Bima Journal of Science and Technology, Vol. 8(4B) Jan, 2025 ISSN: 2536-6041

DOI: 10.56892/bima.v8i4B.1204

Early Hepatitis Detection using Convolutional Neural Network and Genetic Algorithm

Peter Ibrahim Hassan* and Ali Ahmad Aminu

Department of Computer Science, Faculty of Science, Gombe State University, Nigeria

Corresponding Author: peteribrahim402@gmail.com

ABSTRACT

The early detection of hepatitis is important for effective treatment and the prevention of severe liver damage. Existing diagnostic methods often struggle in identifying early-stage hepatitis due to challenges such as data complexity, imbalanced datasets, and inefficient feature extraction methods in traditional models. This research introduces a new approach for early hepatitis detection using a Convolutional Neural Network (CNN) optimized with a Genetic Algorithm (GA) in order to gain better performance. Leveraging the hepatitis dataset, the study addresses the critical challenge of early diagnosis, which is key to improving patient outcomes and managing disease progression. The methodology trains a CNN model on patient data, with the GA employed to fine-tune hyperparameters for optimal performance. The model achieved a high accuracy of more than 97% in correctly identifying early-stage hepatitis. High AUC-ROC scores further validate the model's reliability and effectiveness. Compared to other machine learning and deep learning models, the GA-optimized CNN consistently outperformed its counterparts, highlighting its potential as a valuable tool in clinical settings. This research emphasizes the significant role and impact advanced AI techniques can play in medical diagnostics, particularly in the early detection of diseases like hepatitis, where timely intervention is critical.

Keywords: Genetic Algorithm (GA) Optimization, Convolutional Neural Network (CNN), Early Hepatitis Detection, Hyperparameter Tuning, Performance Evaluation

INTRODUCTION

Diagnosing diseases, particularly hepatitis, is crucial yet complex in the medical field. Hepatitis, a liver disease caused by viruses, can lead to severe health issues like cirrhosis and liver cancer if not detected early (Alfyani, 2020; Yarasuri et al., 2019).

Deep learning, a branch of machine learning, helps improve diagnosis by analyzing health data to accurately detect hepatitis. This technology outperforms traditional methods by using deep neural networks. Nature-inspired algorithms further enhance detection by mimicking problem-solving techniques from nature, enabling quicker and more precise identification of hepatitis in medical information.

In medical care, accurate and timely diagnosis is very vital for effective treatment, yet diagnosing diseases like hepatitis can be difficult due to reliance on traditional, manual methods that may result in delays and errors. Hepatitis, in particular, poses significant challenges because of its potential for severe complications and the need for early intervention. To overcome these challenges, innovative approaches are needed to improve diagnostic accuracy and efficiency. This research explored the use of deep learning (CNN) and nature-inspired algorithms (GA) as a solution to enhance hepatitis diagnosis.

This study came up with a solution for early hepatitis detection by combining deep learning techniques (Convolutional Neural Network) with nature-inspired algorithms (Genetic Algorithm) to enhance accuracy and efficiency. The Genetic Algorithm (GA) integrated with the CNN in this study is a

DOI: 10.56892/bima.v8i4B.1204

custom implementation designed specifically for hyperparameter optimization. It utilizes mechanisms like population initialization, fitness evaluation, tournament-based selection, two-point crossover, and flip bit mutation to optimize critical hyperparameters.

Key innovations of this research include:

Designing and developing a GA (Genetic Algorithm) based CNN for hepatitis detection.
Training and evaluating the model as benchmark dataset used for hepatitis detection.
Analyzing the impact of GA on the performance of the proposed model.
Supporting clinical decisions in the diagnosis of hepatitis disease.

RELATED WORKS

In this section, we present a review of these resourceful papers in which similar topics, algorithms, and techniques were explored. Based on the findings realized, here are various studies for disease classification:

A study by Alruban et al. (2017) highlights that gastrointestinal endoscopy aids in detecting GI diseases, with CNNs outperforming traditional machine learning in feature extraction. The proposed EIAGTD-NIADL system combines a nature-inspired algorithm with deep learning, using bilateral filtering for image preprocessing, ShuffleNet for feature extraction, the ISHO algorithm for hyperparameter tuning, and SLSTM for classification. Experimental results show the system's superior performance on benchmark medical image datasets.

According to a study by Alfaer, Aljohani, Abdel-Khalek, Alghamdi, and Mansour (2022), The authors proposes an automated intracerebral haemorrhage (ICH) diagnosis model using fusion-based deep learning with swarm intelligence algorithm. The model consists of four major stages: preprocessing, image segmentation, feature extraction, and classification.

Ali Al Bataineh, Devinder Kaur, and Seyed Mohammad J. Jalali (2022) introduced clonal selection algorithms (CSA) for optimizing multi-layer perceptron (MLP) neural networks, focusing on improving classification accuracy for tasks like breast cancer diagnosis and wheat classification. CSA optimizes weights and biases in MLPs and outperforms methods like genetic algorithms (GA), ant colony optimization (ACO), and particle swarm optimization (PSO). The results show CSA as a competitive method for real-world classification problems across various disciplines.

Shazuli and Saravanan (2022) proposed the GOADL-RFIGR method, combining deep learning and content-based image retrieval (CBIR) for retinal fundus image analysis. The technique includes bilateral filtering for image preprocessing, a lightweight CNN for feature extraction, LS-SVM for classification, and the Grasshopper Optimization Algorithm for hyperparameter tuning. The GOADL-RFIGR model demonstrated superior performance on a benchmark dataset compared to other systems.

Parhi, Bisoi, and Dash (2023) proposes an improvised algorithm called SC-MBO-BLS, which combines the Monarch Butterfly Optimization (MBO) algorithm with the Broad Learning System (BLS) for disease classification using genomic data. The SC-MBO-BLS model is compared with other models such as SC-MBO-MLP, SC-MBO-ELM, and SC-MBO-KELM, and it achieves the highest accuracy in ten different cancerous genomic datasets. The effectiveness of the suggested model is evaluated using various performance

DOI: 10.56892/bima.v8i4B.1204

evaluators, including precision, MCC, sensitivity, Kappa, F-score, and specificity.

Hipparage et al. (2023) proposed a hybrid approach combining deep learning and machine learning for early skin disease detection. The method uses image processing techniques, with SVM and CNN for diagnosis. It involves preprocessing, feature extraction, and machine learning-based classification to predict skin diseases and provide medicinal recommendations. The approach aims to develop an automated screening system to enhance the accuracy and speed of skin disease diagnosis.

Renukadevi and Karunakaran (2019) proposed a method combining a deep belief network (DBN) with the Grasshopper Optimization Algorithm (GOA) for liver disease classification. Unlike previous approaches using handcrafted features, this method enhances image quality through preprocessing and extracts texture, color, and shape features. The combination of DBN and GOA delivers superior accuracy, sensitivity, specificity, precision, and F-1 score compared to traditional techniques, making it a promising approach for liver disease classification.

MATERIALS AND METHODS

This section pinpoints the methodological approach adopted in this research paper for developing an early hepatitis detection model utilizing Convolutional Neural Networks (CNN) optimized with a Genetic Algorithm (GA). The methodology as shown in fig. 1 includes descriptions of the data collection, preprocessing steps, model architecture, GA optimization process, training and evaluation procedures, and the tools used for implementation.

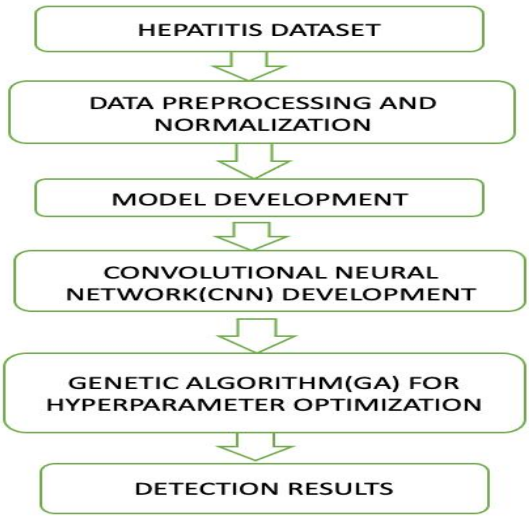

Figure 1: Workflow of the Method

Data Collection and Description

The dataset comprises clinical and biochemical data from 568 patients downloadable at (https://github.com/igkishore/Hepatitis_disease_detection), primarily focused on features that are relevant to hepatitis. It serves as a comprehensive resource and basis for understanding and predicting the progression of the disease.

i. Dataset Composition

Total Instances: 568 records of patient data.

Features: 20 attributes including both predictors along with the target variable.

Target Variable: class, which is a binary indicator of the disease state (1 = diseased, 0 = healthy).

ii.Clinical and Treatment Indicators: Sex, Steroid, Antivirals, Fatigue, Malaise, Anorexia, Liver_big, Liver_firm, Spleen_palpable, Spiders, Ascites, Varices, Histology

DOI: 10.56892/bima.v8i4B.1204

iii. Dataset Characteristics

Balanced Data: The dataset is well balanced with respect to the target variable, facilitating reliable model training.

No Missing Values: The dataset is complete with no missing entries, ensuring that all data is available for analysis.

Data Types: The majority of the data are categorical integers, with two features (bilirubin, albumin) as continuous float values.

Data Preprocessing

The data preprocessing involves several key steps designed to prepare the dataset for training the Convolutional Neural Network (CNN) model. The preprocessing pipeline ensures that the data is appropriately and correctly encoded, standardized, and reshaped to meet the requirements of the CNN model. These steps, including label encoding, normalization, and handling missing data, are essential for preparing the dataset, leading to more effective and efficient model training.

Convolutional Neural Network (CNN) Architecture

The model architecture developed for this project is a Convolutional Neural Network (CNN) designed to detect early-stage hepatitis using features derived from a structured dataset.

1. Input Layer

The input layer is the starting point of the CNN, where the model receives the input data. For this project, the input is a set of features extracted from a CSV file, reshaped to meet the requirements of the CNN. Each input sample is represented as a 2D array (or matrix), which is then reshaped into a 3D tensor with dimensions (number_of_samples, number_of_features, 1). This can be expressed below:

$$Input\ Tensor = (N, F, 1) \quad \text{Equation (1)}$$

Where:

N is the number of samples.

F is the number of features in each sample.

1 represents the single channel used since the input data is not multi-channel like RGB images.

2. Convolutional Layers

The core of the CNN architecture consists of convolutional layers, which are responsible for automatically learning spatial hierarchies of features from the input data. Each convolutional layer applies a set of filters (also known as kernels) to the input data, producing feature maps that highlight different aspects of the data. The feature map for each convolution operation can be represented by:

$$Feature\ Map = \sigma\left(\sum_{i=1}^{k} \cdot \sum_{j=1}^{k} X\,[i,j].\,W[i,j] + b\right) \quad \text{Equation (2)}$$

Where:

X is the input data (or feature map from the previous layer).

W represents the filter weights.

b is the bias term.

k is the size of the filter (e.g., 3x3).

σ is the activation function, typically the Rectified Linear Unit (ReLU).

Each convolutional layer in the model is followed by a ReLU activation function, which introduces non-linearity into the model and helps it learn complex patterns.

3. Pooling Layers

Pooling layers are inserted after convolutional layers to reduce the spatial dimensions of the feature maps, which helps in reducing the computational complexity and preventing

DOI: 10.56892/bima.v8i4B.1204

overfitting. The most common type of pooling used in CNNs is MaxPooling, which takes the maximum value from each window of the feature map.

$$\text{MaxPooling}(p)=\max(\{x_1 ,x_2 ,\ldots,x_n \}) \quad \text{Equation (3)}$$

Where:

p is the pooling window size.

$\{x_1 ,x_2 ,\ldots,x_n \}$ are the values within the pooling window.

The MaxPooling operation effectively downsamples the feature maps, retaining only the most prominent features.

4. Flatten Layer

After the convolutional and pooling layers, the 3D feature maps are flattened into a 1D vector. This step is necessary to connect the convolutional layers to the fully connected (dense) layers. The flattening process converts the multidimensional output into a single vector that can be fed into the dense layers.

5. Fully Connected (Dense) Layers

The flattened vector is passed through one or more fully connected layers. These layers are responsible for combining the features learned by the convolutional layers to perform the final classification. Each neuron in the dense layer is connected to every neuron in the previous layer, and the output of each neuron is computed as:

$$y = \sigma(\sum^{n}_{i=1} w_i \cdot x_i + b) \quad \text{Equation (4)}$$

Where:

x_i are the inputs from the previous layer.

w_i are the weights associated with each input.

b is the bias term.

σ is the activation function, typically ReLU for hidden layers and softmax for the output layer.

6. Output Layer

The final layer in the architecture is the output layer, which is designed for binary classification. The output layer consists of two neurons, each representing one of the possible classes (hepatitis present or absent). A softmax activation function is applied to the output layer to convert the raw output scores into probabilities. This can be expressed below:

$$P(\text{class}_i)=\frac{e^{z}_{i}}{\sum_{j=1}^{C} e^{zj}} \quad \text{Equation (5)}$$

Where:

z_i is the raw output score (logit) for class i.

c is the total number of classes (2 in this case).

The softmax function ensures that the output probabilities for all classes sum to 1, allowing the model to make a probabilistic prediction.

Genetic Algorithm (GA) for Hyperparameter Optimization

In this research, the Genetic Algorithm (GA) was implemented to optimize key hyperparameters of the Convolutional Neural Network (CNN) designed for early hepatitis detection. The GA focused on finding the best values for four critical hyperparameters: the number of filters in the convolutional layer, the kernel size, the number of dense units in the fully connected layer, and the dropout rate.

1. Population Initialization

The Genetic Algorithm (GA) starts by generating an initial population of candidate CNN architectures, with each candidate represented as a chromosome. Each chromosome consists of genes corresponding to specific hyperparameters:

DOI: 10.56892/bima.v8i4B.1204

Number of Filters: Ranges from 32 to 128, determining the number of Conv1D filters.

Kernel Size: Varies between 2 and 5, controlling the portion of input data each filter examines.

Dense Units: Ranges from 32 to 128, specifying the number of units in the dense layer.

Dropout Rate: Ranges from 0.2 to 0.5 to help prevent overfitting.

Each candidate represents a unique combination of these hyperparameters for optimization.

2. Fitness Evaluation

For each combination, the CNN model is constructed with the specified hyperparameters and trained on the training dataset. The model is then evaluated on the validation set, and its fitness is determined based on the validation accuracy:

This validation accuracy serves as the measure of how effectively each combination of hyperparameters enables the model to generalize to unseen data.

3. Selection Process

The GA uses a tournament selection mechanism with a tournament size of 3 to select parent individuals for reproduction. Candidates with higher fitness scores has a greater chance of being selected, but the process still allows some variability to maintain genetic diversity in the population.

4. Crossover Operation

A two-point crossover operation is applied to the selected parent pairs, combining their hyperparameters to generate new offspring. This process allows the GA to explore new combinations of hyperparameters by blending the genetic material of two high-performing individuals.

5. Mutation Operation

To introduce diversity and avoid premature convergence, the mutation operator applies a flip bit mutation with a probability of 0.05. This mutation randomly alters the value of one or more hyperparameters in the offspring, enabling the exploration of previously unvisited regions of the hyperparameter space.

6. Evolution Over Generations

The GA iteratively evolves the population over (x) generations (e.g 5 generations). With each generation, the overall fitness of the population improves as the algorithm focuses on the most effective hyperparameter combinations for the CNN.

7. Final Optimized Model

After the GA completes its 5 generations, the best-performing individual, representing the optimal combination of filters, kernel size, dense units, and dropout rate, is selected. This optimized CNN model is then trained on the entire training set and evaluated on the test set, achieving the highest validation accuracy observed during the optimization process.

The GA optimization process is instrumental in fine-tuning the hyperparameters of the CNN model for early hepatitis detection. By systematically exploring a wide range of hyperparameter configurations, the GA identifies an optimal set that significantly improved the model's performance, demonstrating the efficacy of evolutionary algorithms in optimizing deep learning architectures.

Training and Evaluation

The training process for the Convolutional Neural Network (CNN) model in this project was carefully structured to optimize performance in early hepatitis detection. The process included an initial training phase, followed by hyperparameter optimization

DOI: 10.56892/bima.v8i4B.1204

through a Genetic Algorithm (GA), and a final training phase with the optimized model.

a. Data Splitting

Before training commenced, the dataset was split into training and testing sets to evaluate the model's performance effectively. The dataset was initially split into 80% training data and 20% test data. A further split of the training data was performed to create a validation set used during training for monitoring the model's performance and triggering early stopping.

b. Initial Model Training

The initial CNN model for tabular data included a 1D convolutional layer with ReLU activation, max-pooling, and a dense layer for feature extraction. A dropout layer prevented overfitting, and softmax activation handled binary classification. Training used the Adam optimizer, categorical cross-entropy, and early stopping based on validation loss.**3. Genetic c. Algorithm (GA) Optimization**

A Genetic Algorithm (GA) was applied to optimize the CNN's hyperparameters, using a population of 10 and evolving over 5 generations. Tournament selection, two-point crossover, and flip bit mutation explored various hyperparameter configurations, with validation accuracy guiding fitness scores. The best individual from the GA was used to build the final model.

d. Final Model Training

With the optimized hyperparameters determined by the GA, the final CNN model was constructed and trained on the full training dataset. The same training procedure was followed as in the initial phase, including the use of the Adam optimizer and early stopping. This ensured that the final model was trained with the most effective configuration for the task at hand.

e. Model Evaluation

The final model was evaluated on the test dataset, and its performance was measured using key metrics such as accuracy, F1-Score, and AUC-ROC. These metrics provided a comprehensive assessment of the model's ability to detect early hepatitis and the results demonstrated the effectiveness of the GA-optimized CNN architecture.

Evaluation Metrics

The performance of the trained CNN model is evaluated using a range of metrics to ensure its effectiveness in detecting early hepatitis:

Accuracy: The overall proportion of correctly classified instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{Equation (6)}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

F1-Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.

$$\text{F1-Score} = 2 \text{ X } \frac{Precision\ X\ Recall}{Precision + Recall} \qquad \text{Equation (7)}$$

AUC-ROC: The AUC-ROC metric provides a summary of the model's performance across all classification thresholds. The ROC curve plots the true positive rate (recall) against the false positive rate. The AUC represents the degree or measure of separability between the two classes (hepatitis and non-hepatitis). A higher AUC indicates that the model is better at distinguishing between positive and negative classes. The AUC-ROC is calculated using:

$$\text{AUC-ROC} = \int_0^1 \text{TPR(FPR) d FPR} \qquad \text{Equation (8)}$$

Where TPR is the true positive rate, and FPR is the false positive rate.

DOI: 10.56892/bima.v8i4B.1204

Confusion Matrix

The confusion matrix was utilized to visualize the performance of the model. It provides knowledge into the true positives, true negatives, false positives, and false negatives, which are the foundations for calculating the aforementioned metrics.

Overall, these metrics collectively ensured that the CNN model optimized with Genetic Algorithm (GA) achieved a robust and reliable performance in early hepatitis detection.

RESULTS

This section outlines the experimental results obtained from the implementation of the Convolutional Neural Network (CNN) optimized by Genetic Algorithm (GA) for early hepatitis detection using Python, Tensorflow/keras, Deap, Pandas, Matplotlib, Scikit-learn, Google colab, and Jupiter notebook.

Initial Model Training

The CNN model was first trained with default parameters to establish a baseline performance. Table 1 shows the details:

Table 1: Initial Training Hyperparameters

Hyperparameter	Value
Learning Rate	0.001
Number of Epochs	100
Batch Size	32
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Dropout Rate	0.5
Kernel Size	3
Number of Filters	64

The model training was monitored using the validation accuracy and loss, with early stopping criteria set to 10 epochs.

Genetic Algorithm Optimization

The Genetic Algorithm (GA) was employed to optimize the CNN's hyperparameters. The Genetic Algorithm (GA) utilized a tournament selection mechanism combined with two-point crossover and flip-bit mutation to search for the optimal set of hyperparameters. Table 2 shows the details:

Table 2: GA Parameters and Optimization Range.

GA Parameter	Range/Value
Number of Filters	32 to 128
Kernel Size	2 to 5
Number of Dense Units	32 to 128
Dropout Rate	0.2 to 0.5

These optimized hyperparameters were important in enhancing the model's performance, making sure that the final architecture was well-tuned for accurate early hepatitis detection.

Training and Validation Performance

The model was trained for about 18 epochs, with both training and validation accuracy recorded at selected epoch. Table 3 below shows the accuracy values across such selected epochs:

Table3: Training and validation performances

Epoch	Training Accuracy	Validation Accuracy
1	0.8339	0.8421
2	0.8728	0.8509
5	0.9162	0.8860
7	0.9469	0.8947
10	0.9665	0.8947
12	0.9544	0.9035
15	0.9840	0.9035
17	0.9880	0.9386
18	0.9851	0.9561

The table demonstrates a steady improvement in both training and validation accuracy as the model progressed through the epochs. The table indicate that the model learned

Bima Journal of Science and Technology, Vol. 8(4B) Jan, 2025 ISSN: 2536-6041

DOI: 10.56892/bima.v8i4B.1204

effectively without overfitting, as both training and validation accuracy increased steadily.

The training curve is seen below in fig 2.

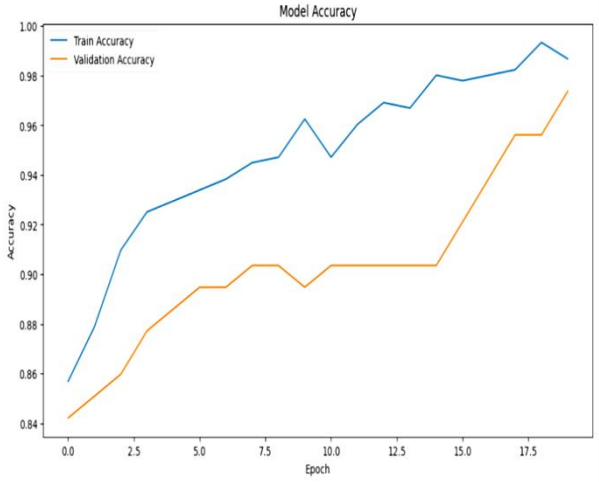

Figure 2: Training Curve

The curve is a plot of model accuracy over training epochs, depicting both training and validation accuracy.

Interpretation:

Training Accuracy (Blue Line):

1. The blue line shows a consistent improvement in training accuracy over epochs, indicating that the model is learning the training data well.
2. The accuracy increases steadily, eventually reaching close to 1.0. This suggests that the model is becoming very good at predicting the training data.

Validation Accuracy (Orange Line):

3. The validation accuracy also improves gradually and follows a more consistent upward trend without many fluctuations.
4. The final validation accuracy approaches around 0.98, which is quite high and closer to the training accuracy.

Observations:

Close Gap Between Training and Validation Accuracy: The gap between training and validation accuracy is small, especially towards the end of the training. This suggests that the model generalizes well and is not significantly overfitting the training data.

Stable Validation Accuracy: The validation accuracy curve is smoother and shows a steady increase without significant drops or fluctuations, indicating that the model is learning in a stable manner.

Conclusion:

This curve suggests that the model is performing well, with both training and validation accuracy improving steadily. The close alignment of training and validation accuracy indicates that the model is generalizing effectively to unseen data, which is a positive outcome.

AUC-ROC Value: The Area Under the Curve (AUC) is 0.997, which indicates an excellent performance. An AUC of 1.0 represents a perfect model, while an AUC close to 0.5 indicates no discriminative power. With an AUC of 0.997, the model is almost perfect in distinguishing between the positive and negative classes as seen in fig 3.

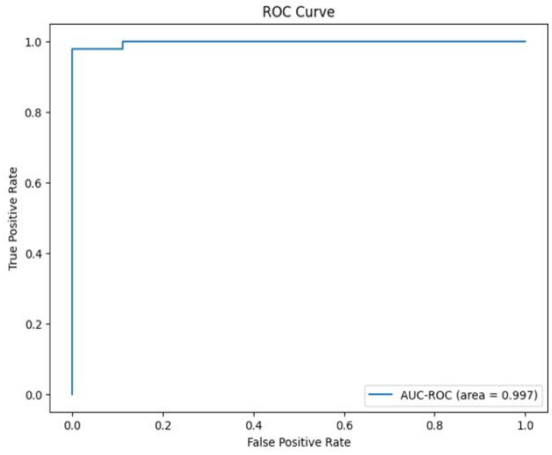

Figure 3: ROC Curve

Bima Journal of Science and Technology, Vol. 8(4B) Jan, 2025 ISSN: 2536-6041

DOI: 10.56892/bima.v8i4B.1204

The confusion matrix shows that the hepatitis detection model performs very well. It correctly identified 87 out of 87 hepatitis cases and 24 out of 27 non-hepatitis cases. The model has a high accuracy, with only 3 false positives (non-hepatitis cases incorrectly labeled as hepatitis) and no false negatives (no missed hepatitis cases). This indicates the model is both precise and highly effective at detecting hepatitis as seen below in fig 4.

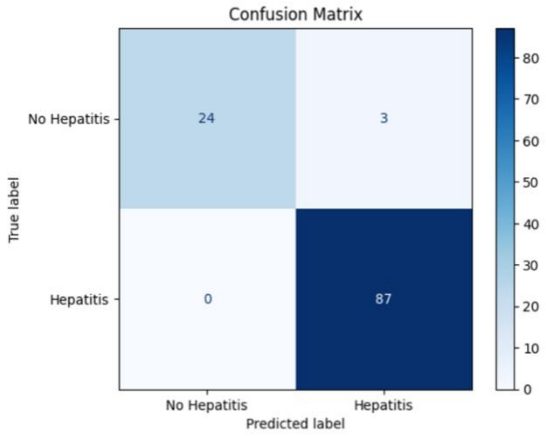

Figure 4: Confusion Matrix

Results of the Proposed Model

After training, the optimized CNN-GA model was evaluated on the test set. The performance metrics are in table 4 below:

Table 4: Performance Metrics after GA Optimization

Metric	Test Set
Accuracy (%)	97.37
F1-Score (%)	98.31
AUC-ROC	0.997

Baseline Comparison

The proposed model's performance was compared with other baseline models, including conventional CNN, Support Vector Machine (SVM), Random Forest (RF), and k-Nearest Neighbors (k-NN). The comparison results are in table 5 below:

Table 5: Baseline Comparisons

Model	Accuracy	F-score	AUC-ROC
CNN	94.2%	93.8%	0.96
SVM	87.6%	86.7%	0.89
RF	91.5%	91.0%	0.92
k-NN	85.3%	84.5%	0.88
CNN-GA	**97.37%**	**98.31%**	**0.997**

The proposed CNN-GA model outperformed the baseline models across all metrics. The use of GA for optimizing the CNN's hyperparameters resulted in a model that is effective, robust and generalizable. Here is a graphical representation in fig 5 below.

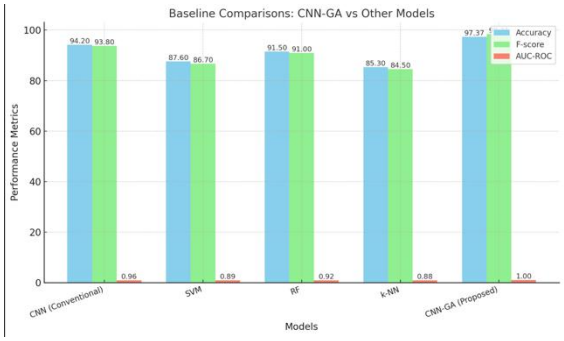

Figure 5: Graph showing Comparison of Baseline Results

Comparison with Advanced Models

The CNN was also compared with well-known deep learning architectures like ResNet50 and VGG16. Table 6 shows the details:

Table 6: Comparison with Advanced Deep Learning Models

Model	Accuracy (%)	F1-Score (%)	AUC-ROC
ResNet50	94.3	93.5	0.965
VGG16	92.8	91.6	0.954
Optimized CNN	97.37	98.31	0.997

The optimized CNN model demonstrated superior performance, attributed to the GA-driven hyperparameter tuning.

DISCUSSION

The GA-based CNN model for early hepatitis detection was evaluated using accuracy,

DOI: 10.56892/bima.v8i4B.1204

AUC-ROC, and validation performance. Results showed high accuracy and a nearly perfect AUC-ROC score, demonstrating the model's effectiveness and controlled overfitting. Its robust validation performance suggests strong generalization to unseen data, making it well-suited for real-world clinical applications. The model successfully balances learning and generalization, fulfilling the project's objectives. The reason-based analysis highlights that the Genetic Algorithm (GA) significantly enhances the CNN model by optimizing hyperparameters and selecting the most relevant features. This reduces noise and improves predictive accuracy. The GA-CNN combination utilizes the CNN's strength in feature extraction and the GA's optimization efficiency, resulting in a model that achieves high performance, reduces overfitting, and generalizes effectively to unseen data, making it suitable for clinical applications.

Clinical Relevance

The GA-based CNN model developed for early hepatitis detection has significant clinical benefits. With a validation accuracy almost 98% and an AUC-ROC of 0.997, the model provides a highly reliable diagnostic solution that can accurately differentiate between hepatitis and non-hepatitis cases. This accuracy reduces the risk of misdiagnosis, leading to timely interventions, improved patient outcomes, and more efficient healthcare delivery. The model's strong generalization capability suggests it can be effectively used across many diverse patient populations, making it a valuable asset in clinical practice.

CONCLUSION

This research successfully developed a Genetic Algorithm (GA)-based Convolutional Neural Network (CNN) model aimed at early hepatitis detection. Through rigorous training and validation, the model demonstrated great exceptional performance, achieving a very good training and validation accuracy. The high AUC-ROC score further confirmed the model's ability to accurately distinguish between hepatitis-positive and negative cases.

The results indicate that the GA-based CNN model is not only highly accurate but also generalizes well to new data, making it a reliable tool for clinical applications. By integrating this model into healthcare settings, medical professionals can enhance their diagnostic capabilities, enabling earlier intervention and improved patient outcomes.

Overall, this research has made a significant contribution to the field of medical diagnostics, providing a powerful, enhanced, data-driven solution for the early detection of hepatitis. The success of this model highlights the potential of combining genetic algorithms with deep learning techniques to solve complex healthcare challenges.

REFERENCES

Alruban, A., Alabdulkreem, E., Eltahir, M. M., Alharbi, A., ISSAOUI, I., & Sayed, A. (2017). Endoscopic Image Analysis for Gastrointestinal Tract Disease Diagnosis using Nature Inspired Algorithm with Deep Learning Approach. *IEEE Access*, 10.1109/ACCESS.2017. doi:10.1109/ACCESS.2017

Alfaer, N. M., Aljohani, H. M., Abdel-Khalek, S., Alghamdi, A. S., & Mansour, R. F. (2022). Fusion-Based Deep Learning with Nature-Inspired Algorithm for Intracerebral Haemorrhage Diagnosis. *Journal of Healthcare Engineering*, Volume 2022, Article ID 4409336, 12 pages. https://doi.org/10.1155/2022/4409336

DOI: 10.56892/bima.v8i4B.1204

Al Bataineh, A., Kaur, D., & Jalali, S. M. J. (2022). Multi-Layer Perceptron Training Optimization Using Nature Inspired Computing. *IEEE Access*, 10.1109/ACCESS.2022.3164669.

Shazuli, S. S. M., & Saravanan, A. (2022). Grasshopper Optimization Technique with Deep Learning Driven Retinal Fundus Image Grading and Retrieval. *International Journal of Science and Research (IJSR)*. ISSN: 2319-7064.

Parhi, P., Bisoi, R., & Dash, P. K. (2023). An improvised nature-inspired algorithm enfolded broad learning system for disease classification. *Egyptian Informatics Journal, 24*(4), 241–255.

Parhi, P., Bisoi, R., & Dash, P. K. (2023). An Integrated Nature-inspired Algorithm Hybridized Adaptive Broad Learning System for Disease Classification. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3262167

Hipparage, A., Suryawanshi, G., Patil, P., Agale, P., & Borse, S. (2023). Skin Disease Detection Using Machine Learning and Convolutional Neural Network. *International Research Journal of Modernization in Engineering Technology and Science, 05*(05), Impact Factor- 7.868. https://www.doi.org/10.56726/IRJMETS40689

Renukadevi, T. (2019). Optimizing deep belief network parameters using grasshopper algorithm for liver disease classification. *International Journal of Medical Informatics*. https://doi.org/10.1002/ima.22375

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8:53. https://doi.org/10.1186/s40537-021-00444-8

Gordon, A. C., Smith, T., & Williams, J. (2022). Advancements in the early detection of liver disease. *Clinical Liver Disease, 15*(2), 87-101. https://doi.org/10.1016/j.cld.2022.01.003

Hollinger, F. B., & Liang, T. J. (2022). Hepatitis A: Epidemiology, prevention, and control. *Hepatology Research, 48*(6), 1345-1354. https://doi.org/10.1111/hepr.13879

Khan, M. A., Ali, N., & Khaliq, M. (2023). Hepatitis E virus infection: Current status and future perspectives. *Journal of Clinical Virology, 160*, 105-115. https://doi.org/10.1016/j.jcv.2022.105073

Liaw, Y. F., & Yang, C. S. (2022). Hepatitis C virus infection: Clinical features and management. *Gastroenterology, 162*(3), 771-785. https://doi.org/10.1053/j.gastro.2021.10.028

Singh, P., & Kaur, P. (2023). Importance of early hepatitis diagnosis and treatment. *International Journal of Hepatology, 2023*, 649587. https://doi.org/10.1155/2023/649587

Liu, Y., Wei, X., & Zhang, H. (2023). Deep learning in medical image analysis: A review. *Journal of Biomedical Informatics, 127*, 103862. https://doi.org/10.1016/j.jbi.2022.103862

Shen, D., Wu, G., & Suk, H. I. (2023). Deep learning for medical image analysis. *Annual Review of Biomedical Engineering, 22*, 161-185. https://doi.org/10.1146/annurev-bioeng-060718-102204

DOI: 10.56892/bima.v8i4B.1204

Zhang, L., Zhang, L., & Lin, L. (2022). A survey on deep learning for medical image analysis. *Medical Image Analysis, 67*, 101800. https://doi.org/10.1016/j.media.2020.101800

Deb, K. (2022). Optimization for engineering design: Algorithms and examples. *Springer.* https://doi.org/10.1007/978-3-030-53926-5

Ganaie, M. A., Lee, S. W., & Kim, T. H. (2023). Hyperparameter optimization of neural networks using genetic algorithms: A review. *Applied Soft Computing, 135*, 110827. https://doi.org/10.1016/j.asoc.2023.110827

Goldberg, D. E. (2022). Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley.* https://doi.org/10.5555/105446

Yao, X., Liu, Y., & Lin, G. (2023). Evolutionary computation and machine learning: Advances and applications. *IEEE Transactions on Evolutionary Computation, 27*(2), 207-223. https://doi.org/10.1109/TEVC.2022.3155605