International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Analysis of Optimal Model with Convolutional Neural Network and Differential Evolutionary Algorithm for Lungs Cancer Detection

[1]Vanita G. Tonge, [2]Asha Ambhaikar

Submitted: 25/11/2023 **Revised**: 05/01/2024 **Accepted**: 15/01/2024

Abstract: Optimizers play a pivotal role in constructing an efficient classification model. This article employs a popular deep-learning model paired with a metaheuristic—specifically, the differential evolutionary algorithm—for the crucial task of detecting lung nodules, a lethal aspect of lung cancer, a life-threatening disease. Timely treatment significantly contributes to increased survival rates, necessitating proper care and early diagnosis. To address these challenges, the Differential Evolutional Convolutional Neural Network (DECNN) emerges as the optimal solution. While Convolutional Neural Networks (CNNs) consistently yield superior results in medical applications, the intricate task of hyperparameter tuning poses a considerable challenge. Traditional optimizers such as genetic algorithms, particle swarm optimization, and random search optimization have been utilized by researchers. Differential Evolution (DE), characterized by a minimal set of parameters including population size, crossover, and mutation factors, stands out as a simple yet effective optimizer. The proposed model was implemented and tested on the IQ-OTH/NCCD datasets. To comprehensively evaluate the performance of the DECNN model optimized by the differential evolutionary optimizer, an initial model was generated and tested without the application of any optimization techniques. Subsequently, the performance and optimization criteria of this baseline model were also assessed using Genetic Algorithm and Particle Swarm Optimizer for a thorough comparative analysis.

Keywords: *Optimizers, Differential Evolutionary Algorithm, Hyperparameters, Imbalanced dataset, Mutation*.

1 Introduction

The accelerated proliferation of abnormal cells gives rise to cancer, a condition that often originates in the lungs but may also spread to other organs, a phenomenon known as metastasis. Lung cancer, a malady that can affect individuals who smoke, have quit smoking, or never smoked, exhibits a higher mortality rate compared to other global diseases. The treatment approach for lung cancer is contingent upon its classification into two types: small-cell and non-small-cell lung cancer. Like many other illnesses, early detection significantly enhances the curability of lung cancer, with an 85 to 90 percent success rate for patients diagnosed with small, early-stage lung cancer.

Various treatment modalities exist for lung cancer, each tailored to its specific type. Surgical intervention involves the removal of cancerous tissue, chemotherapy employs specialized medications to mitigate or eliminate the disease, and radiation treatment utilizes high-energy radiation akin to X-rays for disease eradication. Targeted treatment, on the other hand, utilizes medications to impede the growth and proliferation of cancer cells.

In the context of early detection, this article proposes the integration of an automated deep-learning model with a differential evolutionary algorithm. The use of deep learning models has gained significant traction in medical imaging applications, offering a promising avenue for enhancing the timely identification of lung cancer [1]. CNN's automated feature extraction and end-to-end training are its foremost elements [2]. The research article [3] focuses work on lung cancer detection and classification using CNN and Google Net and achieves a precision of 98%. The author [4] conducted a study on emerging image processing and machine-learning techniques for lung nodule identification. This research summarizes a survey on the various machine learning approaches and concludes that deep learning techniques obtained higher results compared to traditional machine learning methods [5]. This study designed an ANN model with an accuracy of 96.67% for checking the lung lesion considering different symptoms such as chronic disease, fatigue, allergy, anxiety, etc [6]. Work [7] elaborates that the Mortality rate depends on the algorithm’s persistent rate. Researchers select Otsu thresholding to recognize area of interest and cuckoo search method to generate avid characteristics for partitioning lesions and get 96.97% result. Existing work [8] discuss innovative method combination of internet of medical things with deep learning. IoMT devices collects medical information and transfer it to ML-CNN Method for malignancy sensing. Optimal model based on genetic algorithm and CNNs overcome the diagnostic challenges in lungs cancer recognition [9]. Authors proposed a multistrategy-based artificial electric field algorithm for hyperparameter

[1]*Research Scholar, Department of Computer Science and Engineering, Kalinga University, Naya Raipur, Raipur, India vsburadkar@gmail.com*
[2]*Professor, Department of Computer Science and Engineering, Kalinga University, Naya Raipur, Raipur, India,*
asha.ambhaikar@kalingauniversity.ac.in

tuning in CNN and finds the optimal solution. Also address the issue related to imbalanced dataset [10]. This paper [11] combined image processing, deep learning and metaheuristic Marine Predators and applied it to RIDER datasets for early detection of cancer. Result compared with pretrained networks. Because of the lungs cancer structure early diagnosing is tedious task, Cheng-Jian Lin et.al. [12] formed automated system using 2D-CNN and Taguchi_parametric optimizer. Experimental result over (LIDC-IDRI) dataset and International Society for Optics and Photonics with the support of the American Association of Physicists in Medicine (SPIE-AAPM) dataset shows the superiority over the 2d-cnn. Sebastian, et al [13] designed new model comprising four stages image pre-processing, segmentation using otsthu thresholding, feature extraction and classification using CNN-Moth flame optimizer.

2 CNN and DE

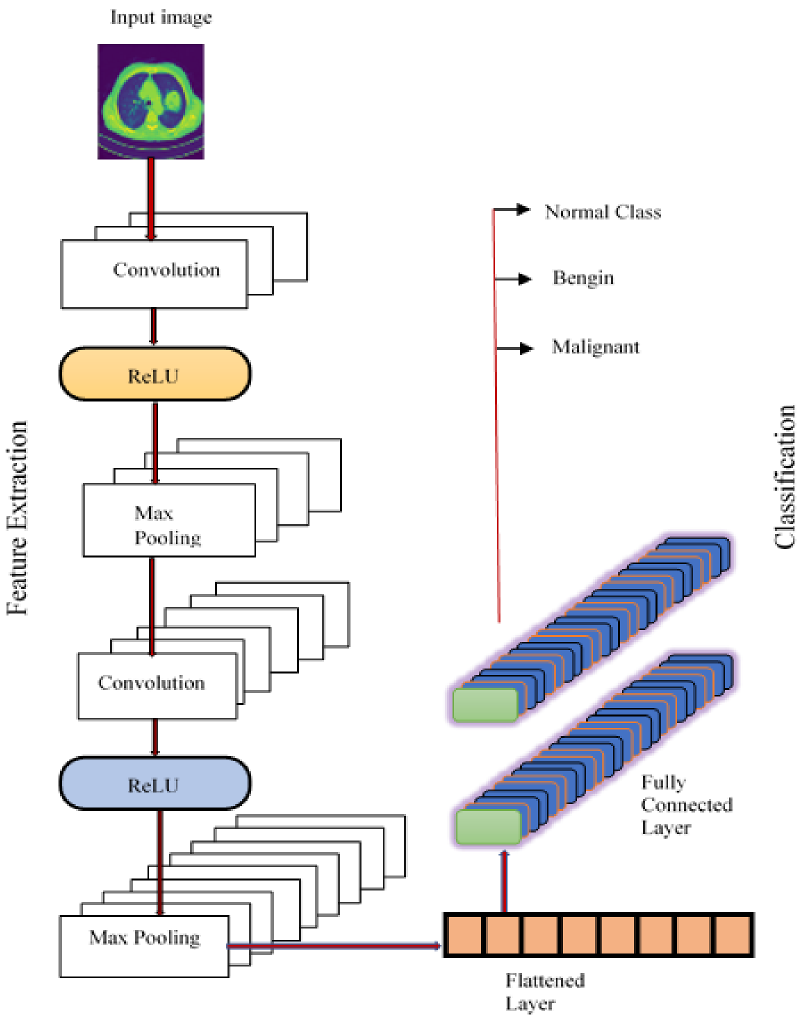

Fig.1 Architecture of CNN

Deep learning outperforms machine learning, particularly excelling in handling vast datasets, preventing overfitting, managing intricate dimensions of images, and delivering heightened accuracy. It represents the culmination of success in machine learning algorithms, drawing inspiration from human cognition. Deep learning operates with tensors, signifying the nesting of matrices. A key component of deep learning is the Convolutional Neural Network (CNN), also known as a multilayer perceptron, which is categorically divided into feature extraction and classification.

The architecture of a deep CNN involves a stack of layers, as depicted in Figure 1. The initial stage processes multiple images, extracting diverse features and passing them through a filtering step. The filter or kernel reduces dimensionality, assesses accuracy, and iteratively extracts more features, culminating in flattening and precise image classification. Each layer encompasses parameters and hyperparameters that contribute to the creation of distinct models.

Selecting appropriate hyperparameters is a meticulous task, encompassing factors such as the type and size of the filter, stride, padding, activation function, depth of the network (hidden layers), width of layers, learning rate, batch size, epochs, and dropout rate. To streamline this process, researchers have explored evolutionary algorithms to optimize parameter selection efficiently. In this article, we delve into the application of a differential evolutionary algorithm—an approach inspired by Darwin's theory—to contribute to the quest for an optimal solution and alleviate the time-intensive variable selection process.

Individual solutions are referred to as genomes or chromosomes within the context of this problem. The primary operators involved in the optimization process are mutation and recombination. Diagram number 2 illustrates the Differential Evolution (DE) algorithm, wherein a target vector is subjected to mutation to produce a donor vector. The donor vector subsequently undergoes recombination to generate a trial vector. The selection of the fittest value between the target and trial vectors determines the composition of the next generation in the evolutionary process.

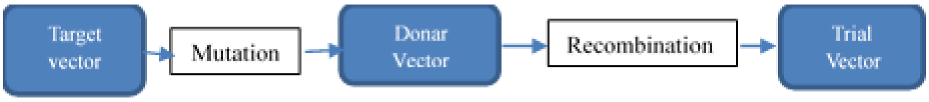

Fig.2 Simple Differential Evolutionary algorithm

3 Optimization Using DE

While Deep Convolutional Neural Networks (Deep-CNN) exhibit superior performance in medical applications, the challenge of tuning hyperparameters poses a formidable task. In cases where a selected set of hyperparameters fails to yield satisfactory results, the model necessitates the time-consuming process of reselecting a new architecture [17]. The proposed automated model introduces an innovative Deep Learning structure integrated with a Differential Evolutionary Algorithm specifically designed for hyperparameter tuning. Within the CNN structure, one or more convolutional layers [18] are considered as the population NP, where the population represents a set of chromosomes. The utilization of binomial crossover factors and DE/rand/1 mutation is integral to the optimization process, and these DE parameters significantly influence the algorithm's convergence [19]. The optimization of a Convolutional Neural Network (CNN) entails the fine-tuning of various parameters and hyperparameters to enhance its overall performance. A systematic algorithmic approach for optimizing CNNs is outlined below.

3.1 Data Acquisition and Data Augmentation

The dataset IQ-OTHNCCD [15] taken from Kaggle. Total 2073 images belonging to three classes shown in following fig.3.

Number of Benign Cases -> 120

Number of Malignant Cases -> 1339

Number of Normal Cases -> 614

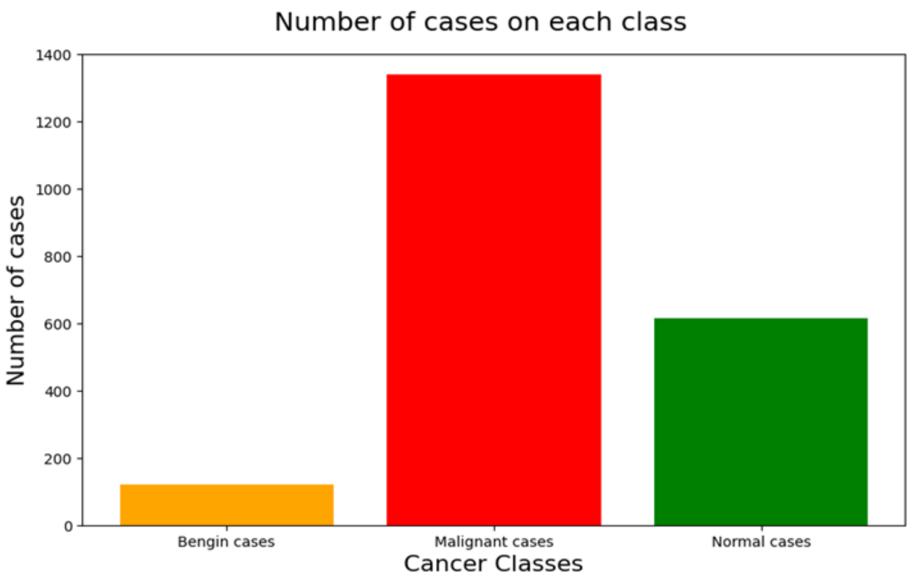

Fig. 3 Dataset with number of images in each class

The quantity of images supplied to the neural network is a crucial determinant in the feature extraction process [16]. Data imbalance can lead to overfitting and suboptimal classification. Theoretical methodologies like SMOTE and data augmentation are implemented to mitigate the risks associated with overfitting.

3.2 Data Pre-processing

The application of image processing filters on each class of images serves the purpose of enhancement and noise reduction, facilitating a smoother feature extraction process. These filters include Adaptive Thresholding, Image Negative, GaussianBlur, and Gray Scale Image, as illustrated in Figure 4 below.

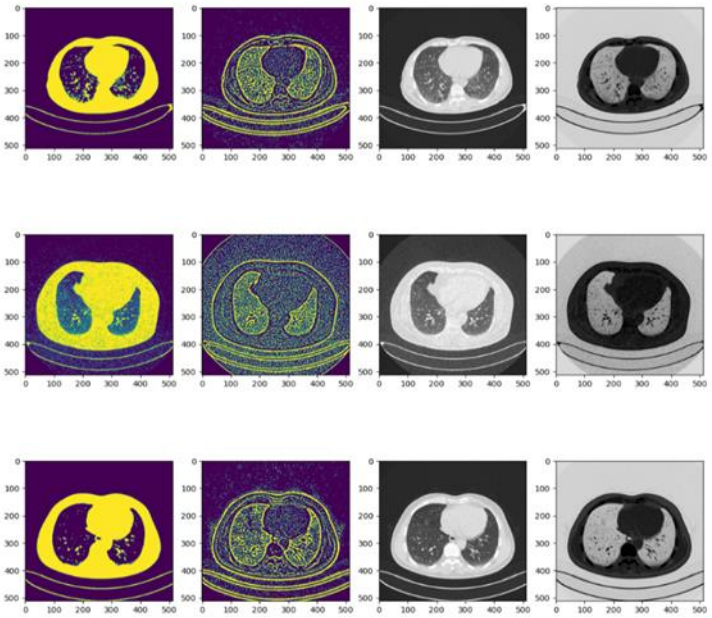

Fig.4 Applying filters on each class of images

3.3 Data Splitting

The dataset is partitioned into training, testing, and validation sets. The training set is employed to assess the parameters of the Convolutional Neural Network (CNN), whereas the validation set plays a crucial role in early stopping and hyperparameter tuning. Finally, the testing set is utilized to evaluate the ultimate performance of the evolved CNN. When selecting the architecture for the Deep-CNN, factors such as the complexity of the task, available computational resources, and the size of the dataset should be carefully considered.

3.4 Model Initialization

The architecture of Chromosome consist parameters like Hyperparametrs :{'filter1': 64, 'filter2': 64, 'filter3': 512, 'kernel_Size': 5, 'activation_function1': 'relu', 'activation_function2': 'selu', 'drop_outrate1': 0.4, 'droup_outrate2': 0.5, 'optimizer': 'adamax', 'epochs': 52}

3.5 DE Population Initialization

Generate a population of individuals, wherein each individual represents a set of chromosomes or genes specific to the Convolutional Neural Network (CNN). Random values for chromosomes, encompassing parameters like learning rate, batch size, epochs, filter size, etc., are assigned to each individual in the population.

3.6 CNN Training and Evaluation

For every individual within the population, conduct training of the Convolutional Neural Network (CNN) using the designated set of genes and the training dataset. Subsequently, assess the trained CNN's performance on the validation set and compute performance metrics indicative of its accuracy in detecting lung cancer. These metrics may encompass overall accuracy, precision, recall, and F1 score.

3.7 Fitness assignment

The performance metric calculated in the preceding step will be used to assign accuracy scores to each individual within the population.

4.1 DE Operators

Utilizing Differential Evolution (DE) operators, namely Crossover and Mutation, the chosen individual is subjected to these operations to generate new offspring for the succeeding generation. In the context of Mutation, a target vector, two randomly selected chromosomes, and a mutation factor (F) are involved. Mutation introduces slight changes to randomly selected vectors, ensuring diversity is preserved within the population. On the other hand, Crossover involves the target vector and donor vector, generating a trial vector.

4.2 Selection

Chromosome among target and trial with higher accuracy will be selected. Individual with higher accuracy have a great chance of being selected, aiming to preserve their genes in the generation and update the population.

4.3 Repeat the step 3.6-3.9

Iterate through multiple generations, repeating steps 3.6 to 3.9, to evolve the population and improve the performance of the CNN.

4.4 Termination condition

Here termination condition is maximum number of generations

4.5 Final Evaluation

Once the termination condition is met, best individual is selected from the final population based on its accuracy. Performance is evaluated on testing set to obtain the final performance metrics.

4 Result Analysis

Traditionally, the exploration of Deep Convolutional Neural Network (CNN) parameters involves a labor-intensive trial-and-error method, demanding considerable time and effort. Particle Swarm Optimization (PSO) initiates its operations with minimal assumptions, akin to Genetic Algorithms (GA) and Differential Evolution (DE), which commence by initializing a population. DE, leveraging differential information within the population, exhibits a faster convergence towards solutions. Similarly, PSO showcases accelerated convergence compared to GA. In this context, the utilization of a DE optimizer introduces an automated mechanism empowering the CNN to autonomously determine its parameters.

The iterative process initiates with an initial generation and the first population. Each parameter is autonomously selected by the CNN itself, undergoing training with subsequent accuracy assessment in every iteration. Tables 1 to 10 elucidate the values of Loss, Accuracy, Val_Loss, and Val_Accuracy observed in each iteration and generation. Remarkably, our model attains competitive accuracy within a mere two iterations and five generations. Tables 11 and 12 present the evaluation outcomes of hyperparameters across generations, delineating the optimized architecture achieved by the CNN.

Iteration:1	Epochs:13			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	364.6488	0.5806	9.0679	0.7527
2	4.341	0.6971	1.0422	0.8172
3	1.4191	0.6738	0.7452	0.7366
4	0.9768	0.7366	0.4101	0.8548
5	0.8152	0.7509	0.3699	0.8548
6	0.7241	0.7748	0.2869	0.8602
7	0.6434	0.7945	0.3007	0.8387
8	0.6106	0.7784	0.1673	0.9516
9	0.6511	0.7843	0.3359	0.8978
10	0.6159	0.8035	0.2517	0.8602
11	0.5355	0.8315	0.1963	0.9086
12	0.4992	0.8405	0.1126	0.9409
13	0.3691	0.8722	0.0931	0.9409

Table 1. Generation_1 and Iterarion_1

Iteration:2	Epochs:14			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	19.6503	0.4474	5.2392	0.6989
2	12.3208	0.5741	5.2606	0.6989
3	10.0005	0.6129	4.158	0.7312
4	9.3325	0.6201	3.9172	0.7527
5	7.4936	0.6487	2.2339	0.7581

6	6.8706	0.6183	2.7135	0.7849
7	5.5267	0.6768	1.5286	0.7688
8	4.4162	0.678	2.2059	0.7258
9	4.4729	0.6786	1.3654	0.7849
10	3.6721	0.7031	1.1616	0.7527
11	3.5163	0.6798	0.7975	0.828
12	3.3573	0.7019	0.83	0.7903
13	2.7913	0.6971	0.4636	0.871
14	2.4223	0.7228	0.4506	0.8441

Table 2. Generation_1 and Iterarion_2

Iteration:3	Epochs:19			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	157.4331	0.6541	2.4915	0.828
2	1.0165	0.8686	0.1919	0.957
3	0.2731	0.9415	0.0385	0.9892
4	0.1453	0.9594	0.0213	0.9892
5	0.1024	0.9642	0.0315	0.9892
6	0.0948	0.9767	0.0258	0.9892
7	0.0742	0.9767	0.0069	1
8	0.0556	0.9869	0.0135	0.9946
9	0.0609	0.9809	0.0136	0.9946
10	0.0571	0.9803	0.01	0.9946
11	0.0213	0.9928	0.0097	0.9946
12	0.025	0.9928	0.0187	0.9946
13	0.0295	0.9898	0.0076	0.9946
14	0.0298	0.9875	0.0139	0.9946
15	0.0704	0.9863	0.4544	0.7796
16	0.129	0.9659	0.0846	0.9839
17	0.2094	0.9432	0.1719	0.957
18	0.3251	0.8787	0.463	0.8011
19	0.4924	0.7867	0.3645	0.8871

Table 3. Generation_1 and Iterarion_3

Iteration:4	Epochs:10			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	1329.3793	0.451	6.8409	0.6183
2	9.975	0.3763	1.0644	0.457
3	2.0008	0.4056	0.8902	0.6237

4	1.7019	0.4438	0.7365	0.6237
5	1.514	0.4612	0.8966	0.6237
6	1.4792	0.4743	0.8803	0.6237
7	1.5226	0.4403	0.8902	0.6237
8	1.3635	0.4821	0.9041	0.6237
9	1.2741	0.5048	0.8817	0.6237
10	1.2435	0.5233	0.8436	0.6237

Table 4. Generation_1 and Iterarion_4

Iteration:5	Epochs:14			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	106.1597	0.6481	0.2979	0.9301
2	0.2571	0.9176	0.133	0.9677
3	0.1538	0.9612	0.0748	0.9946
4	0.0783	0.9851	0.0512	0.9839
5	0.0498	0.9904	0.0355	0.9946
6	0.0511	0.9886	0.0299	0.9946
7	0.031	0.9946	0.0108	1
8	0.0258	0.9958	0.0125	1
9	0.0248	0.9946	0.011	0.9946
10	0.0153	0.997	0.011	1
11	0.0184	0.997	0.0163	0.9946
12	0.0198	0.9952	0.0101	0.9946
13	0.0179	0.9964	0.0126	0.9946
14	0.0125	0.9988	0.0053	1

Table 5. Generation_1 and Iterarion_5

Generation:2

Iteration:1	Epochs:13			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	331.9412	0.5597	15.5495	0.7634
2	4.216	0.5986	0.7497	0.6613
3	1.4837	0.6051	0.5763	0.7849
4	1.4636	0.5812	0.9093	0.7419
5	1.3476	0.5974	0.4328	0.8602
6	1.0437	0.6505	0.2845	0.8925
7	0.9662	0.6876	0.5914	0.6989

8	1.0256	0.6744	0.4865	0.7043
9	0.9145	0.7025	0.4957	0.7634
10	0.9869	0.693	0.4486	0.8118
11	0.8724	0.7073	0.3701	0.8871
12	0.8433	0.7282	0.3415	0.8387
13	0.7538	0.7491	0.4015	0.8925

Table 6. Generation_2 and Iterarion_1

Iteration:2	Epochs:14			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	17.2062	0.4385	4.8176	0.7151
2	10.2148	0.5645	3.9335	0.7151
3	8.2707	0.6039	3.4026	0.7097
4	7.5197	0.6039	2.8705	0.6989
5	5.9374	0.632	2.4874	0.7312
6	5.2695	0.6577	1.2997	0.7688
7	4.4933	0.6768	1.443	0.7581
8	4.1451	0.6798	1.3263	0.7957
9	3.4452	0.6995	0.9007	0.8387
10	3.0027	0.7133	1.3045	0.6774
11	2.7932	0.7037	0.6067	0.8172
12	2.6338	0.7019	0.7512	0.7688
13	2.1418	0.724	0.5583	0.7903
14	1.9749	0.7246	0.5036	0.8011

Table 7. Generation_2 and Iterarion_2

Iteration:3	Epochs:19			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	187.0911	0.5621	0.7119	0.7688
2	1.0996	0.7174	0.3239	0.9194
3	0.6255	0.8082	0.3164	0.8817
4	0.5052	0.8465	0.1583	0.9516
5	0.4673	0.8596	0.2305	0.914
6	0.3386	0.8925	0.0852	0.9785
7	0.2816	0.905	0.0911	0.9839
8	0.2744	0.914	0.0907	0.9785
9	0.2109	0.9211	0.0913	0.957

10	0.2057	0.9283	0.0504	0.9892
11	0.1852	0.9361	0.034	0.9946
12	0.1614	0.9379	0.0247	0.9946
13	0.1625	0.9415	0.0499	0.9892
14	0.1647	0.9486	0.0341	0.9892
15	0.124	0.9576	0.0168	1
16	0.1114	0.9552	0.0363	0.9946
17	0.1418	0.951	0.029	0.9946
18	0.162	0.9421	0.0204	0.9892
19	0.1517	0.9492	0.0162	0.9946

Table 8. Generation_2 and Iterarion_3

Iteration:4	Epochs:10			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	175.2927	0.641	0.4291	0.8548
2	0.3435	0.8925	0.1565	0.9624
3	0.1188	0.96	0.0314	0.9946
4	0.0426	0.9892	0.0272	0.9946
5	0.034	0.9928	0.0235	0.9946
6	0.0144	0.997	0.0416	0.9946
7	0.0339	0.9934	0.0073	0.9946
8	0.0091	0.9982	0.0239	0.9946
9	0.0133	0.9952	0.0131	0.9946
10	0.0146	0.9958	0.0096	0.9946

Table 9. Generation_2 and Iterarion_4

Iteration:5	Epochs:20			
Epochs	Loss	Accuracy	Val_Loss	Val_accuracy
1	204.646	0.5406	4.4933	0.672
2	1.9394	0.6296	0.5835	0.6989
3	1.1191	0.6505	0.6362	0.6882
4	0.8923	0.69	0.4166	0.7903
5	0.7972	0.718	0.4619	0.8065
6	0.8555	0.7216	0.4893	0.8118
7	0.6609	0.7736	0.4137	0.8441
8	0.5675	0.7975	0.41	0.8333

9	0.5285	0.8136	0.3314	0.8495
10	0.4328	0.8351	0.2016	0.9247
11	0.4601	0.8399	0.1784	0.914
12	0.3757	0.8542	0.1098	0.9731
13	0.3473	0.8698	0.1368	0.957
14	0.3957	0.8632	0.3676	0.8387
15	0.4224	0.856	0.183	0.9194
16	0.3594	0.8656	0.1854	0.9086
17	0.3199	0.8787	0.1308	0.9086
18	0.3517	0.8662	0.2046	0.9032
19	0.3249	0.8775	0.1283	0.914
20	0.2079	0.9803	0.00E+00	1

Table 10. Generation_2 and Iterarion_5

Generation1												
Iteration No.	**Epochs**	**Loss**	**accuracy**	**Filter_1**	**Filter_2**	**Filter_3**	**Activation_F1**	**Activation_F2**	**Kernel**	**DO_Rate_1**	**DO_Rate_2**	**Optimizer**
1	13	0.0889	0.9476	32	64	256	relu	selu	3	0.5	0.4	adam
2	14	0.3938	0.8548	32	64	512	elu	relu	5	0.4	0.5	adadelta
3	19	0.0059	0.996	32	128	128	selu	selu	3	0.3	0.3	adadelta
4	15	0.8199	0.6371	64	128	256	elu	elu	5	025	0.5	adam
5	14	0.0146	0.996	32	128	256	selu	relu	3	0.4	0.3	

Table 11. Summary of performance of hyperparameters of generation_1

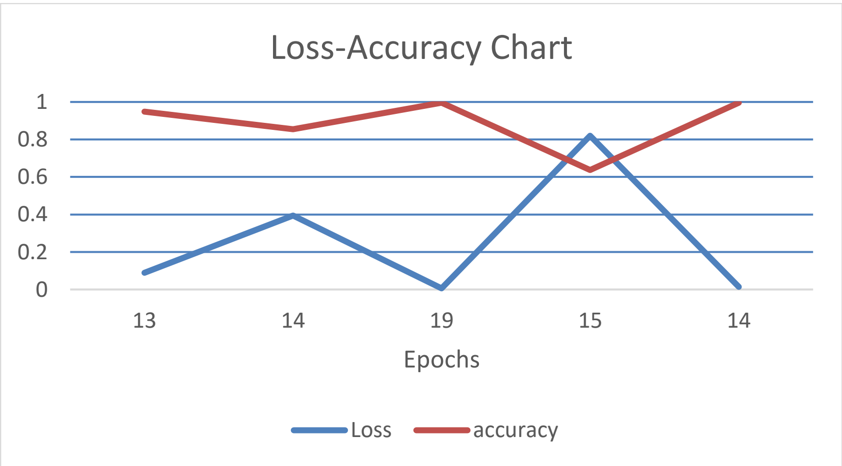

Fig. 5 Loss_Accuracy Plot generation number 1

Generation:2												
Iteration No.	Epochs	Loss	Accuracy	Filter_1	Filter_2	Filter_3	Activation_F1	Activation_F2	Kernel	DO_Rate1	DO_Rate2	Optimizer
1	13	0.3645	0.9153	32	64	256	relu	selu	3	0.5	0.4	adam
2	14	0.5592	0.7944	32	64	512	elu	relu	5	0.5	0.4	adadelta
3	19	0.0206	0.9919	32	128	128	selu	selu	3	0.3	0.3	adagrat
4	10	0.0065	1	64	128	256	elu	relu	3	0.2	0.4	adam
5	20	0.1281	0.9274	64	64	256	selu	elu	3	0.5	0.5	adam

Table 12. Summary of performance of hyperparameters of generation_2

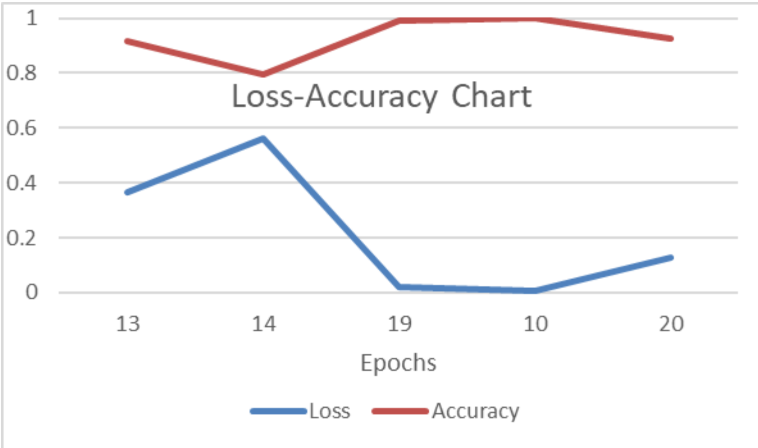

Fig. 6 Loss_Accuracy Plot generation number 2

The accuracy in the second generation is higher compared to the first generation. The diagram below illustrates the final architecture generated and trained by CNN. In comparison with the genetic algorithm, DECNN yields 20% more accuracy. The results of the comparison, utilizing four indicators for each algorithm, are presented in Table 13.

	CNN_withoaut SMOTE and Data augmentation	CNN_Smote and Data Augmentation	CNN_GA	DE_CNN
Accuracy	0.55	0.95	0.7913	0.997
Loss	3620.9067	0.23	0.8454	0.0275
Val_Accuracy	0.5156	0.9627	0.5853	1
Val_Loss	9230.1768	0.12	1.2069	1.46E-04

Table.13 Comparison of Proposed model with another model using same dataset.

Our proposed strategy is also compared with several state-of-the-art approaches. Evolutionary CNN [20] was evaluated on MNIST-FASHION and CIFAR10 datasets. ECNN not only accelerates the execution process but also achieves higher classification accuracy. Researchers proposed Evolutionary CNN using GA on the breast cancer dataset, where GA with the ADAM optimizer resulted in an 85% accuracy [21]. D. Elhani et al. (2023) demonstrate effortless and promising results for image classification using particle swarm optimization [22]. For digital classification on the SVHM dataset, the authors proposed Optimal CNN_GA and achieved a validation accuracy of 92.31% [23].

5. Conclusion

For the deadly disease, lung cancer, early diagnosis can be a lifesaving step. We have endeavored to create an automated model to save time and lives. The proposed DE-CNN algorithm aims to optimize the hyperparameters of the 2D CNN. In the first generation, the automated CNN selects filters as 64, 64, 126, kernel_size 3, activation functions 'elu' and 'selu', dropouts 0.4 and 0.4, optimizers 'admax', and the number of epochs 12. Similarly, in each iteration, the automated CNN generates an optimized architecture automatically. When training the DE_CNN model, it achieves better performance compared to the Basic CNN model, which used manual parameter settings. We evaluate the model with Classic CNN and Genetic algorithm. The simulation results indicate effective and reliable detection of lung cancer. The table above shows that the accuracy increases by 40%, average precision by 15%, recall by 16%, and F1 score by 25%.

Funding declaration

I did not receive financial assistance for the research, authorship, and/or publication of this article.

Competing Interest declaration

The authors declare no competing interests. The co-author has reviewed and approved the manuscript's contents, and there are no financial interests to disclose.

Author Contribution

The author affirms full responsibility for the study's conceptualization and design, data collection, analysis, result interpretation, and the preparation of the paper.

Data Availability Statement

The dataset for implementing the proposed model and conducting result analysis is sourced from Kaggle.

Research Involving Human Participants and/or Animals

The university and my guide affiliated with Kalinga University, Chhattisgarh, have approved this study and its concept.

Informed Consent

No surveys or consent were conducted for this research article. The data utilized for implementing the proposed model is sourced from the Kaggle site.

Acknowledgement

There are no funds or grants available for this article from any organization.

References

[1] D'souza, R.N., Huang, PY. & Yeh, FC (2020) Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size. *Sci Rep* **10**, 834. https://doi.org/10.1038/s41598-020-57866-2

[2] S Balambigai et al 2022 J.Phys(2022)Conf.Ser.2318012040.Detection and optimization of skin cancer using deep learning. Journal of Physics: Conference Series, Volume 2318, 8th International Virtual Conference on Biosignals, Images, and Instrumentation (ICBSII 2022) 16/03/2022 - 18/03/2022 Online

[3] R. Pandian, V. Vedanarayanan, D.N.S. Ravi Kumar, R. Rajakumar(2022) Detection and classification of lung cancer using CNN and Google net, Measurement: Sensors,Volume 24,2022,100588,ISSN 2665-9174,https://doi.org/10.1016/j.measen.2022.100588. (https://www.sciencedirect.com/science/article/pii/S2665917422002227)

[4] Vikul J. Pawar,Kailash D. Kharat,Suraj R. Pardeshi, Prashant D. Pathak(2020) Lung Cancer Detection System Using Image Processing and Machine Learning Techniques, International Journal of Advanced Trends in Computer Science and Engineering, ISSN:2278-3091,Volume:9,IssueNo.4,July–August 2020,http://www.warse.org/IJATCSE/static/pdf/file/ijatcse260942020.pdf.

[5] Dakhaz Mustafa Abdullah & Nawzat Sadiq Ahmed (2021)A Review of most Recent Lung Cancer Detection Techniques using Machine Learning,International Journal of Science and Business,Volume: 5, Issue: 3 Page: 159-173,ijsab.com/ijsb

[6] Ibrahim M. Nasser, Samy S. Abu-Naser(2019) Lung Cancer Detection Using Artificial Neural Network,International Journal of Engineering and Information Systems (IJEAIS) ISSN: 2000-000X Vol. 3 Issue 3, March – 2019, Pages: 17-23

[7] Venkatesh C, Ramana K, Lakkisetty SY, Band SS, Agarwal S, Mosavi A.(2022) A Neural Network and Optimization Based Lung Cancer Detection System in CT Images. Front Public Health. 2022 Jun 7;10:769692. doi: 10.3389/fpubh.2022.769692. PMID: 35747775; PMCID: PMC9210805.

[8] Hussain Ali, Y.; Sabu Chooralil, V.; Balasubramanian, K.; Manyam, R.R.; Kidambi Raju, S.; T. Sadiq, A.; Farhan, A.K(2023) Optimization System Based on Convolutional Neural Network and Internet of Medical Things for Early Diagnosis of Lung Cancer. Bioengineering 2023, 10, 320. https://doi.org/10.3390/ bioengineering10030320

[9] Pfeffer, M.A.; Ling, S.H(2022) Evolving Optimised Convolutional Neural Networks for Lung Cancer Classification. Signals 2022, 3, 284–295. https://doi.org/10.3390/ signals3020018.

[10] P.Sinthia,M.Malathi(2020) Cancer detection using convolutional neural network optimized by multistrategy artificial electric field algorithm,international journal of Imaging system and Technology,19 december 2020, https://doi.org/10.1002/ima.22530

[11] Xinrong Lu , Y. A. Nanehkaran , and Maryam Karimi Fard,(2021) A Method for Optimal Detection of Lung Cancer Based on Deep Learning Optimized by Marine Predators Algorithm Hindawi,Computational Intelligence and Neuroscience,Volume 2021, Article ID 3694723, 10 pages,https://doi.org/10.1155/2021/3694723

[12] Cheng-Jian Lin * , Shiou-Yun Jeng and Mei-Kuei Chen (2020) Using 2D CNN with Taguchi Parametric Optimization for Lung Cancer Recognition from CT Images,Appl. Sci. 2020, 10, 2591; doi:10.3390/app10072591 www.mdpi.com/journal/applsci.

[13] Sebastian, A.E., Dua, D(2023) Lung Nodule Detection via Optimized Convolutional Neural Network: Impact of Improved Moth Flame Algorithm. Sens Imaging 24, 11 (2023). https://doi.org/10.1007/s11220-022-00406-1

[14] Mohamad Faiz Ahmad, Nor Ashidi Mat Isa, Wei Hong Lim, Koon Meng Ang (2022),Differential evolution: A recent review based on state-of-the-art works,Alexandria Engineering Journal, Volume 61, Issue 5,2022,Pages 3831-3872,ISSN 110-0168,https://doi.org/10.1016/j.aej.2021.09.013.(https://www.sciencedirect.com/science/article/pii/S111001682100613X)

[15] Buradkar, V.S., Ambhaikar, A. (2023). Applications in Medical Technology for Optimized Convolutional Neural Network Using Differential Evolutionary Algorithm. In: Sarkar, D.K., Sadhu, P.K., Bhunia, S., Samanta, J., Paul, S. (eds) Proceedings of the 4th International Conference on Communication, Devices and Computing. ICCDC 2023. Lecture Notes in Electrical Engineering, vol 1046. Springer, Singapore. https://doi.org/10.1007/978-981-99-2710-4_26

[16] Sampathila N, Chadaga K, Goswami N, Chadaga RP, Pandya M, Prabhu S, Bairy MG, Katta SS, Bhat D, Upadya SP)2022) Customized Deep Learning Classifier for Detection of Acute Lymphoblastic Leukemia Using Blood Smear Images,*Healthcare*. 2022; 10(10):1812. https://doi.org/10.3390/healthcare10101812

[17] Balambigai S1, Elavarasi K1, Abarna M1, Abinaya R1 and Arun Vignesh N2(2022)Detection and optimization of skin cancer using deep learning,Journal of Physics: Conference Series 2318 (2022) 012040 IOP Publishing doi:10.1088/1742-6596/2318/1/012040

[18] Md Rafiul Hassan,Walaa N. Ismail,Ahmad Chowdhury,Sharara Hossain,Shamsul Huda,Mohammad Mehedi Hassan,(2021)A framework of genetic algorithm-based CNN on multi-access edge computing for automated detection of COVID-19,The Journal of Supercomputing (2021) 78:10250–10274 https://doi.org/10.1007/s11227-021-04222-4

[19] Abdelrahman Ezzeldin Nagib, Mostafa Mohamed Saeed,Shereen Fathy El-Feky, and Ali Khater Mohamed,(2022) Hyperparameters Optimization of Deep Convolutional Neural Network for Detecting COVID-19 Using Differential Evolution,Decision Sciences for COVID-19 (pp.305-325),February 2022 DOI:10.1007/978-3-030-87019-5_18.

[20] Wang, Zhao, Di Lu, Huabing Wang, Tongfei Liu, and Peng Li.(2021). "Evolutionary Convolutional Neural Network Optimization with Cross-Tasks Transfer Strategy" *Electronics* 10, no. 15: 1857. https://doi.org/10.3390/electronics10151857

[21] Davoudi K, Thulasiraman P.(2021) Evolving convolutional neural network parameters through the genetic algorithm for the breast cancer classification problem. Simulation.;97(8):511-527. doi: 10.1177/0037549721996031. Epub 2021 Mar 5. PMID: 34366489; PMCID: PMC8293734.

[22] D. Elhani, A.C. Megherbi, A. Zitouni, F. Dornaika, S. Sbaa, A. Taleb-Ahmed, Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification, Expert Systems with Applications,Volume 229, Part A,2023,120411,ISSN0957-4174,https://doi.org/10.1016/j.eswa.2023.120411. (https://www.sciencedirect.com/science/article/pii/S0957417423009132)

[23] A. Al-Hyari and M. Abu-Faraj, "Hyperparameters Optimization of Convolutional Neural Networks using Evolutionary Algorithms," *2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA)*, Karak, Jordan, 2022, pp. 1-6, doi: 10.1109/ETCEA57049.2022.10009778.