha * (\boldsymbol{p_i} - \frac{1}{2}) \quad (9)$$

$$x_i = x_i + \alpha * (\boldsymbol{p_i} - \frac{1}{2}) \quad (10)$$

In equation (9), the term x_i which is the first term is the present position of a firefly, and the term $\beta_0 \exp(-\gamma r_{ij}^2) * (x_j - x_i)$ which is the second term is meant for movement of the firefly towards the most attractive of the firefly by the intensity of light and the third term is meant

to cater for the random movement of a firefly (random part), when it lacks the brighter ones. The α coefficient is a parameter for randomization, its value depends on the problem that is to be solved, while '$\boldsymbol{p_i}$' is consistently distributed using roulette wheel selection. In Equation (10), the movement of the best candidate is done randomly.

The challenges of imbalance between exploration and exploitation experienced by the standard firefly algorithm were resolved in this study by enhancing the attractiveness of fireflies with the application of chaotic theory and sinusoidal mapping. This described chaotic absorption coefficient (CA) which controls the decrease of the light intensity, but not constricting these fireflies to search space boundaries but counts on the nature of the chaotic system that generates random and unpredictable outputs from preceding conditions. The new attractiveness of the firefly was expressed in Equation (11), Equation (12) and Equation (13).

$$CA_{old} = \frac{mod(abs(\beta,\beta_0)}{\beta_0} \quad (11)$$

$$CA_{new} = \alpha * CA_{old}{}^2 \sin(\pi CA_{old}) \quad (12)$$

$$c\beta(r) = CA_{new} \times sign(\beta_0 \exp(-\gamma r^2))\beta(r) \quad (13)$$

In which
β_0 = is the initial attractiveness at r = 0,
r = is the distance between any two fireflies,
γ = is an absorption coefficient which controls the decrease of the light intensity,
$\beta(r)$ = was the existing light intensity update,
CA_{new} = was the chaotic sinusoidal mapping, where α = 2.3 as chaotic map parameters.
CA_{old} = was calculated to transform $\beta(r)$.
$c\beta(r)$ = is the enhanced updated light intensity of the firefly.

2.5 Development of COVID-19 Patients' Fatality System using EFA-CNN

The main feature of this work is to create a model for the classification of chest X-ray images using deep learning convolutional neural network. The classified lung chest X-ray images (severe, mild or moderate) were given as the input to this model so that the output could be the exact classified image. CNN was fine-tuned by using the EFA algorithm. By using this optimization approach, CNN was retrained with lung chest X-ray images to achieve the exact classification output. This developed approach was an efficient way of improving the CNN network's efficiency by pre-trained CNN networks, that is, VGG-19. To achieve the best performance of the developed approach, the hyper-parameters such as the number of layers, number of filters per layer, filter size and batch size of the CNN were optimized using the EFA algorithm.

After the convolutional and pooling layers, fully connected layers were located to merge the features obtained and in the last, the SoftMax layer, the output was computed. The strategy was to combine fully connected layers blocks by studying a nonlinear combination of the extracted features and also to execute the resultant classification.

.

2.5.1 The Optimization of Hyper-parameters

The pre-trained CNN architectures have several limitations. The noted limitations are that most of the hyper-parameters of any such pre-trained CNN cannot be modified and has some of the hyper-parameters which require adjustment namely, the batch size and also the unit numbers in every dense layer and the dropout layer. In this research, the EFA algorithm was employed in the CNN architecture model classifier section to optimize the batch size and dropout layer rate.

The dynamic parameters optimized through EFA were the number of convolutional layers, the size of the filters used in each convolutional layer, the number of convolutional filters, and the batch size. The overall methodology of the developed model is presented in Figure 2 which expressed the flowchart of the optimization process of the CNN by the EFA algorithm. The "training and optimization" block is the most important part of the whole process, where the CNN was initialized to integrate the parameter optimization by applying the EFA algorithm. In this process, the EFA was initialized according to the parameter given for the execution and this generated the fireflies. Each firefly is a possible solution and its position has the parameter to be optimized, so each solution represents a complete CNN training.

The training process is an iterative cycle that ends when all the fireflies generated by the EFA are evaluated for each generation. The computational cost is higher and, it depends on the database size, the size of fireflies, the number of iterations of the EFA and, the number of fireflies in each iteration. For instance, if the EFA is executed with 10 fireflies and 10 iterations, the CNN training process would be executed 100 times. The steps to optimize the CNN by the EFA algorithm are illustrated in Figure 2 and explained as follows:

i. Input the COVID-19 database to train the CNN. This step consists of selecting the COVID-19 database to be processed and classified for the CNN (COVID-19 severe, mild and moderate patients).
ii. Generate the firefly population for the EFA algorithm. The EFA parameters are set to include the number of iterations and the number of fireflies. This step involves the design of the fireflies.
iii. Initialize the CNN architecture, with the parameter obtained by the EFA (convolution layers number, the filter size, the number of convolution filters, and the batch size). The CNN parameter is initialized and in conjunction with the additional parameter specified, the CNN is ready to train the input COVID-19 database.
iv. CNN training and validation: The CNN reads and processes the input COVID-19 database taking the images for training, validation, and testing; this step produces a recognition rate. These values return to the EFA as part of the objective function.
v. Evaluate the objective function: The EFA algorithm evaluates the objective function to determine the best value.
vi. Update EFA parameters: At each iteration, each firefly updates its light intensity depending on its own best-known movement and attractiveness in the search space and the global best-known movement and attractiveness of the firefly.
vii. The process is repeated, evaluating all the fireflies until the stop criteria are found (in this case, it is the number of iterations).
viii. Finally, the optimal CNN parameters were selected. In this process, the firefly represented by the global best-known movement and attractiveness is the optimal one for the CNN model.

2.5.2 Learning Phase

In the learning phase, the CNN architecture models were used to classify the Chest X-Ray images as depicted in Figure 3. Hence, feature extraction and fine-tuning were employed to adjust the VGG-19 network model to the current database used for this work. The Convolutional layer was stuck within the feature extraction process, while the classifier segment was swapped by the corresponding one. There were several layers in the current classifier: the fully connected layers consist of a dropout layer, flatten layer, a batch normalization layer, and two dense layers. The first fully connected layer consists of neuron groups with a rectified linear unit and the second fully connected layer consists of four function units of SoftMax. After training the classifier for the number of iterations, the fine-tuning was achieved by reactivating the Convolutional last two layers and retraining with the classifier. Once the training process was completed, all these were merged to create the final prediction of Covid-19 using Chest X-Ray images which average their posteriors of SoftMax class

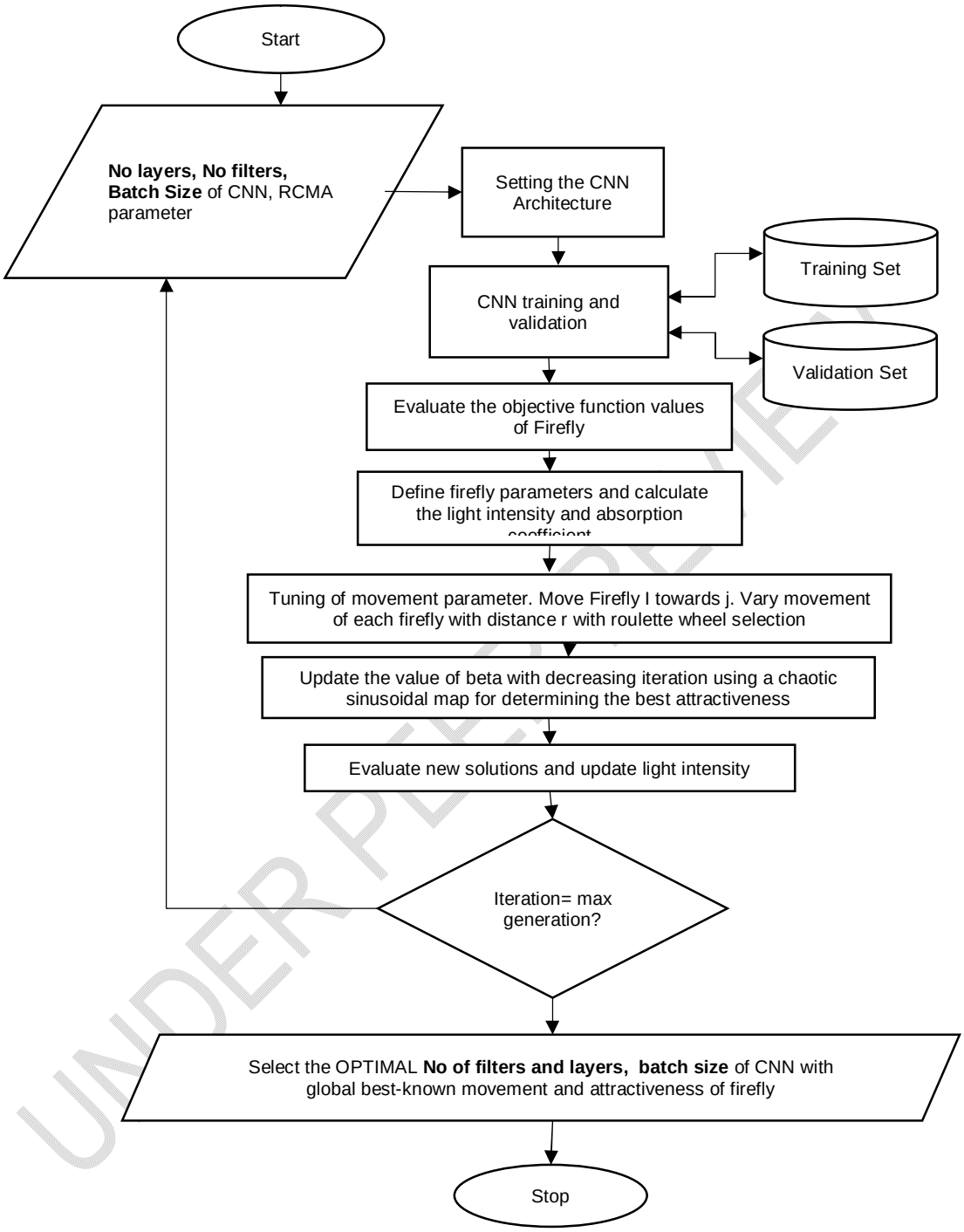


Figure 2: Flowchart of Enhanced Firefly Algorithm-based Convolutional Neural Network model (EFA-CNN) classifier

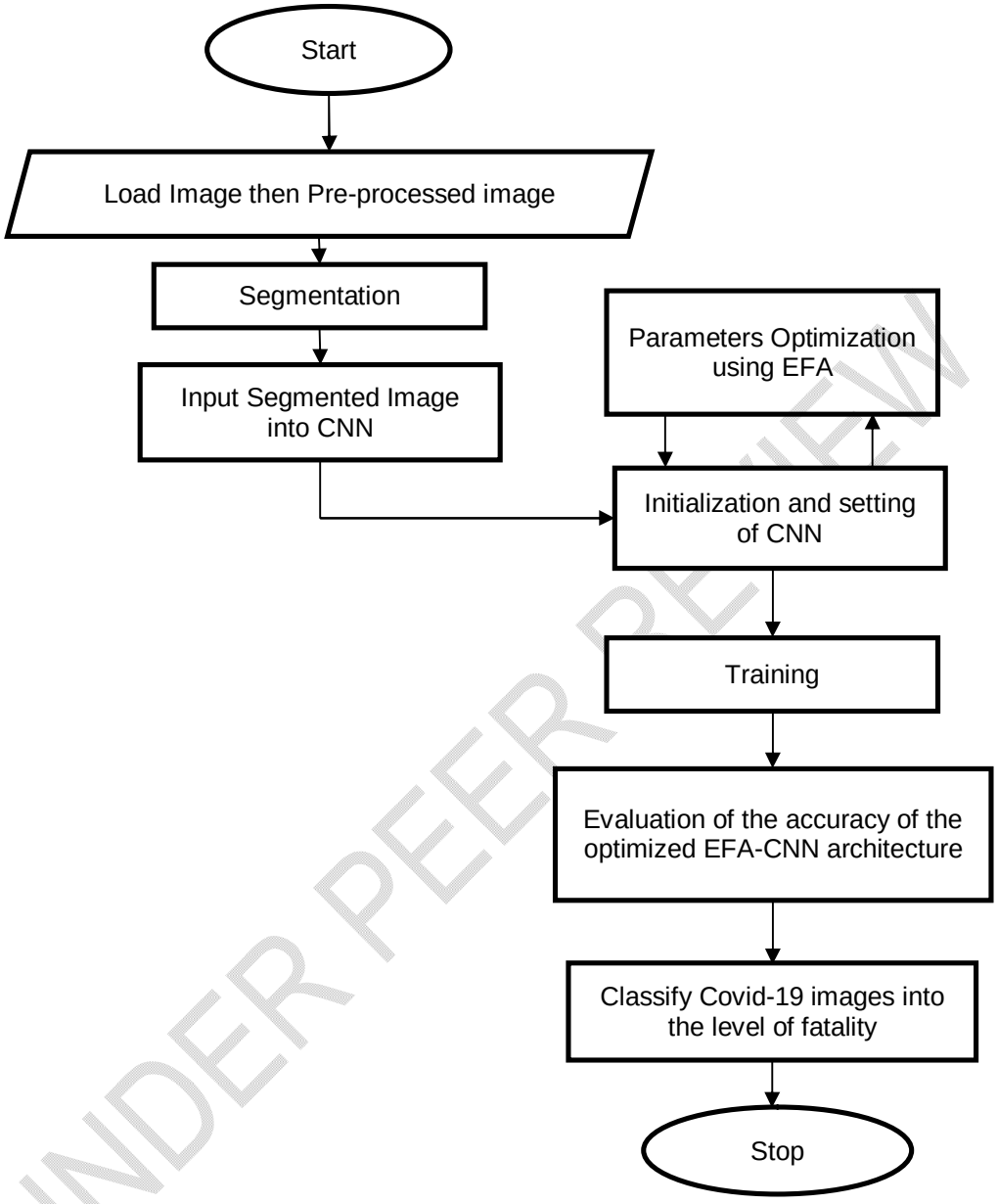


Figure 3: COVID-19 Classification using Enhanced Firefly Algorithm-based Convolutional Neural Network model (EFA-CNN)

2.6 Implementation of EFA-CNN for COVID-19 Patients' Fatality Prediction System

An interactive Graphic User Interface (GUI) application was developed for the COVID-19 Patients' Fatality Prediction System. The GUI was designed using toolboxes such as image processing and optimization in MATLAB 2020a. The MATLAB software package was used for the implementation on a computer system and run on Hewlett-Packard G56 with Intel® Core ™ i5 Duo, Windows 10 Professional 64-bit operating system, Central

Processing Unit (CPU) with a speed of 2.7GHz, 6GB Random Access Memory (RAM) and 1 TB hard disk drive.

2.7 Evaluation Measures

The models were evaluated using Specificity, Sensitivity, False Positive Rate (FPR), Accuracy, and Recognition time/rate performance evaluation metrics to determine their efficiency and quality. A confusion matrix was used to determine the values of these performance metrics. The confusion matrix is performance measurement in machine learning classification problems. It describes all combinatorially possible outcomes of a classification system and lays the foundations necessary to understand accuracy measurements for a classifier. It is a 2 by-2 table showing the True Positive (**TP),** False Positive (**FP),** False Negative **(FN)** and True Negative **(TN)**. When considering multi-class classification, the confusion matrix table takes the size equal to the number of classes squared.

TP contains the amount of the dataset instances that are correctly identified as positive. **FP** contains the amount of the dataset instances which are negative but predicted as positive. **TN** is the number of dataset instances that are negative and predicted as negative. **FN** is the number of dataset instances that are positive but predicted as negative.

The specificity evaluation metric shows the percentage of COVID-19-negative patients correctly predicted by the models, as it is expressed in Equation (14).

$$\textbf{Specificity} = \frac{\textbf{TN}}{\textbf{TN+FP}} \tag{14}$$

The sensitivity evaluation metric shows the percentage of COVID-19-positive patients correctly predicted by the models, as it is expressed in Equation (15).

$$\textbf{Sensitivity} = \frac{\textbf{TP}}{\textbf{TP+FN}} \tag{15}$$

The False Positive Rate evaluation metric is a measure of accuracy which shows the percentage of the COVID-19 negative patients incorrectly predicted as COVID-19 positive patients in the dataset by the models, as it is expressed in Equation (16),

$$\textbf{False Positive Rate (FPR)} = \frac{\textbf{FP}}{\textbf{TN+FP}} \tag{16}$$

The accuracy evaluation metric shows the percentage of the dataset instances correctly predicted by the models, as it is expressed in Equation (17)

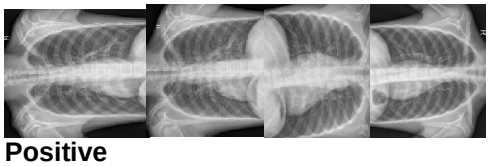
$$\textbf{Accuracy} = \frac{\textbf{TP+TN}}{\textbf{TP+TN+FP+FN}} \tag{17}$$

$$\textbf{Recognition Time} = \frac{\textbf{No of Predicted Covid-19 image}}{\textbf{Total number Covid-19 images}} \boldsymbol{X\ Time} \tag{18}$$

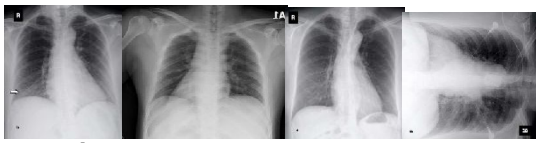
3. RESULTS AND DISCUSSION

The developed deep learning models (EFA-CNN and CNN) were experimented with by recognizing lung COVID-19 diseases using lung chest X-ray images. The recognition was

further analyzed based on the degree of severity such as severe, mild and moderate cases. The average features extracted values from the positive lung COVID-19 diseases using Chest X-ray images ranges from 0.23 to 0.98. The average features value range of 0.23-0.34 depicted moderate COVID-19 diseases, 0.34-0.78 portrayed mild COVID-19 disease and 0.79-0.98 described severe COVID-19 disease. The pre-processed and segmented images used for the study were presented in Figure 4. The Graphical User Interface (GUI) of the lung chest x-ray image based on COVID-19 during training, testing and classification were expressed in Figure 5 and Figure 6 respectively.

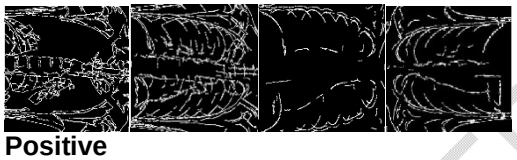


Positive

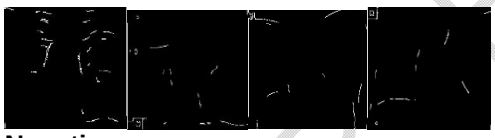


Negative

(a) Original Image



Positive



Negative

(b) Segmented Image

Figure 4: Pre-processed and Segmented Image

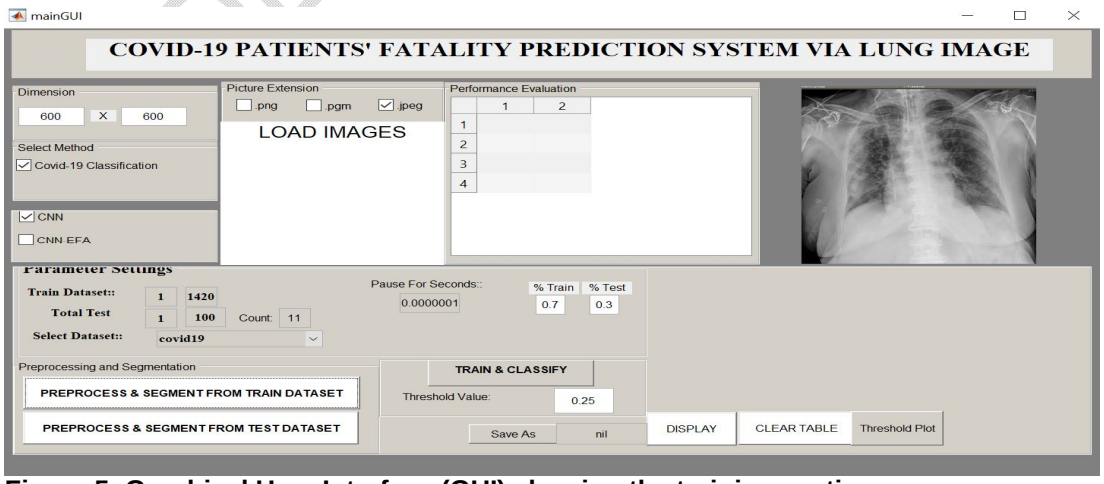


Figure 5: Graphical User Interface (GUI) showing the training section

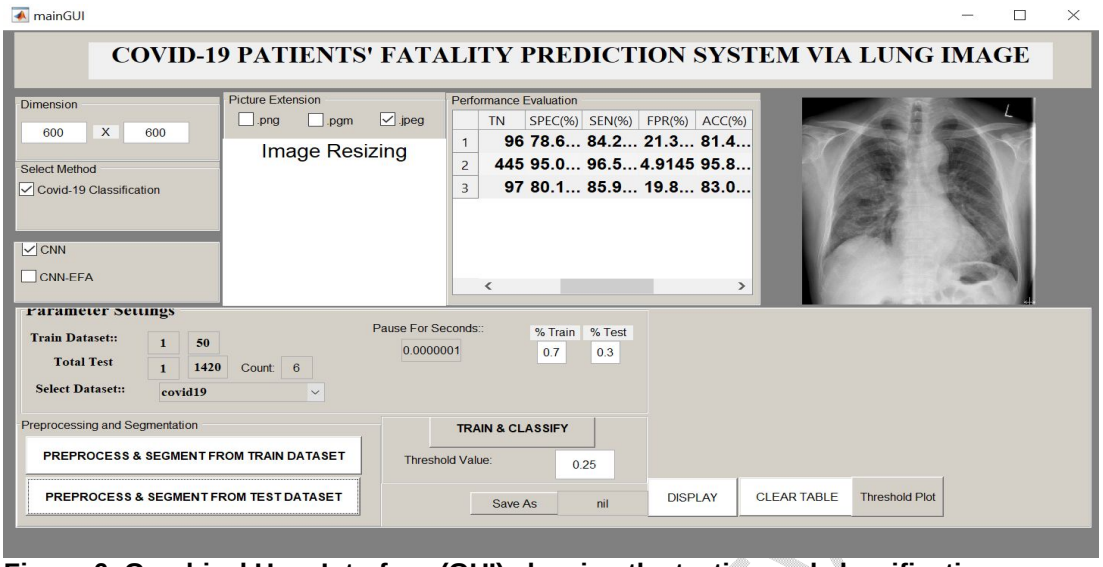


Figure 6: Graphical User Interface (GUI) showing the testing and classification phase

The developed system was tested and evaluated using the following performance metric: Sensitivity (SEN), Specificity (SPEC), False Positive Rate (FPR), Accuracy (ACC) and Recognition time (Time). All performance metrics were analyzed using a square dimension pixel resolution at different average thresholds of 0.30, 0.40, 0.50 and 0.80 from a range of thresholds of 0-0.30, 0.31-0.40, 0.41-0.50 and 0.51-1.00 respectively. Each threshold value range generated the same accuracy but changes at 0.30, 0.40, 0.50 and 0.80 threshold values respectively as defined in Figure 7 were significant. Figure 7 showed the choice of threshold value used in this study. Sixty per cent (60%) of the total images were used for training and forty (40%) were used for testing using the random sampling cross-validation method.

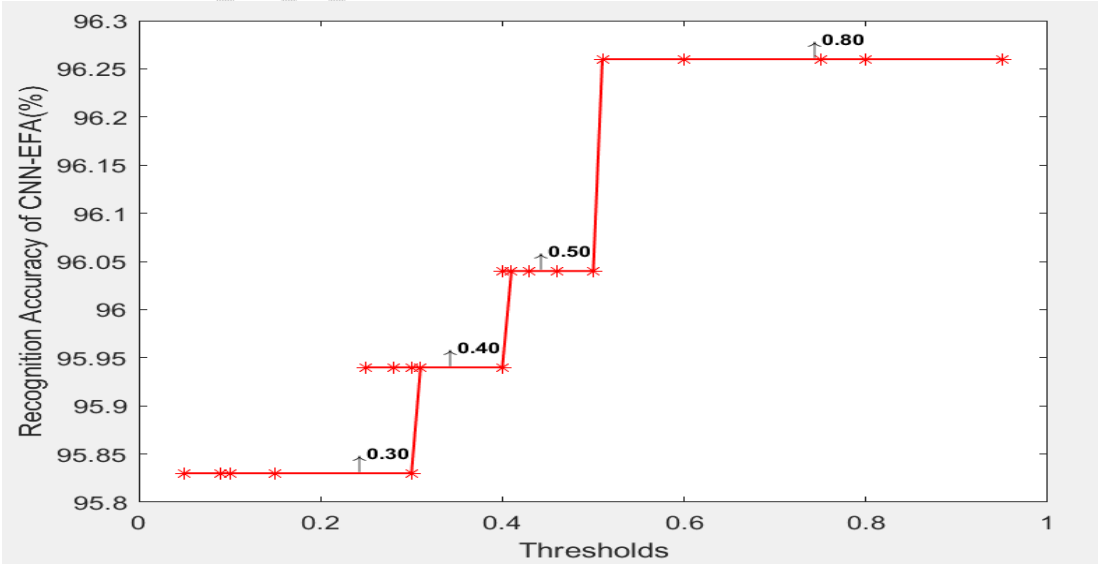


Figure 7: Graph showing a choice of the threshold used for the evaluation

Table 1 showed the optimization results of EFA respectively on CNN at 30 iterations with different filter sizes, number of filters, number of convolution layers and batch sizes. Accuracy was used as the objection function of EFA. The best recognition rate was achieved by EFA at a value of 99.23% as depicted in Table 1. Based on the results, the optimal CNN architecture attained by the application of EFA were as follows: 3 convolutional layers, 128 filters per layer with a filter size of 7 × 7, and the batch size with a value of 256 for EFA

Table 1. Optimization results obtained by the EFA-CNN

S/N	No. of Layers	No. of Filters	Filter Size	Batch Size	Recognition Rate (%)
1	3	58	7x7	230	95.31
2	1	67	3x3	196	99.16
3	1	97	7x7	137	95.27
4	2	60	4x4	115	95.69
5	1	61	7x7	230	95.50
6	1	128	6x6	107	98.10
7	2	128	7x7	109	98.12
8	2	128	4x4	221	98.60
9	3	128	5x5	256	95.86
10	3	128	5x5	256	97.30
11	3	128	4x4	256	95.95
12	3	128	4x4	196	96.00
13	2	128	7x7	256	95.90
14	1	128	3x3	256	96.50
15	3	128	5x5	196	95.39
16	2	128	3x3	256	98.49
17	3	128	5x5	256	97.42
18	2	128	7x7	256	98.39
19	2	128	4x4	256	96.70
20	2	128	7x7	256	98.22
21	2	128	3x3	256	98.26
22	1	128	7x7	256	95.78
23	1	128	4x4	256	95.66
24	1	128	7x7	256	95.79
25	1	128	5x5	256	95.84
26	3	128	3x3	256	98.17
27	1	128	4x4	256	95.87
28	2	128	7x7	256	97.66
29	2	128	3x3	256	95.43

30	**3**	**128**	**7x7**	**256**	**99.23**

.

4.1 Results for CNN

The result presented in Table 2 depicts the result of the CNN model based on the 3550 lung Chest X-ray image dataset. The lung Chest X-ray image comprises 2130 trained datasets and 1420 positive and negative test datasets. There are 709 positive datasets and 711 negative datasets. The 709 positive test datasets consist of 121 severe cases, 467 mild cases and 121 moderate cases. As shown in Table 2, at the optimum threshold of 0.80, out of 121 severe cases test lung chest X-ray image datasets, 99 severe lung chest X-ray image datasets were classified correctly as severe, 22 lung chest X-ray image datasets were misclassified as negative cases while out of 122 negative lung chest X-ray image datasets, 103 lung chest X-ray image datasets were classified correctly as negative and 19 lung chest X-ray image datasets were misclassified as severe cases. In addition, out of 467 mild cases, 448 mild lung chest X-ray image datasets were classified correctly as mild, 19 lung chest X-ray image datasets were misclassified as negative cases while out of 468 negative lung chest X-ray image datasets, 452 lung chest X-ray image datasets were classified correctly as negative and 16 lung chest X-ray image datasets were misclassified as mild cases. Also, out of 121 moderate cases, 101 moderate lung chest x-image datasets were classified correctly as moderate, 20 lung chest X-ray image datasets were misclassified as negative cases while out of 121 negative lung chest X-ray image datasets, 104 lung chest X-ray image datasets were classified correctly as negative and 17 lung chest X-ray image datasets were misclassified as moderate cases.

Additionally, Table 2 depicts the result obtained by the CNN model at different threshold values concerning the performance metrics. The results obtainable from Table 2 at an optimum threshold value of 0.80 revealed that CNN had FPR of 15.57%, 3.42% and 14.05%, Sensitivity of 81.82%, 95.93% and 83.47%, Specificity of 84.43%, 96.58% and 85.95%, and accuracy of 83.13%, 96.26% and 84.71% at 35.94 seconds, 34.90% seconds and 32.51 seconds, for severe, mild and moderate cases of COVID-19 disease, respectively.

Table 2: Performance of the CNN model

TP	FN	FP	TN	SPEC (%)	SEN (%)	FPR (%)	ACC (%)	R Time (sec)	Threshold	Type
102	19	26	96	78.69	84.30	21.31	81.48	35.94	0.30	Severe
451	16	23	445	95.09	96.57	4.91	95.83	34.90	0.30	Mild
104	17	24	97	80.17	85.95	19.83	83.06	32.51	0.30	Moderate
101	20	24	98	80.33	83.47	19.67	81.89	35.94	0.40	Severe
450	17	21	447	95.51	96.36	4.49	95.94	34.90	0.40	Mild
103	18	22	99	81.82	85.12	18.18	83.47	32.51	0.40	Moderate
100	21	22	100	81.97	82.64	18.03	82.30	35.94	0.50	Severe
449	18	19	449	95.94	96.15	4.06	96.04	34.90	0.50	Mild
102	19	20	101	83.47	84.30	16.53	83.88	32.51	0.50	Moderate
99	22	19	103	84.43	81.82	15.57	83.13	35.94	0.80	Severe
448	19	16	452	96.58	95.93	3.42	96.26	34.90	0.80	Mild
101	20	17	104	85.95	83.47	14.05	84.71	32.51	0.80	Moderate

4.2 Results for EFA-CNN

The results depicted in Table 3 were obtained from 709 positive datasets and 711 negative datasets. The 709 positive test datasets consist of 121 severe cases, 467 mild cases and 121 moderate cases. As shown in Table 3, at an optimum threshold of 0.80 with the EFA-CNN model, having selected CNN optimum parameter of 3 convolutional layers, 128 filters per layer with a filter size of 7 × 7, and the batch size with a value of 256 by EFA, out of 121 severe cases test lung chest X-ray image datasets, 113 severe lung chest x-ray image datasets were classified correctly as severe, 8 lung chest X-ray image datasets were misclassified as negative cases while out of 122 negative lung chest x-ray image datasets, 117 lung chest X-ray image datasets were classified correctly as negative and 5 lung chest X-ray image datasets were misclassified as severe cases. In addition, out of 467 mild cases, 458 mild lung chest X-ray image datasets were classified correctly as mild, 9 lung chest x-ray image datasets were misclassified as negative cases while out of 468 negative lung chest x-ray image datasets, 462 lung chest X-ray image datasets were classified correctly as negative and 6 lung chest X-ray image datasets were misclassified as mild cases. Also, out of 121 moderate cases, 114 moderate lung chest X-ray image datasets were classified correctly as moderate, 7 lung chest X-ray image datasets were misclassified as negative cases while out of 121 negative lung chest X-ray image datasets, 117 lung chest X-ray image datasets were classified correctly as negative and 4 lung chest X-ray image datasets were misclassified as moderate cases.

Furthermore, Table 3 presented the result achieved by the EFA-CNN model at different threshold values concerning the performance metrics. The results obtainable from Table 3 at an optimum threshold value of 0.80 revealed that EFA-CNN had FPR of 4.10%, 1.28% and 3.31%, Sensitivity of 93.39%, 98.07% and 94.21%, Specificity of 95.90%, 98.72% and 96.69%, and Accuracy of 94.65%, 98.40% and 95.45% at 16.61 seconds, 16.14 seconds and 15.16 seconds, for severe, mild and moderate cases of COVID-19 disease, respectively.

Table 3. Performance of the EFA-CNN model

TP	FN	FP	TN	SPEC (%)	SEN (%)	FPR (%)	ACC (%)	R Time (sec)	Threshold	Type
116	5	13	109	89.34	95.87	10.66	92.59	16.61	0.3	Severe
461	6	14	454	97.01	98.72	2.99	97.86	16.14	0.3	Mild
117	4	12	109	90.08	96.69	9.92	93.39	15.16	0.3	Moderate
115	6	10	112	91.80	95.04	8.20	93.42	16.61	0.4	Severe
460	7	11	457	97.65	98.50	2.35	98.07	16.14	0.4	Mild
116	5	9	112	92.56	95.87	7.44	94.21	15.16	0.4	Moderate
114	7	7	115	94.26	94.21	5.74	94.24	16.61	0.5	Severe
459	8	8	460	98.29	98.29	1.71	98.29	16.14	0.5	Mild
115	6	6	115	95.04	95.04	4.96	95.04	15.16	0.5	Moderate
113	8	5	117	95.90	93.39	4.10	94.65	16.61	0.8	Severe
458	9	6	462	98.72	98.07	1.28	98.40	16.14	0.8	Mild
114	7	4	117	96.69	94.21	3.31	95.45	15.16	0.8	Moderate

4.4 Discussion of Results

The results depicted in Table 2 and Table 3 described the performance of the developed deep learning models (CNN and EFA-CNN) respectively. The results illustrated that there is significant variation in the performance metrics across all metrics (Sensitivity, Specificity, FPR, Accuracy and Recognition time) for the CNN model and EFA-CNN model. At an optimum threshold value of 0.80, the EFA-CNN gave Recognition Accuracy of 94.65%, 98.40% and 95.45% while the CNN technique had 83.13%, 96.26% and 84.71% Recognition Accuracy for all classification cases respectively. Similarly, the EFA-CNN model produced a False Positive Rate of 4.10%, 1.28% and 3.31% and a Recognition Time of 16.61 seconds, 16.14 seconds and 15.16 seconds for all classification cases while the CNN model produced a False Positive Rate of 15.57%, 3.42% and 14.05% and Recognition Time of 35.94 seconds, 34.90 seconds and 32.51 seconds in all classification cases respectively. It can be inferred from the results based on the performance metrics that the EFA-CNN model gave an increased Recognition Accuracy of 11.52%, 2.14% and 10.74% and decreased False Positive Rate of 11.47%, 2.14% and 10.74% over the CNN model.

Given the results, the EFA-CNN model is more accurate due to the low number of false positives at reduced recognition time. The improved recognition accuracy and reduced recognition time of the developed EFA-CNN was as a result of (1) the imbalance problem between exploration and exploitation experienced in the standard firefly algorithm that was resolved in this study by enhancement of the attractiveness of firefly with the application of chaotic theory and sinusoidal mapping, and (2) the hyper-parameters for the CNN architecture that was optimally selected with EFA (the number of convolutional layers, the size of the filters used in each convolutional layer, the number of convolutional filters, and the batch size) compared with CNN.

The result of the EFA-CNN model confirmed the previous literature [44] that indicated that improved recognition accuracy and reduced recognition time can only be obtained if hyper-

parameters (the number of convolutional layers, the size of the filters used in each convolutional layer, the number of convolutional filters, and the batch size) for the CNN architecture can be optimally selected.

4. CONCLUSION

In this work, the deep learning model which is the best in predicting COVID-19 patient fatality was determined. The study was able to deduce that, out of the two deep learning models considered EFA-CNN is more accurate in prediction with higher recognition accuracy, reduced false positive rate and reduced recognition time. Also, the researchers have been able to ascertain that, enhancement of the attractiveness of firefly with the application of chaotic theory and sinusoidal mapping in EFA and optimal selection of the hyper-parameters for the CNN architecture (the number of convolutional layers, the size of the filters used in each convolutional layer, the number of convolutional filters, and the batch) by EFA contributed immensely to the improved recognition accuracy as well as reduced recognition time of the developed COVID-19 patients' fatality prediction system. The researchers are of the view that the developed system in this research can aid the government and healthcare workers in providing the needed computational capability for the prediction of the fatality level of a positively tested COVID-19 patient and also help to guide and plan to reduce the severe public health and socio-economic burden resulted from COVID-19 pandemic. The model and the system developed could also help healthcare clinicians and radiologists for further diagnosis, tracking and control of the disease progression. In addition, the formulated Enhanced Firefly Algorithm (EFA) could be applied to other fields of optimization, design and applications

REFERENCES

1. Chen N, Zhou M, Dong X, Qu J, Gong F, Han Y. et al. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study. The Lancet. 2020; 395(10223):507-513.
2. World Health Organization Clinical management of severe acute respiratory infection when novel coronavirus (2019-nCoV) infection is suspected: interim guidance. In Clinical management of severe acute respiratory infection when novel coronavirus (2019-nCoV) infection is suspected: Interim guidance. 2020, 21-21.
3. Zumla A, Hui, DS, Perlman S. (2015). Middle East Respiratory Syndrome. Lancet, 2015;386(9997):995-1007, doi:10.1016/S0140-6736(15)60454-8.
4. Olapegba PO, Ayandele O, Kolawole SO, Oguntayo R, Gandi JC, Dangiwa AL, et al. A preliminary assessment of novel coronavirus (COVID-19) knowledge and perceptions in Nigeria, 2020. Data in Brief, doi: 10.1016/j.dib.2020.105685.
5. Pham Q, Nguyen DCT, Huynh-The WH, Pathirana PN. Artificial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: A survey on the state-of-the-arts. IEEE Access, 2020; 8:130820-130839.
6. World Health Organization Situation Report-94 Coronavirus disease 2019 (COVID-19). Accessed 10 March 2021, Available:https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200423-sitrep-94-covid-19.pdf?sfvrsn=b8304bf0_4 ()
7. Yang C, Sha D, Liu Q, Li Y, Lan H, Guan WW, et al.Taking the pulse of COVID-19: A spatiotemporal perspective, 2020a, arXiv 2020. arXiv preprint arXiv:2005.04224.

8. Bogoch II, Watts A, Thomas-Bachli A, Huber C, Kraemer MU, Khan K. Pneumonia of unknown aetiology in Wuhan, China: potential for international spread via commercial air travel. Journal of travel medicine, 2020; 27(2), taaa008.
9. Eikenberry SE, Mancuso M, Iboi E, Phan T, Eikenberry K, Kuang Y. et al.. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic. Infectious Disease Modelling,2020; 5:293-308.
10. Shah K, Chaudhari G, Kamrai D., Lai A, Patel RS How essential is it to focus on physician's health and burnout in the coronavirus (COVID-19) pandemic?Cureus, 2020;12(4):
11. Khalifa NEM, Taha, MHN., Manogaran G, Loey M. A deep learning model and machine learning methods for the classification of potential coronavirus treatments on a single human cell. Journal of Nanoparticle Research, 2020;22(11): 1-13.
12. Muhammad LJ, Algehyne EA, Usman SS, Ahmad A, Chakraborty C, Mohammed IA. Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset. SN computer science, 2021;2(1):1-13.
13. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural computation, 2006;18(7):1527-1554.
14. Bengio Y. Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2009;2(1):1-127.
15. Bengio Y, LeCun Y. Scaling learning algorithms towards AI. Large-scale kernel machines,2007;34(5):
16. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on pattern analysis and machine intelligence, 2013;35(8): 1798-1828.
17. Arel I, Rose DC, Karnowski TP. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. IEEE Computational Intelligence Magazine, 2010;5(4):13-18.
18. Hinton GE, Osindero S, Teh, YW. A fast learning algorithm for deep belief nets. Neural computation, 2006;18(7): 1527-1554.
19. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 2007;19:153.
20. Hussain L, Nguyen T, Li H, Abbasi AA, Lone KJ, Zhao Z. et al. Machine-learning classification of texture features of portable chest X-ray accurately classifies COVID-19 lung infection. BioMedical Engineering OnLine, 2020;19(1): 1-18.
21. Ardabili SF, Mosavi A, Ghamisi P, Ferdinand F, Varkonyi-Koczy AR. et al. Covid-19 outbreak prediction with machine learning. Algorithms, 2020;13(10):249.
22. Narin A, Kaya C, Pamuk Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks.2020; arXiv 2003.10849
23. Yang Z, Zeng Z, Wang K, Wong S-S, Liang W, Zanin M, et al. Modified SEIR and AI prediction of the epidemic trend of COVID-19 in China under public health interventions. J Thorac Dis, 2020;12(3):165–74. https :// doi.org/10.21037 /jtd.2020.02.64.
24. Ayyoubzadeh SM, Ayyoubzadeh SM, Zahedi H, Ahmadi M, Kalhori SRN, et al. Predicting COVID-19 incidence through analysis of Google trends data in Iran: data mining and deep learning pilot study. JMIR Public Health Surveill, 2020;6(2):e18828.
25. Dianbo L, Leonardo C, Canelle P, Xiyu D, Matteo C, Jessica TD, et al. A machine learning methodology for real-time forecasting of 2019–2020 COVID-19 outbreak using Internet searches, news alerts, and estimates from mechanistic models, 2020; Accessed 20 September 2021; Available: https://arxiv .org/abs/2004.04019
26. Edison O, Mei UW, Anthony H, Yongqun H. COVID-19 Coronavirus Vaccine Design Using Reverse Vaccinology and Machine Learning. Frontiers in Immunology,2020; doi: 10.3389/femme.2020.01581
27. Malik M. Machine learning the phenomenology of COVID-19 from early infection dynamics, 2020; Accessed 15 April 2021; Available: https://arxiv .org/ abs/2003.07602

28. Fregoso J, Gonzalez CI, Martinez GE. Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition. Axioms, 2921;10(3):139.
29. Odeniyi OA, Adeosun ME, Ogundunmade TP. Prediction of terrorist activities in Nigeria using machine learning models, Innovations, 2022;71:87-96 (www.journal-innovations.
30. LeCun Y, Bengio Y, Hinton G. Deep learning. nature, 2015;521(7553):436-444.
31. Tan T, Zhang X, Sun Z, Zhang H. Noisy iris image matching by using multiple cues. Pattern Recognition Letters, 2012;33 (8), 970-977
32. Gao W, Zhang X, Yang L, Liu, H. An improved Sobel edge detection. 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, 2010; 67-71, doi: 10.1109/ICCSIT.2010.5563693.