CNN-Driven detection of abnormalities in PCG signals using Gammatonegram analysis

A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Tata Sumanth (21471A05D2)

Katragadda Somnath (21471A0596)

L. V. Pavan Kumar (21471A05D7)

Lingala Brahmaiah (21471A05A0)

Under the esteemed guidance of

Dr. Moturi Sireesha, **M.Tech., Ph.D.**

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)
Accredited by NAAC with A+ Grade and NBA under cycle -1 NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-522601

2024- 2025

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "**CNN-Driven Detection of Abnormalities in PCG Signals Using Gammatonegram Analysis**" is a bonafide work done by the team Tata Sumanth (21471A05D2), Katragadda Somnath (21471A0596), Laghumavarapu Ventaka Pavan Kumar (21471A05D7), Lingala Brahmaiah (21471A05A0) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

Dr. Sireesha Moturi, **M. Tech, Ph.D.**

Associate Professor

PROJECT CO-ORDINATOR

Dr. Sireesha Moturi, **M. Tech, Ph.D.**

Associate Professor

HEAD OFTHE DEPARTMENT

Dr. S. N. Tirumala Rao, **M.Tech., Ph.D.,**

Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled **"CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis"** is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

By

Tata Sumanth	(21471A05D2)
Katragadda Somnath	(21471A0596)
L. V. Pavan Kumar	(21471A05D7)
Lingala Brahmaiah	(21471A05A0)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao**, **B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, Ph.D.,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,** HOD of CSE department and also to our guide **Dr. Sireesha Moturi, M.Tech.,Ph.D.** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. Sireesha Moturi, M.Tech.,Ph.D.,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying our doubts which had really helped us in successfully completing our project.

By

Tata Sumanth (21471A05D2)

Katragadda Somnath (21471A0596)

L.V. Pavan Kumar (21471A05D7)

Lingala Brahmaiah (21471A05A0)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research.

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era

in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed bythe contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engagein independent and life-long learning in the broadest context of technological change.

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S)

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO’s and PO’s:

1. Low level

2. Medium level

3. High level

Project mapping with various courses of Curriculum with Attained PO’s:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C2204.2, C22L3.2	Gathering the requirements and defining the problem, planning to develop a model for detecting abnormalities in PCG signals using CNN and Gammatonegram analysis.	PO1, PO3
CC421.1,C2204.3, C22L3.2	Each requirement is critically analyzed, and the process model for training and testing CNN on PCG signal data is identified.	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done using neural network architecture planning, involving team collaboration to structure the model effectively.	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each module (data preprocessing, CNN model, evaluation) is tested, integrated, and validated for performance.	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done collaboratively by all four members, covering dataset details, methodology, and results.	PO10
CC421.5, C2204.2, C22L3.3	Each phase of the project is periodically presented, demonstrating progress and findings.	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done, and in future updates, the model can be improved for real-time heart sound analysis.	PO4, PO7
C32SC4.3	The physical design includes a web-based interface where users can upload PCG recordings to detect heart abnormalities.	PO5, PO6

ABSTRACT

Phonocardiogram (PCG) signals contain valuable data related to heart function, providing a critical means for early detection of heart diseases. Traditional methods for analyzing these signals often involve manual feature extraction, which can be time-consuming and prone to inaccuracies. In contrast, this study introduces an automated approach that leverages deep learning, specifically Convolutional Neural Networks (CNNs), to classify heart abnormalities. Using Gammatonegram representations, which mimic human auditory processing, the temporal and spectral features of PCG signals are captured more effectively than with traditional techniques like the Short-Time Fourier Transform (STFT) or Mel-Frequency Cepstral Coefficients (MFCCs). These Gammatonegram images are preprocessed through resizing for uniformity and data augmentation to increase the diversity of the training set, improving the CNN's ability to generalize across various heart sounds. The CNN model is then trained to distinguish between healthy and abnormal heart sounds. Several optimization algorithms, including Stochastic Gradient Descent (SGD), Adagrad, Adadelta, RMSProp, and Adam, are evaluated for their performance, with Adam achieving the best result—a perfect test accuracy of 100%. This method not only provides a robust, non-invasive approach to diagnosing heart disease but also offers a scalable solution that can be integrated into telemedicine platforms or portable diagnostic devices, making advanced heart monitoring more accessible in remote or underserved areas. Future directions could include expanding the dataset to cover a broader range of cardiac conditions, refining the CNN architecture, and exploring other machine learning methods such as transfer learning to further enhance diagnostic accuracy. This study underscores the potential of deep learning in transforming cardiovascular care by enabling earlier detection and intervention.

INDEX

LIST OF FIGURES

1. INTRODUCTION

The heart is one of the major organs highly involved in sustaining life by pumping blood throughout the body. It is responsible for ensuring oxygen and other essential nutrients reach tissues while at the same time facilitating the removal of carbon dioxide and other waste products. Heart problems, also referred to as cardiovascular disorders or heart diseases, are many conditions that may involve changes in the structure, function, or electrical conduction of the heart. Such conditions can lead to serious health concerns; therefore, early diagnosis and treatment are of the essence for any good outcomes, preventing major complications and improving results among patients. These abnormalities can arise due to various factors, including genetic predisposition, lifestyle choices, infections, or underlying medical conditions. Common heart problems include arrhythmias (irregular heartbeats), coronary artery disease (narrowing or blockage of the heart's arteries), heart valve issues, cardiomyopathies (diseases of the heart muscle), and congenital heart defects (structure issues present at birth) [1]. These conditions can lead to symptoms such as chest pain, shortness of breath, palpitations, fatigue, and in severe cases, heart failure or sudden cardiac arrest.

Heart diseases are still one of the top causes of death throughout the world, and there is an urgent need to develop trustful and non-invasive methods to diagnose them [2]. Early detection and diagnosis of heart conditions are essential for good care and treatment. Advances in medical technology, such as electrocardiograms (ECGs), echocardiograms, and cardiac imaging techniques, have significantly improved the ability to detect heart abnormalities [3]. Deep learning has transformed many fields, including medical diagnostics. This is because CNNs are effective for image classification tasks as they have the ability to learn features automatically from data. The CNN can examine images, including X-rays, MRI scans, and in this case, Gammatonegram representations of PCG signals, during a medical diagnosis. Phonocardiogram, or PCG, signals, which record the sounds of the heart, serve as a useful modality toward the early detection of heart conditions. The analysis of these signals can provide critical insights into cardiac health, enabling timely intervention and treatment [4].

Traditional methods of abnormality detection in PCG signals generally include two major steps: feature extraction and classification [5]. The Gammatonegram, as in this work, is used here as one of the major techniques for feature extraction from PCG signals. This Gammatonegram represents the time-frequency features of a signal that will most closely match how the human ear actually responds to sound. By converting PCG signals into Gammatonegram images, we can capture both temporal and spectral information that is crucial for distinguishing between healthy and abnormal heart sounds. Compared to other methods similar to the Mel-frequency cepstral coefficients (MFCCs) or Short-Time Fourier Transform (STFT) [6], Gammatonegram provides a more physiologically relevant analysis, especially for biomedical signals like PCG. By converting PCG signals into Gammatonegram images, the intricate patterns of heart sounds are captured effectively, allowing the CNN model to learn from these rich representations [7] . This approach enhances the model's ability to detect abnormalities with higher accuracy, leveraging the detailed auditory features embedded within the Gammatonegram images.

The domain of this research falls within the fields of Biomedical Signal Processing and Medical Diagnostics, with a focus on Cardiovascular Health. To put it precisely, deep learning techniques, particularly those on Convolutional Neural Networks, which have been applied in the analysis and classification of Phonocardiogram signals, the audio recordings of heart sounds. The project also incorporates aspects of Time Frequency Analysis, leveraging Gammatonegram representations to enhance the detection of abnormalities in heart signals. This interdisciplinary domain blends elements of healthcare, signal processing, and artificial intelligence [8].

The technology applied for this is Deep Learning, specifically a variation called the Convolutional Neural Network. CNNs are useful in such tasks as classifying images and as such are highly suitable for Gammatonegram pictures looking at PCG signals. The Gammatonegram, a time-frequency representation of audio signals, is computed using signal processing techniques, providing a rich feature set that repeat human auditory perception. The application is developed with the Python programming language, which is quite robust in handling machine learning and deep learning. Important libraries include TensorFlow or PyTorch for the building and training of the CNN model. These libraries offer the tools necessary to design the neural network, handle the data, and improve the model's performance. NumPy and Pandas are used to handle and prepare data efficiently. Matplotlib or Seaborn helps in showing data and results, and OpenCV is also there, with GPU acceleration making the calculations faster. At the end of the research, this application offers a powerful tool for transforming cardiovascular healthcare delivery, particularly in outside environments. By helping to find problems early, making tests more available, and working with telemedicine and portable devices, it can greatly improve public health, especially in areas with few healthcare services. This progress could save lives, lower healthcare costs, and make life better for many people around the world. innovation.

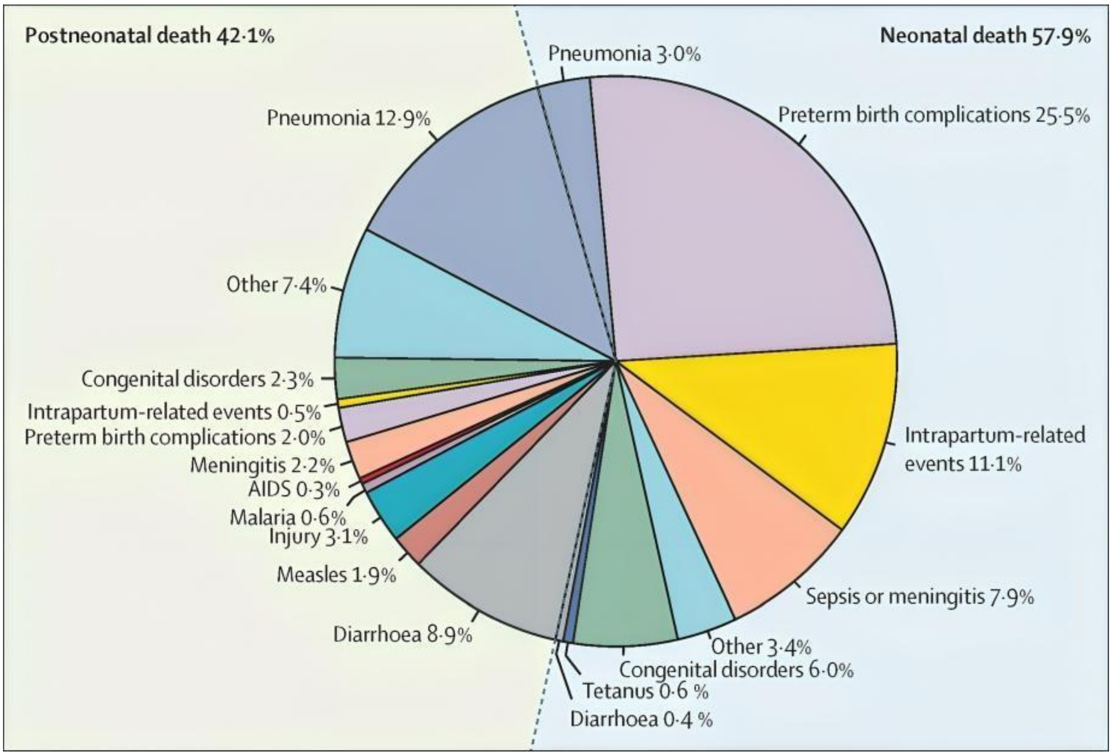


Fig 1.1: Causes of Neonatal and Postneonatal Deaths: A Comparative Analysis.

The Fig 1.1 illustrates the distribution of causes of neonatal (0-28 days) and postneonatal (1-11 months) deaths, highlighting the major contributors to infant mortality. Neonatal deaths account for 57.9% of total infant deaths, with preterm birth complications (25.5%)**,** intrapartum-related events (11.1%)**,** and sepsis or meningitis (7.9%) being the leading causes. Meanwhile, postneonatal deaths make up 42.1%**,** primarily due to pneumonia (12.9%)**,** diarrhea (8.9%)**,** and other infections. The chart provides critical insights into the need for targeted healthcare interventions, emphasizing the importance of neonatal care and infection prevention strategies to reduce infant mortality rates.

2. LITERATURE SURVEY

Before starting this research, we performed a literature survey of various research papers in this domain. This research study provides meaningful insights to move forward in the project:

Ashinikumar Singh, Sinam Ajitkumar Singh, and Aheibam Dinamani Singh developed a robust model based on ensemble learning for the accurate classification of heart sounds. The research leveraged CNN architectures combined with Gammatonegram images for feature representation. The use of ensemble learning techniques such as bagging and boosting resulted in higher accuracy (99.51%) when tested on the PhysioNet 2016 dataset, outperforming traditional singular CNN models. The study highlighted how ensemble models help mitigate overfitting and improve model generalization, ensuring consistent performance across varying datasets. Furthermore, feature extraction techniques like PCA were applied to reduce dimensionality, boosting the computational efficiency of the model while maintaining high classification performance. The success of this model demonstrates the potential of deep learning in practical healthcare applications, particularly in detecting cardiac abnormalities from heart sounds [11].

In this research, Kriti Taneja, Vinay Arora, and Karun Verma explored heart sound classification through the combined use of Gammatonegram images and support vector machines (SVMs). The study proposed the application of texture-based feature extraction techniques such as Linear Ternary Pattern (LTP) and Local Phase Quantization (LPQ) to improve classification accuracy. By experimenting with these feature extraction methods, the authors achieved a classification accuracy of 94% using the PhysioNet CinC 2016 dataset. Moreover, various feature selection methods, including Mutual Information and ReliefF, were employed to optimize the most discriminative features, further enhancing the classification results. Hyperparameter tuning of SVM parameters—such as kernel type (linear, radial, polynomial) and regularization factors—was conducted to maximize classification performance. This research underscores the significance of biologically inspired auditory representations in developing effective heart sound classification systems and provides an alternative to deep learning methods [12].

Arnab Maity, Akanksha Pathak, and Goutam Saha focused their research on the application of transfer learning techniques for heart valve disease classification using Phonocardiogram (PCG) signals. The model utilized pre-trained convolutional neural networks (CNNs) that were fine-tuned with time-frequency representations such as spectrograms and scalograms. With transfer learning, the researchers were able to achieve a high accuracy of 92.23% on the PhysioNet dataset and an outstanding 99.83% on the HVD dataset, proving the efficacy of transfer learning for tasks with limited labeled data. The study explored two main transfer learning strategies—feature extraction and fine-tuning—to enhance classification accuracy. Various pre-trained models, including ResNet, VGG, and Inception, were compared for their performance on PCG signals, with Inception showing superior results. The research successfully demonstrated the feasibility of using transfer learning for medical signal processing, allowing healthcare practitioners to leverage powerful pre-trained models even with small datasets [13].

Yineng Zheng, Xingming Guo, Yang Yang, Hui Wang, Kangla Liao, and Jian Qin proposed a novel approach for identifying left ventricular diastolic dysfunction (LVDD) using a CatBoost machine learning model. The study employed PCG-based spectrogram features to train the model, incorporating linear discriminant analysis (LDA) for feature fusion. The resulting model achieved an AUC (area under the curve) of 91.1%, accuracy of 88.2%, sensitivity of 82.1%, and specificity of 92.7%, making it a highly accurate tool for non-invasive LVDD detection. To further enhance classification accuracy, the authors explored various spectrogram representations, including Short-Time Fourier Transform (STFT) and Wavelet Transform. The CatBoost model's hyperparameters, such as learning rate, depth, and regularization, were finely tuned to improve its robustness and prevent overfitting. This research illustrates the growing role of machine learning models, like CatBoost, in advancing cardiovascular diagnostics, offering a highly accurate and efficient method for diagnosing diastolic dysfunction [14].

Mustafa R. Ismael, Haider J. Abd, and Raad Z. Homod introduced a new classification approach for Phonocardiogram (PCG) signals, utilizing the Wavelet Scattering Transform (WST) combined with the Equilibrium Optimization algorithm. WST was employed to extract robust time-frequency features from the PCG signals, achieving an impressive classification accuracy of 99.5%. This method proved to be highly effective in preserving the intricate characteristics of heart sound signals at various time-frequency scales. By utilizing the Equilibrium Optimization technique, the model's classifier parameters were fine-tuned to maximize performance. A comparative study was conducted against other feature extraction techniques, including Mel-Frequency Cepstral Coefficients (MFCCs) and Gammatonegram representations, with the Wavelet Scattering Transform outperforming these methods. Further, additional optimization strategies, such as Particle Swarm Optimization (PSO), were evaluated to improve the overall classification accuracy. This research highlights the advantages of wavelet-based techniques in heart sound classification and the importance of optimization in building highly accurate and efficient classification models [15].

Cardiovascular disease (CVD), generally called heart illness, is a collective term for various ailments that affect the heart and blood vessels. Heart disease is a primary cause of fatality and morbidity in people worldwide, resulting in 18 million deaths per year. By identifying those who are most vulnerable to heart diseases and ensuring they receive the appropriate care, premature demise can be prevented. Machine learning algorithms are now crucial in the medical field, especially when using medical databases to diagnose diseases. Such efficient algorithms and data processing techniques are applied to predict various diseases and offer much potential for accurate heart disease prognosis. Therefore, this study compares the performance logistic regression, decision tree, and support vector machine (SVM) methods with and without Boruta feature selection. The Cleveland Clinic Heart Disease Dataset acquired from Kaggle, which consists of 14 features and 303 instances, was used for the investigation. It was found that the Boruta feature selection algorithm, which selects six of the most relevant features, improved the results of the algorithms. Among these classification algorithms, logistic regression produced the most efficient result, with an accuracy of 88.52 % [7].

"Deep Learning-Based Multimodal Approach for Heart Sound and ECG Signal Classification" by A. Sharma, P. Gupta, M. Kumawat, and S. Jain, proposed a multimodal approach that combined heart sound signals (PCG) and ECG data for improved diagnosis. By utilizing a hybrid CNN-LSTM model, they effectively extracted spatial features from heart sounds and captured temporal dependencies from ECG signals, achieving a high classification accuracy of 98.7% on publicly available datasets. This dual-signal approach demonstrated the benefits of fusing multiple biosignals to enhance diagnostic precision, presenting a novel solution for non-invasive heart disease detection [8].

3. EXISTING SYSTEM

Traditional methods for detecting heart abnormalities in Phonocardiogram (PCG) signals rely on handcrafted feature extraction and classical machine learning techniques. These methods involve extracting time-domain and frequency-domain features from PCG signals and then classifying them using Support Vector Machines (SVM), k-Nearest Neighbors (KNN), or other conventional classifiers. However, these approaches suffer from several limitations:

Manual Feature Extraction:

Requires domain expertise to select meaningful features, making it time-consuming and prone to human error.

Limited Accuracy:

Traditional machine learning models are often unable to capture the intricate patterns in PCG signals, leading to lower classification performance.

Lack of Generalization:

Models trained on specific datasets may not generalize well to new, unseen data due to variations in recording conditions, noise, and patient differences.

Difficulty in Handling Large Datasets:

As datasets grow larger, traditional approaches struggle to scale effectively, leading to computational inefficiencies.

Feature Engineering Challenges:

Traditional methods require extensive feature engineering, making them less adaptable to complex variations in heart sound signals.

High False Positives and Negatives:

Misclassification rates are higher due to the limited ability of classical approaches to differentiate between subtle abnormalities.

4. PROPOSED METHODOLOGY

To overcome the limitations of traditional methods, this study employs a deep learning-based approach using Convolutional Neural Networks (CNNs) to classify PCG signals. The proposed methodology consists of the following key steps:

1. Data Collection.
2. Data Pre-processing.
3. CNN Architecture.
4. Training Process.
5. Performance Evaluation.

The Fig 4.1, describes the research paradigm in this study:

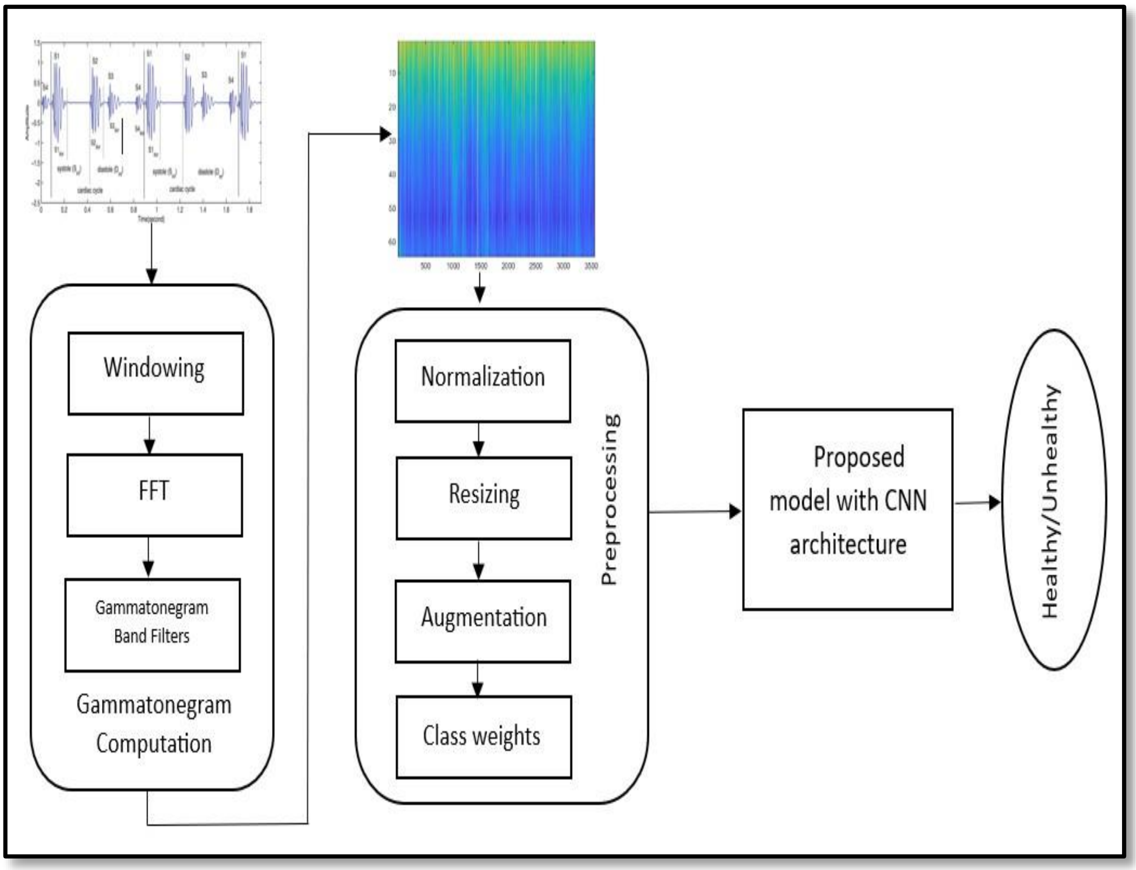


Fig 4.1: Flowchart of Proposed Methodology.

5. SYSTEM REQUIREMENTS

5.1 Hardware Requirements:

- System Type : intel®core™i3-7500UCPU@2.40gh
- Cache memory : 4MB or higher
- RAM : 8GB or higher
- Hard Disk : 256GB or higher

5.2 Software Requirements:

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Flask
- Browser : Any Latest Browser like Chrome

6. SYSTEM ANALYSIS

6.1 Gammatonegram Representation

The Gammatonegram is a graphical representation of an audio signal, modeled after the frequency response of the human auditory system. It is derived using the Gammatone filterbank, a series of bandpass filters that simulate the cochlea's function by capturing energy distribution across different frequencies over time. This transformation allows for effective analysis of Phonocardiogram (PCG) signals, as it emphasizes the perceptual characteristics of heart sounds.

Fig 6.1 represents the transformation from raw PCG signals to Gammatonegram images involves:

Windowing: Dividing the signal into smaller segments.

Fast Fourier Transform (FFT): Converting time-domain signals into the frequency domain.

Filtering: Passing the frequency-domain signal through Gammatone subband filters to simulate human auditory perception.

Image Transformation: Converting filtered outputs into visual representations, highlighting spectral characteristics of healthy and unhealthy heart sounds.

The advantage of using Gammatonegram in medical signal analysis lies in its ability to mimic human auditory perception, making it particularly effective in detecting anomalies in heart sounds that may be subtle yet clinically significant.

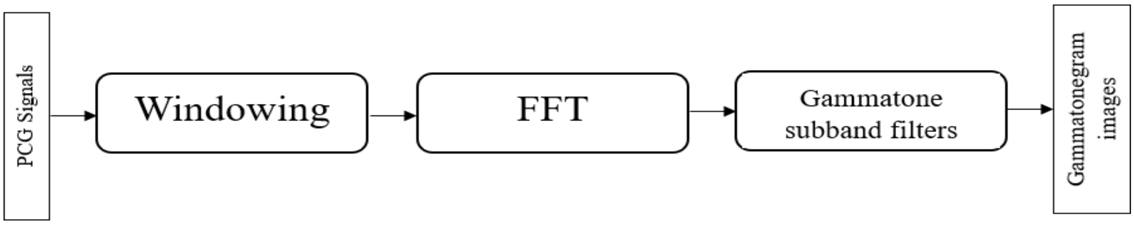


Fig 6.1: PCG Signals to Gammatonegram images.

6.2 Dataset Preparation

The dataset consists of 3,541 Gammatonegram images derived from PCG signals, categorized as 'healthy' or 'unhealthy.' These images serve as training and validation data for the CNN model. The dataset breakdown includes:

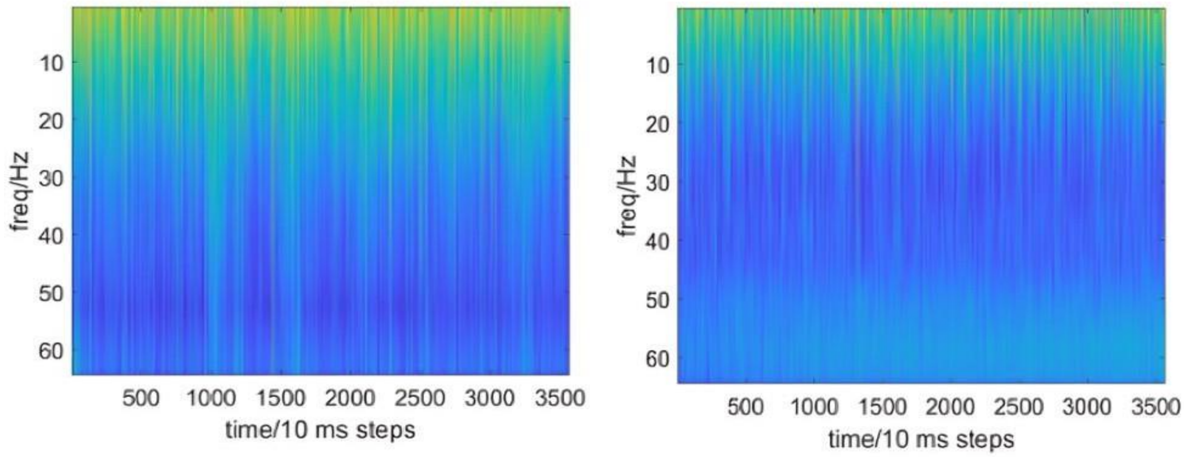
Training Data: 3,240 images (2,575 healthy, 665 unhealthy)

Validation Data: 301 images (150 healthy, 151 unhealthy)

To ensure a high level of model generalization, the dataset is curated with diverse heart sound samples collected from multiple sources, including clinical trials and open- access medical databases.

Ensuring balanced class distribution is crucial for minimizing bias in model predictions. Data augmentation techniques are employed to enhance model robustness by artificially increasing dataset diversity. The dataset preprocessing ensures that signals affected by background noise, patient movement, or varying recording conditions do not introduce artifacts into the training set.

Below Fig 6.2 shows the Gammatonegram images of Healthy and Unhealthy



a) Healthy PCG gammatone image **b) Unhealthy PCG gammatone image**

Fig 6.2: Gammatonegram images of Healthy and Unhealthy.

6.3 Data Preprocessing

To optimize the CNN model's performance, several preprocessing techniques are applied:

Normalization: Scales pixel values between 0 and 1 for uniform input, improving model convergence.

Resizing: Adjusts images to 150×150 pixels for compatibility with the CNN architecture, ensuring consistency in input dimensions.

Data Augmentation: Enhances dataset variability through transformations like flipping, rotation, and zooming, thereby improving model robustness against variations in heart sound recordings.

Fig 6.3 represents the Gammatonegram images of Healthy and Unhealthy after Pre-processing

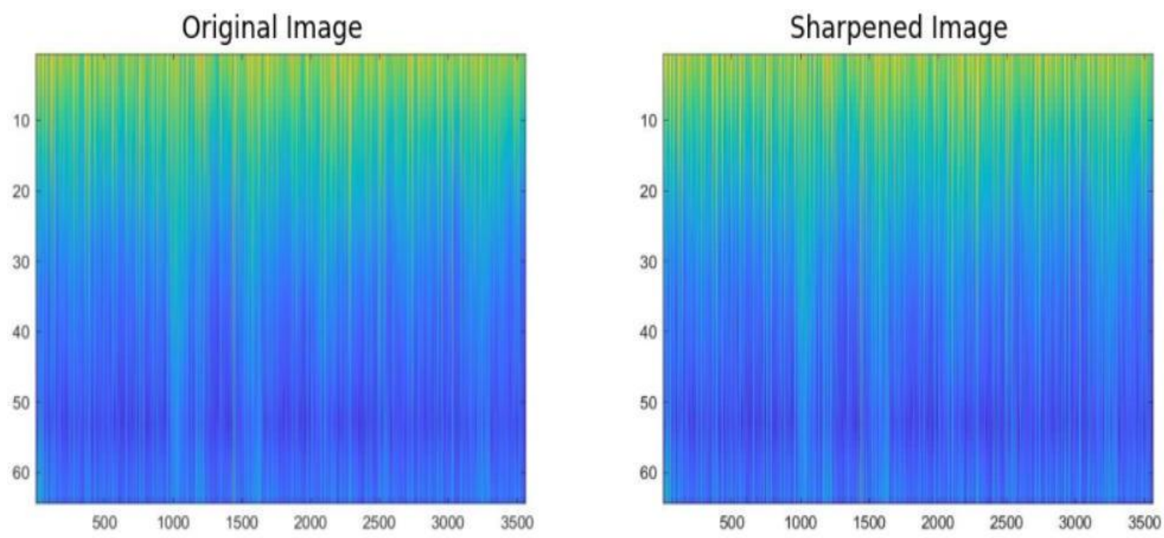


Fig 6.3: Gammatonegram images of Healthy and Unhealthy after Pre-processing

6.4 Algorithms Used

The suggested model uses a Convolutional Neural Network (CNN) design to find problems in Phonocardiogram (PCG) signals by analyzing them with Gammatonegram. The process starts with the input PCG signals, which are changed using Gammatonegram computation to create time-frequency representations needed for capturing important features. Before feeding these representations into the CNN, several preprocessing techniques are applied. These include normalization to scale the data within a specific range, resizing the Gammatone gram images to a consistent dimension, and data augmentation to enhance the model's robustness by artificially expanding the training dataset through techniques like rotation, scaling, and flipping. Additionally, class weights are adjusted to address class imbalance, ensuring the model pays equal attention to underrepresented categories. The CNN architecture is composed of three parallel processing streams. Each stream is initiated with a Convolutional layer that extracts out the spatial features. Then, there is Batch Normalization to make learning more stable, with subsequent use of the Rectified Linear Unit (RELU) as an activation function to introduce non-linearity. Such outputs from these branches are combined into a central block consisting of several layers of Convolution-BatchNorm- ReLU (CBR) combined with Max Pooling layers mixed in. Smaller feature maps are obtained by using the Max Pooling layers, which reduce the size of the feature maps without losing important patterns. The CBR layers are repeated in order to enhance feature extraction and allow the model to better capture small details in the signals. After passing through these convolutional layers, the features are flattened and passed as input to two fully connected layers. These layers serve as the model's decision-making components, transforming the extracted features into classification outputs. The final layer outputs a decision indicating whether the input PCG signal corresponds to a healthy or unhealthy heart condition. To optimize the performance of the model, several optimizers are used to adjust the model weights by minimizing the loss function during training.

6.5 Model Building and Training

The detailed explanation of model building is shown in Fig 6.5, The convolutional neural network is one of the deep learning methods where the feature choice has changed with model learning. A neural network usually consists of three layers they are input layer, hidden layers, and output layer. The proposal has been subjected to various layers of Conv2D, with the first layer having 12 filters, followed by 20 and 32 in the later stages. Each convolutional layer is followed by batch normalization and ReLU activation for normalization and introducing non-linearity for stabilized learning and modeling complex relations. Max-pooling with a filter size of 2x2 was used after the first two convolutional layers, reducing the feature maps from 150x150 to 75x75 and subsequently to 37x37. This retains only the most important elements of the feature maps. Deeper convolutional layers, represented by CBR layers 2 and 3, capture even more complex and high-level patterns. Additional filters were added to detect new patterns while reducing the image size in. It then flattens into a single line of 43,888 features and goes through fully connected layers (dense), which are 256, 128, and 2 in size. Finally, the sounds are classified as ”Healthy” or ”Unhealthy.” This gradually increased filter size and decreased detail help the CNN systematically gather important features from the input Gammatonegram images, thereby coming out with precise heart sound classification.

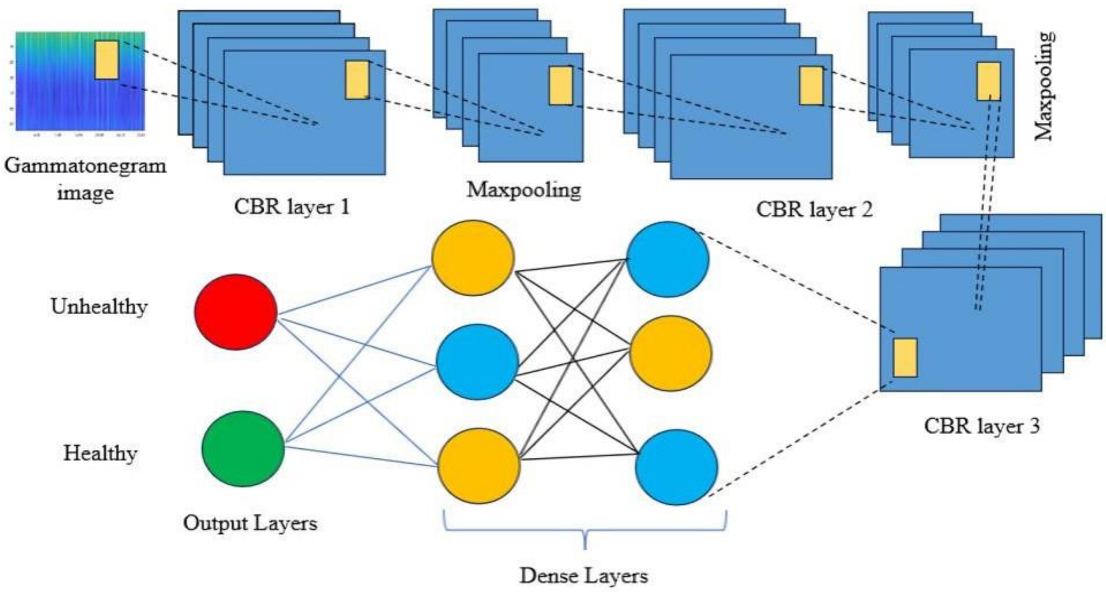


Fig 6.5: Proposed CNN model

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 150, 150, 12)	336
batch_normalization_6 (BatchNormalization)	(None, 150, 150, 12)	48
re_lu_6 (ReLU)	(None, 150, 150, 12)	0
max_pooling2d_4 (MaxPooling2D)	(None, 75, 75, 12)	0
conv2d_7 (Conv2D)	(None, 75, 75, 20)	2,180
batch_normalization_7 (BatchNormalization)	(None, 75, 75, 20)	80
re_lu_7 (ReLU)	(None, 75, 75, 20)	0
max_pooling2d_5 (MaxPooling2D)	(None, 37, 37, 20)	0
conv2d_8 (Conv2D)	(None, 37, 37, 32)	5,792
batch_normalization_8 (BatchNormalization)	(None, 37, 37, 32)	128
re_lu_8 (ReLU)	(None, 37, 37, 32)	0
flatten_2 (Flatten)	(None, 43808)	0
dense_6 (Dense)	(None, 256)	11,215,104
dense_7 (Dense)	(None, 128)	32,896
dense_8 (Dense)	(None, 2)	258

Total params: 11,256,822 (42.94 MB)
Trainable params: 11,256,694 (42.94 MB)
Non-trainable params: 128 (512.00 B)

Fig 6.6: Proposed CNN Architecture

- **Convolutional Layers:** Extract spatial features from input images, identifying patterns indicative of heart sound abnormalities.
- **Batch Normalization:** Stabilizes learning, accelerating training and improving model robustness.
- **ReLU Activation:** Introduces non-linearity to improve feature representation and mitigate gradient vanishing issues.
- **Max Pooling:** Reduces dimensionality while preserving essential information, preventing overfitting.
- **Fully Connected Layers:** Processes extracted features to generate classification outputs with high confidence levels.
- **Softmax Layer:** Determines final classification probabilities, distinguishing between healthy and unhealthy cases.

6.6 Optimizers

Optimizers, in general, play an important role in training neural networks, including CNNs. They adapt the model's weights in a manner that minimizes the loss function of the model, therefore making it perform better. Optimizers modify the model's parameters-weights-after the gradients are computed during backpropagation. The idea is to converge on an optimal set of parameters that minimize the loss function, hence yielding the optimal performance of the model. How this set of changes is accomplished may vary with the optimizer, which then reflects on the speed and manner in which the model will learn. The choice of optimizer can greatly influence convergence speed, escaping local minimum, and overall performance of a model. The optimizers used in the proposed model are SGD, Adagrad, Adadelta, RMSProp, and Adam. More information about these optimizers is discussed below.

a) Stochastic Gradient Descent (SGD):

Stochastic Gradient Descent changes the model weights based on the gradient of the loss concerning each parameter. Contrary to regular Gradient Descent, which uses the whole dataset to calculate the gradient, in SGD, updates of weights are made for every training example or small groups of them. This makes it quicker and, moreover, far better when big datasets are taken into consideration.

b) Adagrad (Adaptive Gradient Algorithm):

Adagrad updates the learning rate independently for every parameter. It makes the learning rate smaller by the magnitude of past gradients. As a result, parameters with large gradients get smaller updates while parameters with small gradients get larger updates. This can be useful when dealing with sparse data.

c) Adadelta:

Adadelta can be seen as a variant of Adagrad that tries to solve the problem of decreasing learning rate. Rather than accumulating all formerly squared gradients, Adadelta restricts the accumulation of gradients to a fixed size. This allows the learning rate to remain adaptive over time.

d) RMSProp:

RMSProp does the same thing as Adagrad but it uses an exponentially decaying average over the squared gradients, instead of accumulating all the squared gradients. This keeps the learning rate manageable and prevents it from shrinking too much.

e) **Adam (Adaptive Moment Estimation):**

Adam is one of the most popular optimizers, combining the advantages of both RMSProp and Momentum (another optimization technique that accelerates SGD). It computes adaptive learning rates for each parameter, and it includes momentum by keeping a running average of both the gradients and their second moments.

The optimizer parameters are

Number of Epochs:

The algorithm is executed multiple times on the training dataset. In this case, we train the model for 25 epochs.

Batch Size:

Batch size is among the most important hyperparameters in the training of neural networks. It defines the number of examples that are used in one model iteration or one forward and backward pass of the model. Batch Size gives the number of samples to update the model parameters. In this model, the batch size used is 32.

Learning rate:

Learning rate is the scaling factor that is needed for updating the model weights. The proposed model uses different optimizers with different learning rates. It is a hyperparameter, which has a greater influence on both the improvement of training and activity of model which is based on the assessment metrics.

Loss Function:

A loss function in general will measure how well your prediction model is doing with respect to predict the expected output. The loss used within this example model is cross entropy loss.

Class Weights:

Class weights are used to handle imbalanced datasets where certain classes have significantly more samples than others. Applying class weights ensures that the model pays equal attention to all classes, preventing bias towards the majority class. The dataset used for this study consists of 3541 Gammatonegram images, separated into training and testing sets with a ratio of approximately 9:1. So class weights play an important role to help the model pay equal attention to all classes and preventing bias towards the majority class.

7. DESIGN

The system begins with a Gammatonegram image as input, representing a transformed version of a PCG signal. The image undergoes preprocessing, which includes resizing to fit the CNN input dimensions, normalization to scale pixel values for better training stability, and augmentation to enhance model generalization by introducing variations in the data. Once preprocessed, the image is passed through a Convolutional Neural Network (CNN), which extracts essential features and learns patterns associated with healthy and unhealthy conditions. The CNN processes the image through multiple layers, including convolution, pooling, activation functions, and fully connected layers, to classify the input. Finally, the model makes a classification decision, determining whether the input corresponds to a healthy or unhealthy heart condition, and the result is displayed to the user.

The 7.1 describes the design overview of Proposed CNN model .

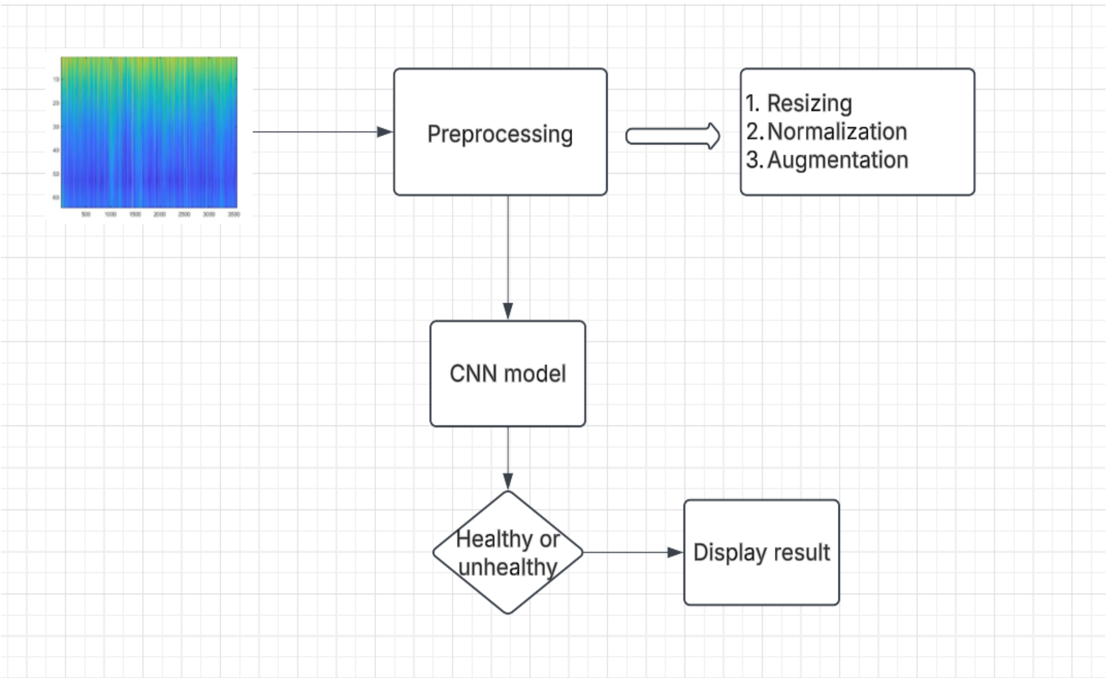


Fig 7.1: Design Overview

8. IMPLEMENTATION

Mounting to Google drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Unzipping the folder

```
!unzip '/content/drive/MyDrive/Gammatonegram.zip'
```

Importing all libraries

```
import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
BatchNormalization, Activation

from tensorflow.keras.optimizers import SGD, Adagrad, Adadelta, RMSprop, Adam
from sklearn.utils.class_weight import compute_class_weight
import numpy as np
```

Defining the data generators with preprocessing steps

```
train_gen = ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True
)

test_gen = ImageDataGenerator(rescale=1./255)
```

Loading data

```
train_data = train_gen.flow_from_directory(

    '/content/Gammatone of heart sounds/Gammatone of heart sounds/train',
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary'
)
```

```
test_data = test_gen.flow_from_directory(
    '/content/Gammatone of heart sounds/Gammatone of heart sounds/val',
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary'
)
```

Class Indicies

```
train_data.class_indices
```

Calculate class weights

```
y_train = train_data.classes
class_weights = compute_class_weight(class_weight='balanced',
classes=np.unique(y_train), y=y_train)
class_weights = dict(enumerate(class_weights))
print(class_weights)
```

Model Building

```
def create_model():
    model = Sequential()
    #CBR Layer1
    model.add(Conv2D(12, (3, 3), input_shape=(150, 150, 3)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #CBR Layer2
    model.add(Conv2D(20, (3, 3)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #CBR Layer3
    model.add(Conv2D(32, (3, 3)))
    model.add(BatchNormalization())
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
```

```
    model.add(Dense(128))
    model.add(Activation('relu'))
    model.add(Dense(64))
    model.add(Activation('relu'))
    model.add(Dense(2, activation='softmax'))
    return model
```

Model training

```
def train_model(optimizer, optimizer_name):
    model = create_model()
    model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
    history = model.fit(
        train_data,
        epochs=25,
        validation_data=test_data,
        class_weight=class_weights
    )
```

Evaluate the model

```
    loss, accuracy = model.evaluate(test_data)
    print(f'Test loss with {optimizer_name}: {loss*100:.4f}')
    print(f'Test accuracy with {optimizer_name}: {accuracy*100:.4f}')
    highest_val_accuracy = max(history.history['val_accuracy'])
    print(f'Highest Validation Accuracy with {optimizer_name}:
            {highest_val_accuracy * 100:.2f}%')
    return history
```

List of optimizers

```
optimizers = {
    'SGD': SGD(learning_rate=0.001),
    'Adagrad': Adagrad(learning_rate=0.01),
    'Adadelta': Adadelta(learning_rate=0.01),
    'RMSProp': RMSprop(learning_rate=0.001),
    'Adam': Adam(learning_rate=0.0001)}
```

Train and evaluate the model with different optimizers

```
histories = {}
for opt_name, opt in optimizers.items():
    print(f"Training with {opt_name} optimizer")
    histories[opt_name] = train_model(opt, opt_name)
```

Prediction code

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

def preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(150, 150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0  # Rescale the image
    return img_array



img_path = '/content/Gammatone of heart sounds/Gammatone of
heartsounds/train/healthy/a0007.jpg'
```

Preprocess the image

```
img_array = preprocess_image(img_path)
print(img_array)
```

Load a saved model

```
model_path = '/content/model_Adam.h5'
model = load_model(model_path)
predictions = model.predict(img_array)
print(predictions)
predicted_class = np.argmax(predictions, axis=1)
class_labels = {v: k for k, v in train_data.class_indices.items()}
print(class_labels)
print(predicted_class[0])
```

predicted_label = class_labels[predicted_class[0]]

print(f'Predicted class: {predicted_label}')

Evaluation using Confusion Metrics and Classification Report

```
import numpy as np
import tensorflow as tf
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
model_path = 'model_Adam.h5'
model = tf.keras.models.load_model(model_path)
# Get true labels and predictions
y_true = []
y_pred = []
for images, labels in test_data:
    preds = model.predict(images)
    y_true.extend(labels)
    y_pred.extend(np.argmax(preds, axis=1))
    if len(y_true) >= test_data.samples:
        break
# Convert to numpy arrays
y_true = np.array(y_true)
y_pred = np.array(y_pred)
# Compute confusion matrix
cm = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:")
print(cm)
# Plot confusion matrix
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Healthy',
'Unhealthy'], yticklabels=['Healthy', 'Unhealthy'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
```

plt.show()

 # Compute classification report

report = classification_report(y_true, y_pred, target_names=['Healthy', 'Unhealthy'])

print("Classification Report:")

print(report)

Flask Code to Connect Front End

import os

import tensorflow as tf

from flask import Flask, request, jsonify from

flask_cors import CORS

from tensorflow.keras.preprocessing import image

import numpy as np

Initializing the Flask app

app = Flask(__name__)

CORS(app) # Enable Cross-Origin Resource Sharing (CORS)

Preprocess the image to match the model input format

```
def preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(150,
    150)) img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0
    img_array /= 255.0  # Normalize the image to [0, 1]
    model=
    tf.keras.models.load_model('./model_Adam.h5')
    predictions = model.predict(img_array)
    predicted_class = np.argmax(predictions,
    axis=1)
    class_labels = {0: 'Healthy', 1: 'Unhealthy'}
    predicted_label = class_labels.get(predicted_class[0],
    'Unknown')
    return predicted_label
```

Route to handle image upload and prediction

```
@app.route('/predict',
methods=['POST']) def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'}), 400
    file
    = request.files['file’]
    if file.filename == '':
        return jsonify({'error': 'No selected file'}), 400
    if file:
        img_path = os.path.join('uploads', file.filename)
        file.save(img_path)
        print(img_path)
        res = preprocess_image(img_path)
        # predicted_label = predict_image(img_array)

        # Return the result as JSON
        return jsonify({'result': res})
    return jsonify({'error': 'Failed to process the
image'}), 500
    if _name_== '_main_':
    # Ensure the 'uploads' directory exists
    if not os.path.exists('uploads'):
        os.makedirs('uploads')
    app.run(debug=True)
```

Home Page HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Abnormality Prediction</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="home.css">
</head>
```

```html
<body>

    <div class="navbar">
        <h2 class="logo">GammaPulse</h2>
        <div class="menu">
            <ul>
                <li><a href="./home.html">HOME</a></li>
                <li><a href="./about.html">ABOUT</a></li>
                <li><a href="./service.html">SERVICE</a></li>
                <li><a href="./Index.html">PREDICTION</a></li>
                <li><a href="./contact.html">CONTACT</a></li>
            </ul>
        </div>
        <div class="search">
            <input class="srch" type="search" name="" placeholder="Type to Search">
            <button class="btn">Search</button>
        </div>
    </div>

    <div class="main">
        <div class="content">
            <h1>Abnormality Prediction</h1>
            <p>Making a difference with accurate health predictions.</p>
            <button class="cn"><a href="#">Learn More</a></button>
        </div>
    </div>

</body>
</html>
```

Home Page CSS

```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

```
body {
  font-family: Arial, sans-serif;
}

/* Navbar */
.navbar {
  width: 100%;
  background-color: #333;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 1rem;
  position: sticky;
  top: 0;
  z-index: 1000;
}

.logo {
  font-size: 2rem;
  color: #ff7200;
  font-family: 'Arial', sans-serif;
  font-weight: bold;
}

.menu ul {
  display: flex;
  list-style: none;
}

.menu ul li {
  margin-left: 2rem;
}

.menu ul li a {
  text-decoration: none;
  color: white;
  font-weight: bold;
```

```
    font-size: 1.2rem;
    transition: color 0.3s ease;
}

.menu ul li a:hover {
    color: #ff7200;
}

.search {
    display: flex;
    align-items: center;
}

.srch {
    padding: 0.5rem;
    border-radius: 5px;
    border: 1px solid #ff7200;
    outline: none;
    font-size: 1rem;
    width: 200px;
    margin-right: 0.5rem;
}

.btn {
    padding: 0.5rem 1rem;
    background-color: #ff7200;
    border: none;
    border-radius: 5px;
    color: white;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.btn:hover {
    background-color: #fff;
    color: #ff7200;
```

```
}

.main {
    width: 100%;
    height: 89vh;
    background: linear-gradient(rgba(0, 0, 0, 0.3), rgba(0, 0, 0, 0.3)),
url('C:/Users/DELL/Final_Project_Frontend/Main/home1.png') no-repeat center
center/cover;
    display: flex;
    justify-content: flex-start;
    align-items: flex-start;
    position: relative;
    color: white;
    text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.7);
    padding-left: 2%;
    padding-top: 5%;
}

/* Text on Image */
.main .content {
    text-align: left;
}

.main .content h1 {
    font-size: 3rem;
    margin-bottom: 1rem;
    color: #ff7200;
}
```

```
.main .content p {
   font-size: 1.5rem;
   color: white;
   margin-top: 1rem;
}

.main .content .cn {
   display: inline-block;
   width:auto;
   margin-top: 2rem;
   padding: 1rem 2rem;
   background-color: #ff7200;
   border-radius: 5px;
   text-transform: uppercase;
   transition: background-color 0.3s ease;
   text-align: center;
   color: white;
}

.main .content .cn:hover {
   background-color: #fff;
   color: #ff7200;
}

.main .content .cn a {
   text-decoration: none;
   color: inherit;
}

/* Responsive Design */
@media (max-width: 768px) {
   .navbar {
      flex-direction: column;
      align-items: flex-start;
   }
```

```
    .menu ul {
        flex-direction: column;
        margin: 1rem 0;
    }

    .search {
        margin-top: 1rem;
    }

    .main .content h1 {
        font-size: 2rem;
    }

    .main .content p {
        font-size: 1.2rem;
    }
}
```

Service Page HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Heart Abnormality Prediction</title>
  <link rel="stylesheet" href="./service.css">
</head>
<body>
  <div class="container">
    <div class="symptoms-form">
      <h2>Heart Abnormality Symptoms</h2>
      <p>Select symptoms that apply to you.</p>
      <p>A brief description will be displayed when you hover over a symptom.</p>

      <form id="symptomForm">
        <div class="symptoms">
          <label>
```

```
        <input type="checkbox" name="symptom" value="Palpitations"
onmouseover="showDescription('Palpitations')" onmouseout="hideDescription()">
        Palpitations
      </label>
      <label>
        <input type="checkbox" name="symptom" value="Chest Discomfort"
onmouseover="showDescription('Chest Discomfort')"
onmouseout="hideDescription()">
        Chest Discomfort
      </label>
      <label>
        <input type="checkbox" name="symptom" value="Dizziness/Fainting"
onmouseover="showDescription('Dizziness/Fainting')"
onmouseout="hideDescription()">
        Dizziness/Fainting
      </label>
      <label>
        <input type="checkbox" name="symptom" value="Fatigue"
onmouseover="showDescription('Fatigue')" onmouseout="hideDescription()">
        Fatigue
      </label>
      <label>
        <input type="checkbox" name="symptom" value="Shortness of Breath"
onmouseover="showDescription('Shortness of Breath')"
onmouseout="hideDescription()">
        Shortness of Breath
      </label>
      <label>
        <input type="checkbox" name="symptom" value="Swelling in Legs/Feet"
onmouseover="showDescription('Swelling in Legs/Feet')"
onmouseout="hideDescription()">
        Swelling in Legs/Feet
      </label>
      <label>
```

```
            <input type="checkbox" name="symptom" value="Irregular Heartbeat"
onmouseover="showDescription('Irregular Heartbeat')"
onmouseout="hideDescription()">
            Irregular Heartbeat
          </label>
        </div>

        <div class="buttons">
          <button type="button" onclick="submitSymptoms()">Submit</button>
          <button type="button" onclick="showReport()">Show Report</button>
        </div>
      </form>

      <div id="report" class="report hidden">
        <p class="report-content">Based on your symptoms, further medical evaluation
is recommended.</p>
      </div>
    </div>
    <div class="image-section"></div>

    <div id="symptomPopup" class="popup hidden">
      <p id="popupText"></p>
    </div>
  </div>

  <script src="./service.js"></script>
</body>
</html>
```

Service Page CSS

```
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
  }
```

```
body {
  font-family: Arial, sans-serif;
  background-color: #e6f2ff;
  height: 100vh;
  overflow: hidden;
}

.container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  position: relative;
}

.symptoms-form {
  background-color: #fff;
  padding: 40px;
  border-radius: 15px;
  box-shadow: 0px 6px 12px rgba(0, 0, 0, 0.2); /* Added a larger shadow */
  width: 450px; /* Increased the width of the card */
  index: 10;
  text-align: center; /* Center the text */
}

.symptoms-form h2 {
  color: #34a853;
  font-size: 24px;
  margin-bottom: 10px;
}

.symptoms-form p {
  color: #333;
  font-size: 14px;
  margin-bottom: 10px;
}
```

```css
.symptoms {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  margin-top: 20px;
}

.symptoms label {
  flex: 0 0 48%;
  margin-bottom: 15px; /* Increased the spacing between labels */
  font-size: 16px; /* Increased font size for better readability */
}

.symptoms {
  display: grid;
  grid-template-columns: repeat(2, 1fr); /* Two columns for checkboxes */
  gap: 10px; /* Spacing between checkboxes */
  margin-top: 20px;
}

.symptoms label {
  display: flex;
  align-items: center; /* Vertically aligns the checkbox and label text */
  font-size: 16px; /* Increased font size for better readability */
}

.symptoms input[type="checkbox"] {
  margin-right: 10px; /* Adds space between the checkbox and the label text */
}


.buttons {
  text-align: center;
  margin-top: 30px; /* Added more margin above buttons */
}

.buttons button {
```

```
  background-color: #007bff;
  color: #fff;
  border: none;
  padding: 12px 25px; /* Increased button size */
  border-radius: 8px; /* Made buttons more rounded */
  cursor: pointer;
  margin: 0 15px; /* Increased spacing between buttons */
  font-size: 16px;
}

.buttons button:hover {
  background-color: #0056b3;
}

.image-section {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: url('./service.jpg');
  background-size: cover;
  background-position: center;
  z-index: 1;
}

.image-section::before {
  content: '';
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.3); /* Added a dark shade overlay */
  z-index: 2;
}
```

```css
.popup {
  position: absolute;
  top: 20%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: rgba(255, 255, 255, 0.9);
  border: 1px solid #ccc;
  padding: 20px;
  width: 300px;
  text-align: center;
  border-radius: 10px;
  z-index: 20;
}

.popup.hidden {
  display: none;
}

.report {
  background-color: #ffe9b3;
  border-radius: 10px;
  padding: 10px;
  margin-top: 10px;
  text-align: center;
}

.report.hidden {
  display: none;
}

.report-content {
  color: #333;
}
```

Service Page JavaScript

```javascript
// Get references to the form, checkboxes, and report container
```

```
const form = document.getElementById("symptomForm");
const checkboxes = form.querySelectorAll('input[type="checkbox"]');
const report = document.getElementById("report");
const reportContent = document.querySelector(".report-content");

// Function to handle form submission
function submitSymptoms() {
  // Get selected symptoms
  const selectedSymptoms = [];
  checkboxes.forEach((checkbox) => {
    if (checkbox.checked) {
      selectedSymptoms.push(checkbox.value);
    }
  });

  // Display an alert or message if no symptoms are selected
  if (selectedSymptoms.length === 0) {
    alert("Please select at least one symptom before submitting.");
    return;
  }

  // Example logic for different symptom outcomes (you can customize this logic)
  let riskLevel = "";
  if (selectedSymptoms.length <= 2) {
    riskLevel = "low risk. Continue monitoring your symptoms.";
  } else if (selectedSymptoms.length <= 4) {
    riskLevel = "medium risk. It's advisable to consult a doctor.";
  } else {
    riskLevel = "high risk. Please seek medical attention immediately and take
prediction test";
  }

  // Update the report content dynamically
  reportContent.innerHTML = `Symptoms indicate ${riskLevel}`;

  // Show the report section after submission
  report.classList.remove("hidden");
```

```
}
// Function to show the report (on button click)
function showReport() {
  if (!report.classList.contains("hidden")) {
    alert("Report is already displayed.");
  } else {
    alert("Please submit the form first to generate the report.");
  }
}
```

Prediction Page HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Abnormality Prediction</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link rel="stylesheet" href="index.css">
</head>
<body>
    <div class="container1">
        <div class="row">
            <div class="col-6">
                <img class="img" src="./Fig1.jpeg" />
            </div>
            <div class="col-6">
                <h1>Abnormality Prediction</h1>
                <div class="upload-container">
                    <form id="imageForm" onsubmit="return false;">
                        <div class="upload-box"
onclick="document.getElementById('imageInput').click();">
```

```
                    <p>Drag & drop your image here or</p>
                    <button type="button" class="upload-button">Browse
Image</button>
                    <input type="file" id="imageInput" accept="image/*"
onchange="displayFileName(event)">
                </div>
                <div class="file-name" id="fileNameDisplay">
                    <!-- File name will be displayed here -->
                </div>
                <br>
                <button class="upload-button"
onclick="submitImage()">Submit</button>
            </form>
        </div>
        <h2 id="result"></h2>
      </div>
    </div>
  </div>

  <script src="index.js"></script>
</body>
</html>
```

Prediction Page CSS

```
*{
  margin: 0;
  padding: 0;
}

body{
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
```

```
    background-color: lightgrey;
}
.container1{
    background-color: white;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px gray;
    text-align: center;
    width:80%;
}
h1{
    margin-top: 10px;
}
.img{
    width:100%;
    height: 100%;
}
input{
    width: 80%;
    height:50px;
    font-size: 25px;
    padding: 10px;
    border-radius: 15px;
}
button{
    padding:5px 20px;
    font-size: 25px;
    border-radius: 10px;
    background-color: rgb(70, 105, 132);
    color:aliceblue;
}
```

```
.upload-container {
  background-color: white;
  padding: 30px;
  border-radius: 10px;
  text-align: center;
 }

 .upload-container h2 {
  margin-bottom: 20px;
  color: #333;
 }

 .upload-box {
  border: 2px dashed #007bff;
  padding: 30px;
  border-radius: 10px;
  position: relative;
  cursor: pointer;
  transition: all 0.3s ease;
 }

 .upload-box:hover {
  background-color: #f0f8ff;
 }

 .upload-box input[type="file"] {
  display: none;
 }

 .upload-box p {
  color: #555;
  font-size: 16px;
  margin: 0;
 }

 .upload-box .upload-button {
  background-color: #007bff;
```

```css
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    margin-top: 10px;
  }

  .upload-box .upload-button:hover {
    background-color: #0056b3;
  }

  .file-name {
    margin-top: 20px;
    font-size: 16px;
    color: #333;
  }
```

Prediction Page JavaScript

```javascript
function displayFileName(event) {
  var file = event.target.files[0];
  var fileNameDisplay = document.getElementById('fileNameDisplay');
  fileNameDisplay.innerHTML = ''; // Clear the previous file name

  if (file) {
      // Display the file name
      fileNameDisplay.textContent = 'Selected file: ' + file.name;
  }
}

function submitImage() {
  const fileInput = document.getElementById("imageInput");
  const formData = new FormData();
  formData.append("file", fileInput.files[0]);
```

```
  // Send the image to the Flask server for prediction
  fetch("http://127.0.0.1:5000/predict", {  // URL to your Flask backend
      method: "POST",
      body: formData,
  })
  .then(response => response.json())
  .then(data => {
      console.log("Prediction result:", data);
      document.getElementById("result").innerHTML = "Prediction: " + data.result;
  })
  .catch(error => {
      console.error("Error:", error);
      alert("Failed to fetch the prediction. Please try again.");
  });
}
```

9. RESULT ANALYSIS

The Gammatonegram images derived from the raw PCG signals were processed through a CNN model, with the accuracy evaluated using different optimizers. The model was trained for 25 epochs, using a batch size of 32, and the training and testing performance was compared across various optimizers. The Adam optimizer showed exceptional results, achieving a test accuracy of 100% by the 25th epoch, with a learning rate of 0.0001.

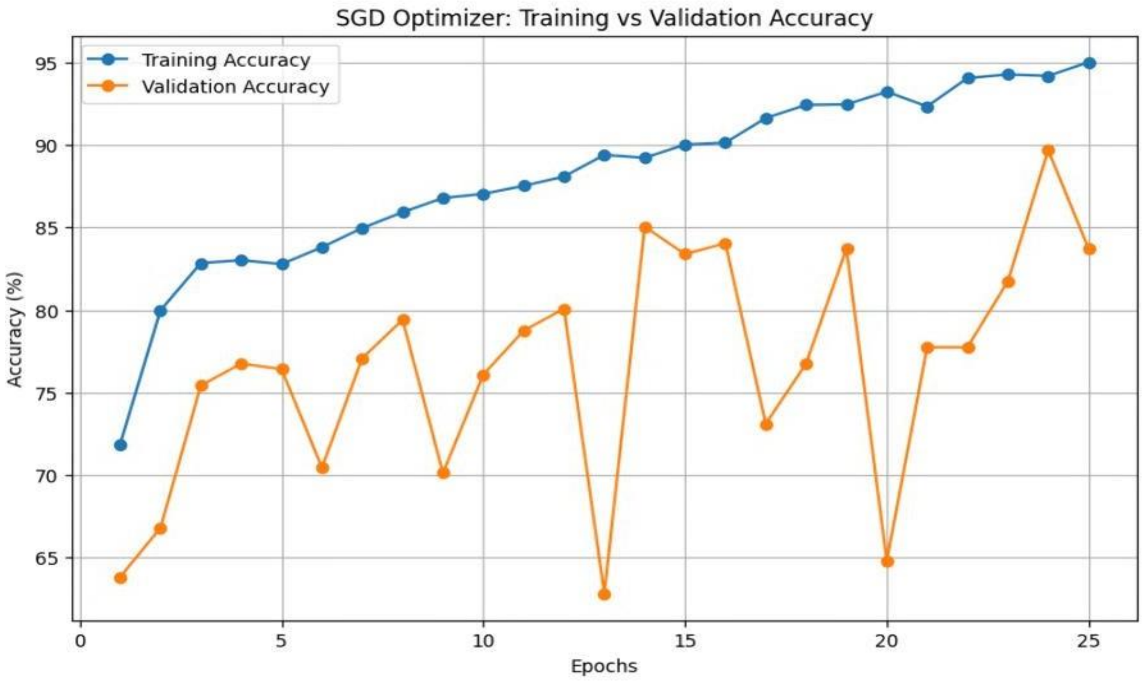


Fig 9.1: Training & validation accuracy of SGD optimizer.

The Fig 9.1 represents the graphs for the SGD optimizer, which reached the maximum training accuracy of 95.03% with a test accuracy of 89.70%.

The Fig 9.2 represents the graph for the Adagrad optimizer, which reached the maximum training accuracy of 97.56% with a test accuracy of 96.01%.

The Fig 9.3 results with Adadelta optimizer, shows that training accuracy is up to 96.76% and test accuracy reaches 92.69%.

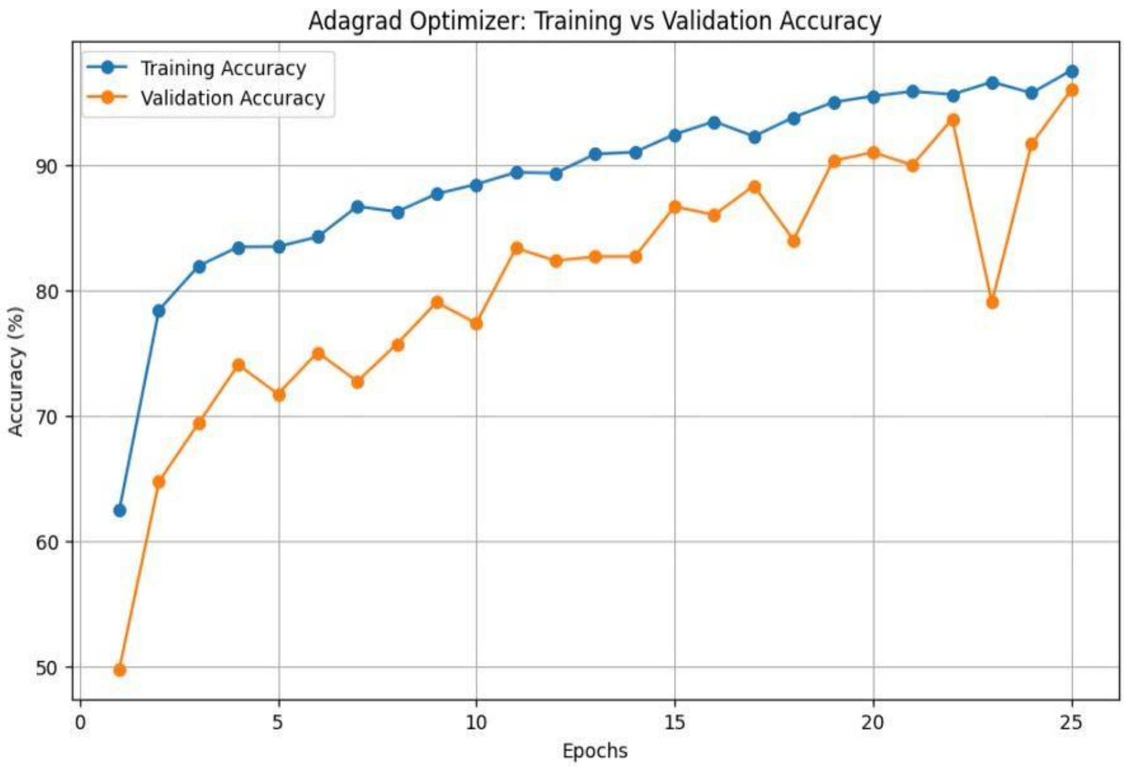


Fig 9.2: Training & validation accuracy of Adagrad optimizer.

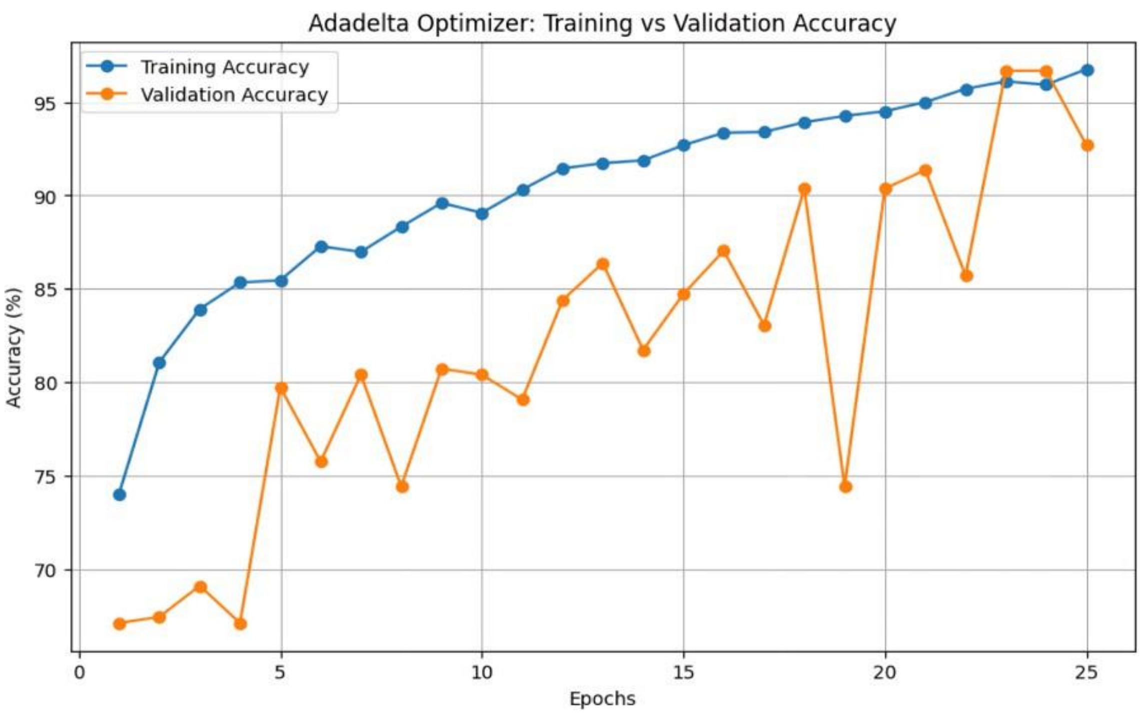


Fig 9.3: Training & validation accuracy of Adadelta optimizer.

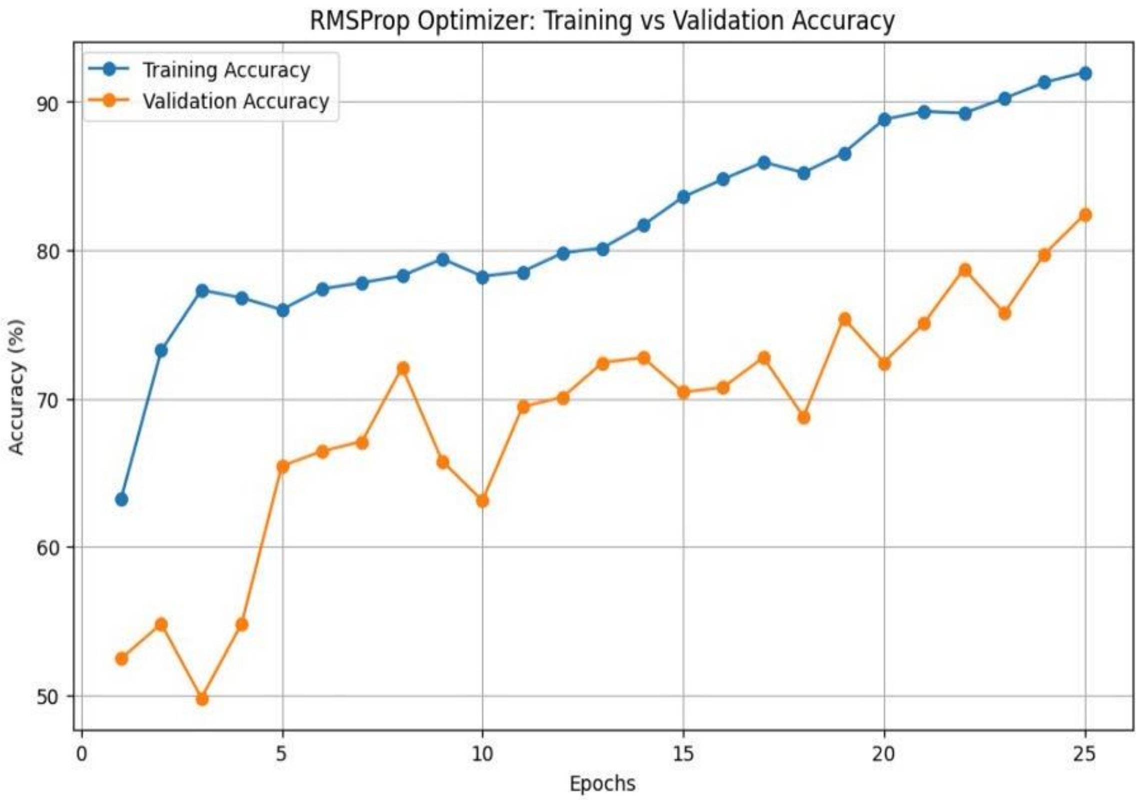


Fig 9.4: Training & validation accuracy of RMSProp optimizer.

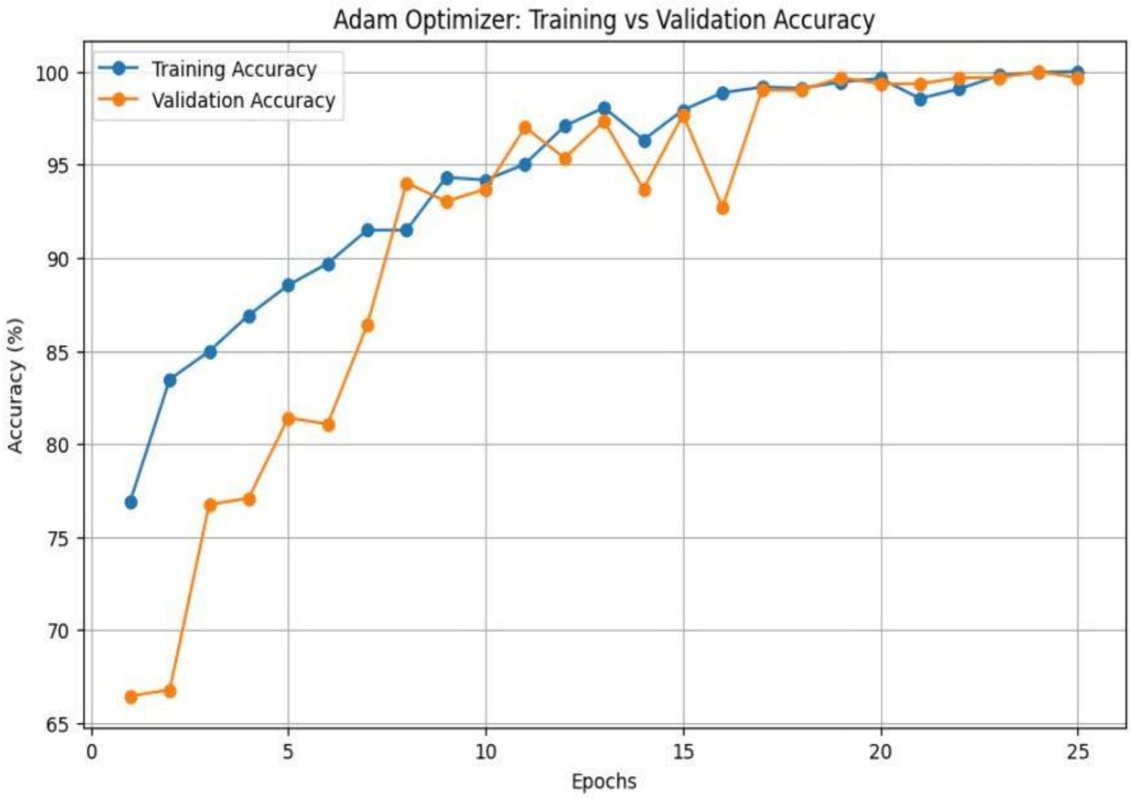


Fig 9.5: Training & validation accuracy of Adam optimizer.

The Fig 9.4 shows Maximum of 91.98% on training accuracy and maximum of 82.39% on test accuracy achieved by RMSProp.

The Fig 9.5, Adam optimizer's performance is represented, which reached the highest training accuracy of 100.00% and best test accuracy of 100%.

The Fig 9.6, represent the evaluation metrics of a classification model of CNN model for abnormality detection in PCG signals.

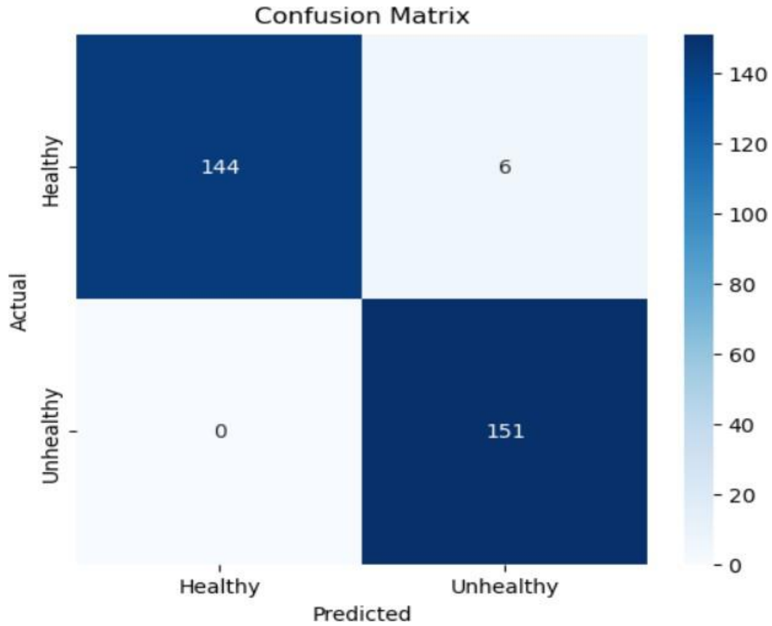


Fig 9.6: Confusion Metrics

The Fig 9.7, represent the classification report of CNN model for abnormality detection in PCG signals.

```
Classification Report:
              precision    recall  f1-score   support

     Healthy       1.00      0.96      0.98       150
   Unhealthy       0.96      1.00      0.98       151

    accuracy                           0.98       301
   macro avg       0.98      0.98      0.98       301
weighted avg       0.98      0.98      0.98       301
```

Fig 9.7: Classification Report

The Comparison of test accuracies across various optimizers is shown in below figure:

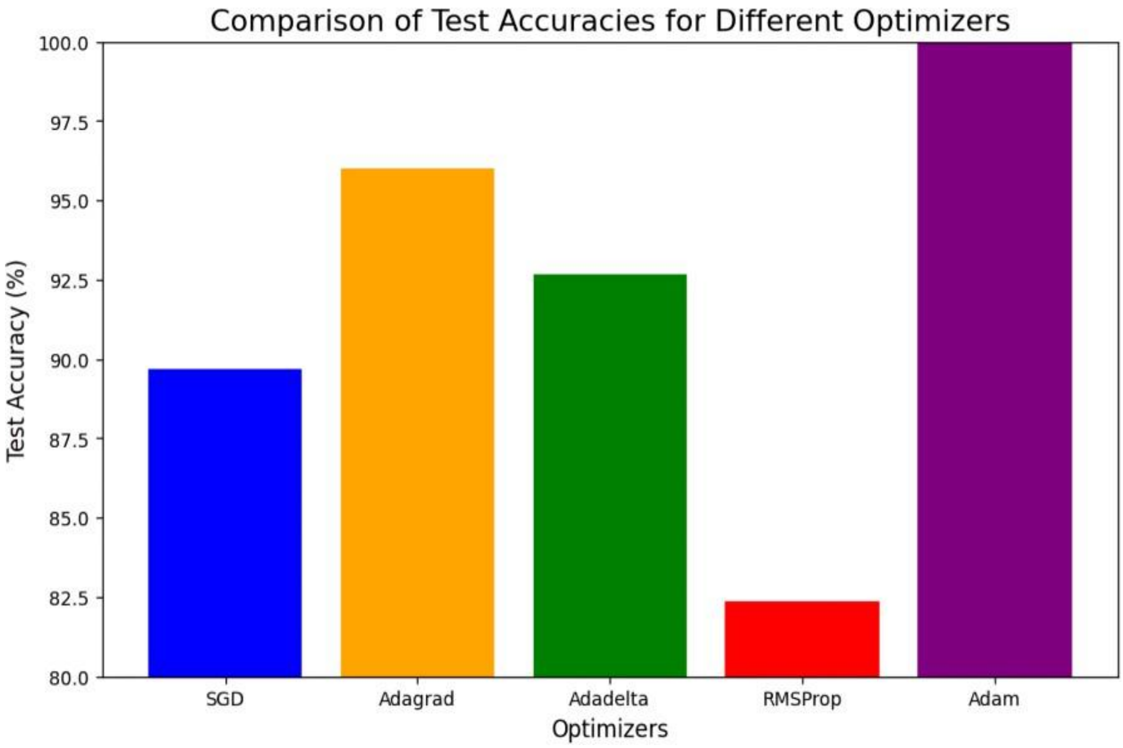


Fig 9.8: Comparison of test accuracies across various optimizers.

The above Fig 9.8 compares the maximum test accuracies of all optimizers. The Adam optimizer achieved the highest training accuracy of 100.00%. The Adagrad and Adadelta optimizers also performed well, with test accuracies of 96.01% and 92.69%, respectively. In contrast, the SGD and RMSProp optimizers exhibited lower performance, with test accuracies of 89.70% and 82.39%. These results underscore the superior effectiveness of the Adam optimizer for the task of PCG signal classification using CNNs. Overall, the Adam optimizer proved to be the most efficient, yielding the highest accuracy and faster convergence, while other optimizers like Adagrad and Adadelta delivered moderate performance.

10. TEST CASES

Test case 1: Healthy

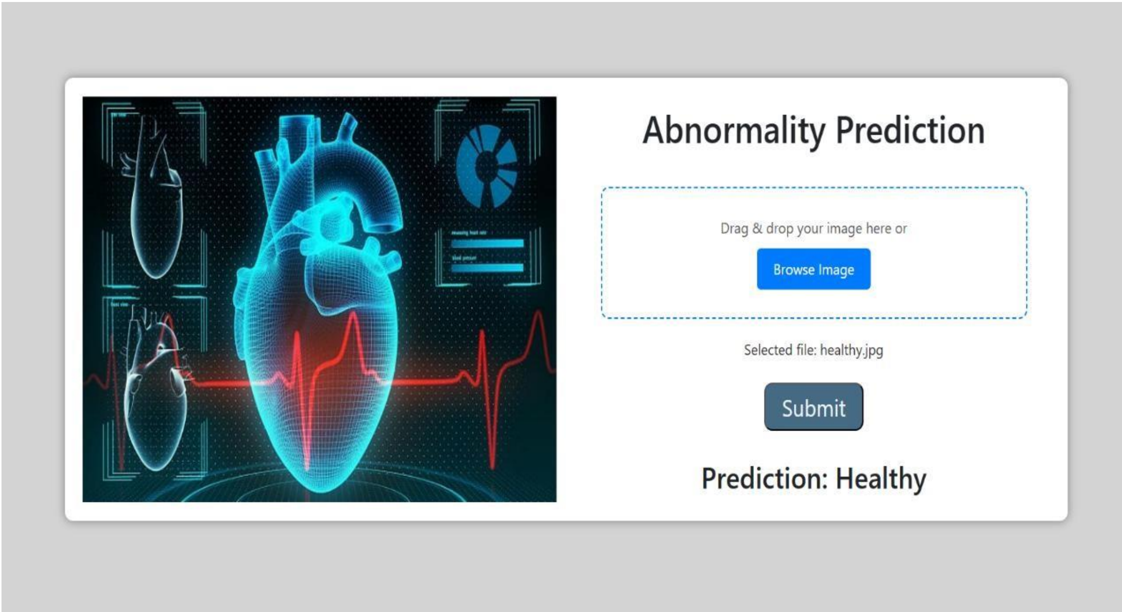


Fig 10.1: Abnormality Prediction as "Healthy".

The Fig 10.1 displays an Abnormality Prediction system for heart health analysis. A healthy heart image is uploaded and predicted as Healthy by the model.

Test case 2: Unhealthy

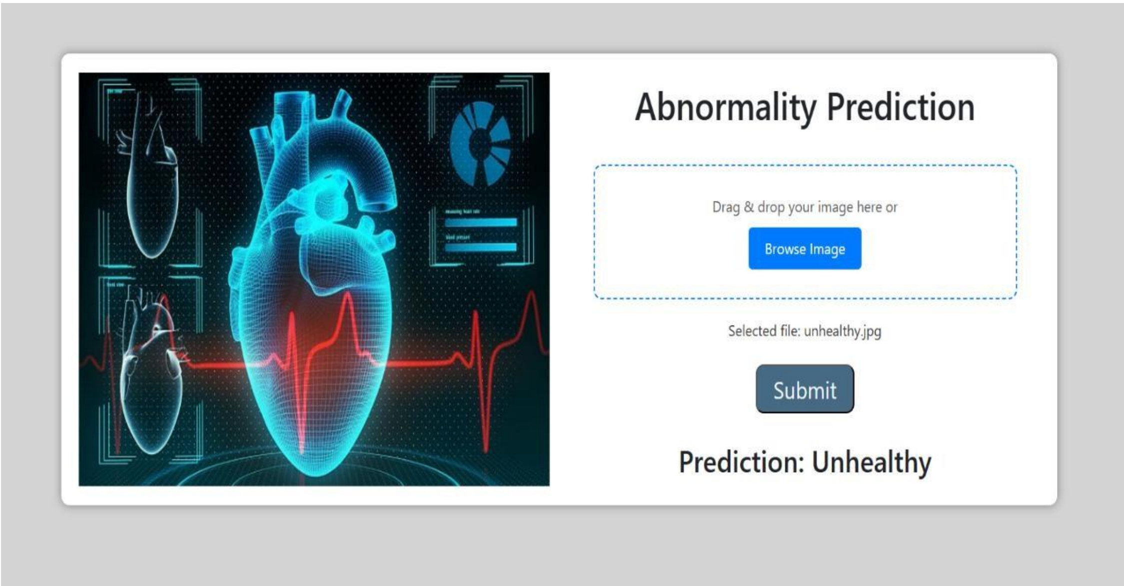


Fig 10.2: Abnormality Prediction as "Unhealthy".

The Fig 10.2 displays an Abnormality Prediction system for heart health analysis. An unhealthy heart image is uploaded and predicted as Unhealthy by the model.

11. USER INTERFACE

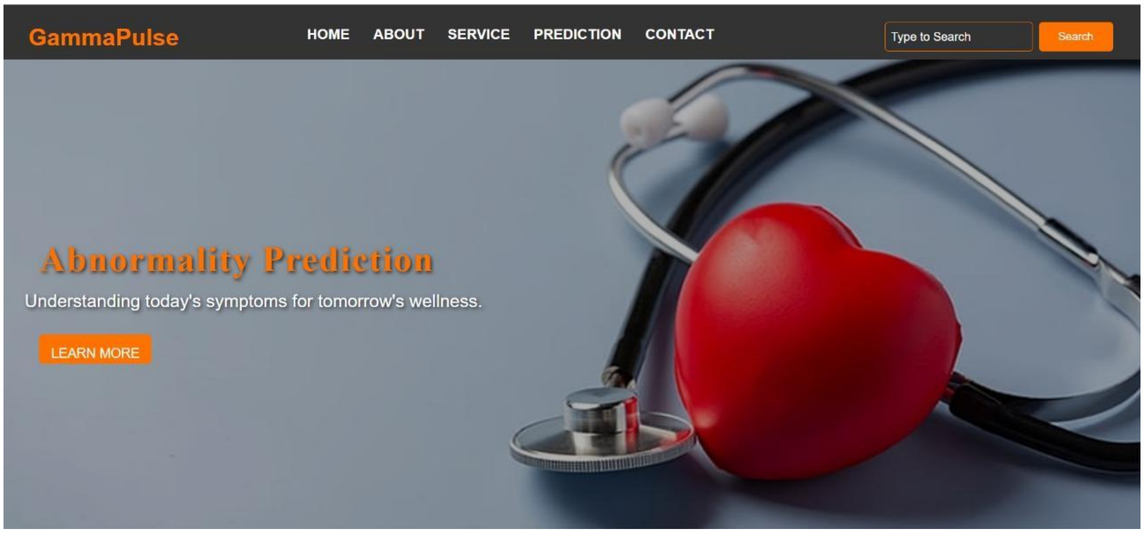


Fig 11.1: Home Screen.

The Fig 11.1 displays the homepage of GammaPulse, a web app for heart abnormality detection. It features a clean UI with a navigation menu and an option to begin symptom analysis.

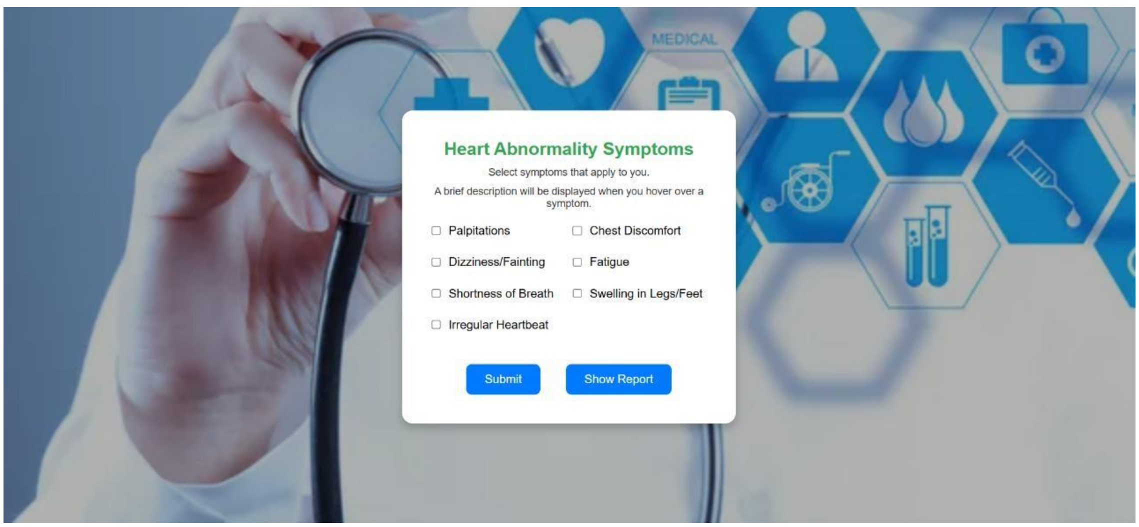


Fig 11.2: Symptoms Screen.

The Fig 11.2 shows the symptom selection interface where users can choose heart related symptoms. A warning system alerts user if critical symptoms are selected.

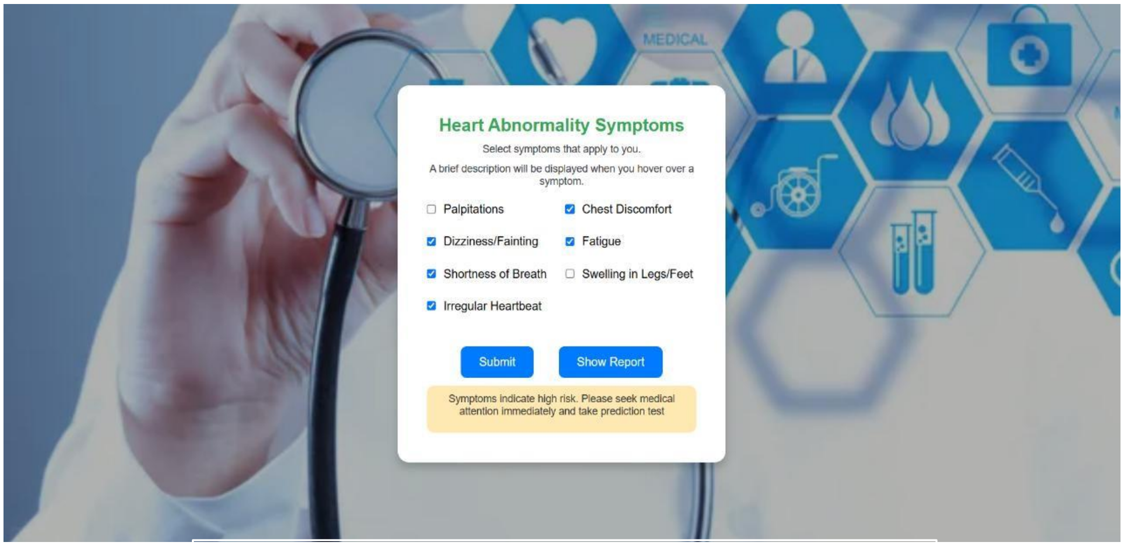


Fig 11.3: Symptoms Screen with Report whether to take prediction test or not.

The Fig 11.3 depicts a pop-up alert informing users of a high-risk condition based on selected symptoms, prompting them to proceed with further diagnostic testing.

12. CONCLUSION & FUTURE WORK

This study presents a simple yet effective approach for detecting abnormalities in Phonocardiogram (PCG) signals using Gammatonegram-based analysis with a Convolutional Neural Network (CNN). By converting PCG signals into Gammatonegram images, the model effectively captures crucial frequency patterns, enabling accurate classification of healthy and unhealthy heart sounds. Preprocessing techniques such as image resizing and random horizontal flip augmentation further improved the model's robustness.

Among the various optimizers tested, Adam achieved the highest accuracy of 100%, demonstrating its superior performance in optimizing deep learning models for PCG classification. The results confirm that Gammatonegram representations are well-suited for analyzing heart sounds, providing a reliable method for heart disease detection.

Future work could focus on expanding the dataset, experimenting with different CNN architectures, and exploring real-time applications for clinical use. Overall, this study highlights the potential of deep learning in biomedical signal processing, offering a promising solution for automated heart sound classification and early disease diagnosis.

13. REFERENCES

1. Cheung, Ka Wang, M. Bonet, K. A. Frank, O. T. Oladapo, G. J. Hofmeyr, WHOIntrapartum Care Algorithms Working Group, Livia Ciabati et al. ”Clinical algorithms for management of fetal heart rate abnormalities during labour.” BJOG: An International Journal of Obstetrics & Gynaecology 131 (2024): 102-111.

2. Tunnell, Natalie C., Sarah E. Corner, Andres D. Roque, Juliet L. Kroll, Thomas Ritz, and Alicia E. Meuret. ”Biobehavioral approach to distinguishing panic symptoms from medical illness.” Frontiers in Psychiatry 15 (2024): 1296569.

3. Duca, S ¸tefania-Teodora, Ionut¸ Tudorancea, Radu-S ¸tefan Miftode, Adriana Chetran, Alexandru-Dan Costache, D. Robert Pav˘ al, and IrinaIuliana Costache-Enache. ”Heartbeat Chronicles: Decoding the Interplay of Echocardiography and Heart Rate Variability in Chronic Heart Failure Patients–Unraveling the Mysteries with Traditional and Advanced 24Hour Holter ECG Parameters.” Romanian Journal of Cardiology.

4. Lee, Joon-Hyop, EunKyung Ku, Yoo Seung Chung, Young Jae Kim, and Kwang Gi Kim. ”Intraoperative detection of parathyroid glands using artificial intelligence: optimizing medical image training with data augmentation methods.” Surgical Endoscopy 38, no. 10 (2024): 57325745.

5. M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease Prediction International Journal of Recent Technology and Engineering Vol- 7, No 6, Mar- 2019 ISSN- 2277-3878, Pages– 1754– 1772.

6. Lee, T. H., E. F. Shair, A. R. Abdullah, K. A. Rahman, and N. Nazmi. ”Comparison of Short Fast Fourier Transform and Continuous Wavelet Transform in Study of Stride Interval.”

7. M.Sireesha, Srikanth Vemuru, S.N.Tirumala Rao “Classification Model for Prediction of Heart Disease Using Correlation Coefficient Technique” International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 2, March- April 2020, Pages- 21162123.

8. Chambi, Edwin M., Jefry Cuela, Milagros Zegarra, Erasmo Sulla, and Jorge Rendulich. ”Benchmarking Time-Frequency Representations of Phonocardiogram Signals for Classification of Valvular Heart Diseases Using Deep Features and Machine Learning.” Electronics 13, no. 15 (2024): 2912.

9. Arslan, Muhammad, Muhammad Mubeen, Muhammad Bilal, and Saadullah Farooq Abbasi. ”1D-CNN-IDS: 1D CNN-based Intrusion Detection System for IIoT.” arXiv preprint arXiv:2409.08529 (2024).

10. Ahmad, Ahsan, Ali Husnain, Ashish Shiwlani, Ahad Hussain, Mahnoor N. Gondal, and Ayesha Saeed. ”Ethical and clinical implications of AI integration in cardiovascular healthcare.” World Journal of Advanced Research and Reviews 23, no. 3 (2024): 2479-2501.

11. ASHINIKUMAR SINGH, S. I. N. A. M., S. I. N. A. M. Ajitkumar SINGH, and A. H. E. I. B. A. M. DINAMANI SINGH. ”Ensemble Learning for Accurate Prediction of Heart Sounds using Gammatonegram Images.” Turkish Journal of Electrical Engineering and Computer Sciences 32, no. 4 (2024): 555-573.

12. Taneja, Kriti, Vinay Arora, and Karun Verma. ”Heart sound classification method using gammatonegram and SVM.” Multimedia Tools and Applications (2024): 1-37.

13. Maity, Arnab, Akanksha Pathak, and Goutam Saha. ”Transfer learning based heart valve disease classification from Phonocardiogram signal.” Biomedical Signal Processing and Control 85 (2023): 104805.

14. Zheng, Yineng, Xingming Guo, Yang Yang, Hui Wang, Kangla Liao, and Jian Qin. ”Phonocardiogram transfer learning-based CatBoost model for diastolic dysfunction identification using multiple domain-specific deep feature fusion.” Computers in Biology and Medicine 156 (2023): 106707.

15. Ismael, Mustafa R., Haider J. Abd, and Raad Z. Homod. ”Classification of Phonocardiogram Signals Using the Wavelet Scattering Transform and Equilibrium Optimization Approach.” (2024).

16. For dataset refer this link: https://www.kaggle.com/datasets/brundavanamsatyasai/gammatoneimages-of-heart-sounds

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030

Sireesha Moturi
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF

IEEE SICSR PUNE STUDENT BRANCH

IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030

Sumanth Tata
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF | IEEE SICSR PUNE STUDENT BRANCH | IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

2024 First IEEE International Conference for

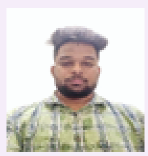
WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030



Somnath Katragadda
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale
General Chair

Prof. Dr. Rajashree Jain
General Chair

Dr. Surekha Deshmukh
Honorary Chair

Prof. Dr. Jatinderkumar R. Saini
Honorary Chair

Our Partners

SSBF

IEEE SICSR PUNE STUDENT BRANCH

IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

IEEE PUNE SECTION

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030

Venkata Pavan Kumar Laghumavarapu
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale	Prof. Dr. Rajashree Jain	Dr. Surekha Deshmukh	Prof. Dr. Jatinderkumar R. Saini
General Chair	General Chair	Honorary Chair	Honorary Chair

Our Partners

SSBF | IEEE SICSR PUNE STUDENT BRANCH | IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo

Flagship Conference of WIE, Pune Section

IEEE Pune Section

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030

Brahmaiah Lingala
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale General Chair	Prof. Dr. Rajashree Jain General Chair	Dr. Surekha Deshmukh Honorary Chair	Prof. Dr. Jatinderkumar R. Saini Honorary Chair

Our Partners

SSBF | IEEE SICSR PUNE STUDENT BRANCH | IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

SYMBIOSIS INSTITUTE OF COMPUTER STUDIES AND RESEARCH

InCoWoCo
Flagship Conference of WIE, Pune Section

IEEE Pune Section

2024 First IEEE International Conference for

WOMEN IN COMPUTING (INCOWOCO 2024)

14 - 15, November 2024 | Pune, Maharashtra, India

CERTIFICATE

This certificate is presented to

WCS 5030

Dodda Venkata Reddy
Dept of CSE,
Narasaraopeta Engineering College,
Narasaraopet-522601, Palnadu,
Andhra Pradesh, India

for presenting the research paper entitled "CNN-Driven detection of Abnormalities in PCG signals using Gammatonegram Analysis." in the 2024 First IEEE International Conference for Women in Engineering (INCOWOCO 2024) held at Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, Maharashtra, India during 14 - 15, November 2024. The conference is technically co-sponsored by IEEE Women in Engineering (WiE) of Pune Section and IEEE Pune Section.

Dr. Priya Gokhale General Chair	Prof. Dr. Rajashree Jain General Chair	Dr. Surekha Deshmukh Honorary Chair	Prof. Dr. Jatinderkumar R. Saini Honorary Chair

Our Partners

SSBF | IEEE SICSR PUNE STUDENT BRANCH | IEEE Education Society PUNE SECTION

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

CNN-Driven Detection of Abnormalities in PCG Signals Using Gammatonegram Analysis

Sireesha Moturi, Sumanth Tata, Somnath Katragadda, Venkata Pavan Kumar Laghumavarapu, Brahmaiah Lingala, Dodda Venkata Reddy
Department of Computer Science and Engineering
Narasaraopeta Engineering College, Narasaraopet-522601, Palnadu, Andhra Pradesh, India
sireeshamoturi@gmail.com, tatasumanth2003@gmail.com, katragaddasomnath22@gmail.com, pavanklaghumavarapu@gmail.com, lingalabrahmaiah399@gmail.com, doddavenkatareddy@gmail.com

2024 First International Conference for Women in Computing (InCoWoCo) | 979-8-3315-1894-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/INCOWOCO64194.2024.10863151

***Abstract*—Phonocardiogram (PCG) signals carry essential data about heart health, offering significant potential for early dis- ease detection. This study presents a method to identify heart abnormalities using Gammatonegram images derived from PCG signals, which are analyzed with a Convolutional Neural Network (CNN). The Gammatonegram, a time-frequency representation mimicking human auditory processing, captures both temporal and spectral features of PCG signals. The images are pre-processed through resizing and data augmentation, the images are input into the CNN model for classification. The model distinguishes between healthy and unhealthy heart sounds, and its performance is evaluated using several optimizers—SGD, Adagrad, Adadelta, RMSprop, and Adam. Adam yielded the best performance, achieving a test accuracy of 100%. This method provides a robust, non-invasive solution for heart disease diagnosis, leveraging deep learning to improve accuracy in detecting heart abnormalities from PCG signals.**

***Index Terms*—PCG signals, Gammatonegram images , Proposed model CNN, Classification, Optimizers.**

I. Introduction

The heart is one of the major organs highly involved in sustaining life by pumping blood throughout the body. It is responsible for ensuring oxygen and other essential nutrients reach tissues while at the same time facilitating the removal of carbon dioxide and other waste products. Heart problems, also referred to as cardiovascular disorders or heart diseases, are many conditions that may involve changes in the structure, function, or electrical conduction of the heart. Such conditions can lead to serious health concerns; therefore, early diagnosis and treatment are of the essence for any good outcomes, preventing major complications and improving results among patients. These abnormalities can arise due to various factors, including genetic predisposition, lifestyle choices, infections, or underlying medical conditions. Common heart problems include arrhythmias (irregular heartbeats), coronary artery disease (narrowing or blockage of the heart's arteries), heart valve issues, cardiomyopathies (diseases of the heart muscle), and congenital heart defects (structure issues present at birth) [1]. These conditions can lead to symptoms such as chest pain, shortness of breath, palpitations, fatigue, and in severe cases, heart failure or sudden cardiac arrest.

Heart diseases are still one of the top causes of death throughout the world, and there is an urgent need to develop trustful and non-invasive methods to diagnose them [2]. Early detection and diagnosis of heart conditions are essential for good care and treatment. Advances in medical technology, such as electrocardiograms (ECGs), echocardiograms, and cardiac imaging techniques, have significantly improved the ability to detect heart abnormalities [3].

Deep learning has transformed many fields, including medical diagnostics. This is because CNNs are effective for image classification tasks as they have the ability to learn features automatically from data. The CNN can examine images, including X-rays, MRI scans, and in this case, Gammatonegram representations of PCG signals, during a medical diagnosis. Phonocardiogram, or PCG, signals, which record the sounds of the heart, serve as a useful modality toward the early detection of heart conditions. The analysis of these signals can provide critical insights into cardiac health, enabling timely intervention and treatment [4].

Traditional methods of abnormality detection in PCG signals generally include two major steps: feature extraction and classification [5]. The Gammatonegram, as in this work, is used here as one of the major techniques for feature extraction from PCG signals. This Gammatonegram represents the time-frequency features of a signal that will most closely match how the human ear actually responds to sound. By converting PCG signals into Gammatonegram images, we can capture both temporal and spectral information that is crucial for distinguishing between healthy and abnormal heart sounds. Compared to other methods similar to the Mel-frequency cepstral coefficients (MFCCs) or Short-Time Fourier Transform (STFT) [6], Gammatonegram provides a more physiologically relevant analysis, especially for biomedical signals like PCG. By converting PCG signals into Gammatonegram images, the intricate patterns of heart sounds are captured effectively, allowing the CNN model to learn from these rich representations [7]. This approach enhances the model's ability to detect abnormalities with higher accuracy, leveraging the detailed auditory features embedded within the Gammatonegram images.

The domain of this research falls within the fields of Biomedical Signal Processing and Medical Diagnostics, with a focus on Cardiovascular Health. To put it precisely, deep learning techniques, particularly those on Convolutional Neural Networks, which have been applied in the analysis and classification of Phonocardiogram signals, the audio recordings of heart sounds. The project also incorporates aspects of Time-Frequency Analysis, leveraging Gammatonegram representations to enhance the detection of abnormalities in heart signals. This interdisciplinary domain blends elements of healthcare, signal processing, and artificial intelligence [8].

The technology applied for this is Deep Learning, specifically a variation called the Convolutional Neural Network. CNNs are useful in such tasks as classifying images and as such are highly suitable for Gammatonegram pictures looking at PCG signals [9]. The Gammatonegram, a time-frequency representation of audio signals, is computed using signal processing techniques, providing a rich feature set that repeat human auditory perception. The application is developed with the Python programming language, which is quite robust in handling machine learning and deep learning. Important libraries include TensorFlow or PyTorch for the building and training of the CNN model. These libraries offer the tools necessary to design the neural network, handle the data, and improve the model's performance. NumPy and Pandas are used to handle and prepare data efficiently. Matplotlib or Seaborn helps in showing data and results, and OpenCV is also there, with GPU acceleration making the calculations faster. Jupyter Notebooks offer the place to work on this complete and modern way of analyzing heart sounds.

At the end of the research, this application offers a pow- erful tool for transforming cardiovascular healthcare delivery, particularly in outside environments [10]. By helping to find problems early, making tests more available, and working with telemedicine and portable devices, it can greatly improve public health, especially in areas with few healthcare services. This progress could save lives, lower healthcare costs, and make life better for many people around the world.

II. Literature Study

In recent years, heart sound analysis has focused on addressing imbalanced signals in cardiovascular diagnostics. A significant study by ASHINIKUMAR SINGH et al. demonstrated the effectiveness of ensemble learning with CNNs and Gammatonegram images for predicting heart sounds. The Turkish Journal has published their research, which attained an accuracy rate of 99.51% on the PhysioNet 2016 dataset, that proves advanced deep learning techniques can contribute to the betterment of heart sound classification and, as a result, cardiovascular disease diagnosis [11].

A study by Taneja, Arora, and Verma enhances heart sound signal classification by using the PhysioNet CinC 2016 dataset. The novel approach introduced involves using Gammatonegram—biologically-inspired images that mimic auditory processing in the cochlea—to enhance event detection in cardiac audio signals. By extracting texture-related features from these Gammatonegrams, including Linear Ternary Pattern (LTP) and Local Phase Quantization (LPQ), the study achieved superior classification performance. The accuracy it achieved was 94.00%. Precision and F1 scores also reached 91.77 and 93.61, respectively. This was better as compared to other image representations [12]. Another study discusses the benefits of transfer learning through pre-trained convolutional neural networks for automatic classification of limited data. It uses standard time-frequency representations as input features, such as spectrogram, log-Mel spectrogram, and scalogram, for fine-tuning lightweight models pre-trained on audio and images for the classification of PCG. Four varieties of heart sound data are classified using the transfer learning method based on YAMNet. The proposed approach obtains an overall accuracy of 99.83%, sensitivity of 99.59%, and specificity of 99.90%. At the same time, it classified the PhysioNet/CinC Challenge 2016 data into two classes with an accuracy of 92.23% [13].

From another paper research a non-invasive method for detecting left ventricular diastolic dysfunction (LVDD) using a PCG-based transfer learning CatBoost model. Features were extracted by four spectrogram representations namely, STFT, MFCCs, S-transform and gammatonegram combined with pre-trained CNNs such as Xception, ResNet50, VGG16 and InceptionResNetv2. Features were fused and classified using CatBoost, outperforming other machine learning classifiers. The best model, with LDA feature fusion, achieved an AUC of 91.1%, 88.2% accuracy, 82.1% sensitivity, 92.7% specificity, and an F1-score of 0.892. This approach could help in non-invasive LVDD detection [14].

III. Proposed work

A. Gammatonegram

The Gammatonegram is in fact a graphical representation of an audio signal, just the name says, similar to the process in which the human auditory system copes with sound. It is based on the Gammatone filterbank, which is a series of bandpass filters modeled after the frequency response of the human ear, particularly the cochlea. Each filter in the bank is tuned to a different frequency, allowing the Gammatonegram to capture the distribution of energy across different frequencies over time, like how the cochlea analyzes sound.

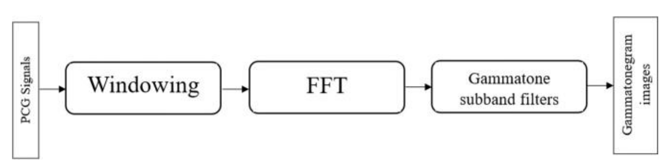


Fig. 1: PCG Signals to Gammatonegram images.

In the context of heart sound analysis, a Gammatonegram can be applied to phonocardiogram (PCG) signals to highlight specific features that may indicate abnormalities in heart function. Fig.1 shows the transformation from a sound signal to a Gammatonegram image. The process begins with a sound signal. Then, the signal is divided into small parts called windows or frames to make it easier to analyze. The next step is to use the Fast Fourier Transform (FFT) on these small parts. The FFT changes the signal from time-based to frequency-based, which helps to look at the different frequency parts of the signal separately. Once the frequency-domain representation is obtained, it is passed through Gammatone subband filters. These filters are designed to simulate the human auditory system's response to various frequencies, particularly by emphasizing frequencies that are more relevant to human hearing. The output from the Gammatone filters is then transformed into Gammatone images, which visually represent the frequency information in a way that reflects the perceptual characteristics of the sound as it would be interpreted by the human ear. This is shown in the figure below, with the healthy and unhealthy PCG signals samples filtered into images using gammatone subband filters.

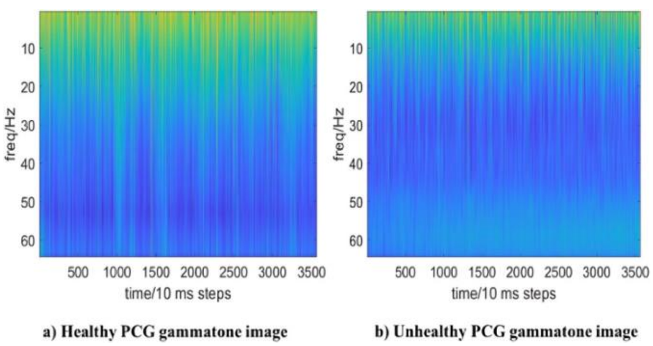


Fig. 2: Healthy and Unhealthy PCG signals samples filtered into images using gammatone subband filters.

B. Dataset

The dataset is crucial to the successful training and validation of the CNN model for detecting abnormalities in Phonocardiogram (PCG) signals. It consists of a total of 3,541 Gammatonegram images derived from PCG recordings, categorized into two main classes: 'healthy' and 'unhealthy'. These images represent the time-frequency analysis of heart sounds, capturing both temporal and spectral information that is essential for distinguishing normal heart function from potential abnormalities. For dataset refer [14]

Dataset Breakdown:

- **Training Data:** Out of the total, 3,240 images are utilized for training the CNN model. This big part of the dataset makes sure that a lot of diverse examples are seen by the model so that it learned in detail about both healthy and unhealthy heart sounds. Out of the total, 3240 images 2575 images are healthy and 665 images are unhealthy.
- **Validation Data:** The remaining 301 images form a validation set that will be very important for model performance evaluation. This subset is essential for ensuring that the model generalizes effectively to new, unseen data. The validation data has 301 images. Out of these 150 images are healthy and 151 images are unhealthy.

C. Preprocessing

Preprocessing is a vital step in machine learning and deep learning workflows, particularly when dealing with raw data. In the context of your project involving Phonocardiogram (PCG) signals and Gammatonegram images, preprocessing ensures that the data is clean, structured, and ready for input into a model, improving its performance and accuracy. The preprocessing techniques involved are normalization, resizing, data augmentation.

Normalization:

Normalization is a technique which is used to scale data into a standard range; often, a range of 0 to 1 or -1 to 1 is considered standard. Normalized data can help the model converge faster during training, as the optimization algorithms (like gradient descent) perform better when the input features are on a similar scale.

Resizing:

Resizing is the process of the altering the dimensions of the images to match the input size required by the neural network. CNNs typically require fixed-size inputs, so all images must be resized to ensure uniformity. Thus, the input images are resized to 150×150 and fed to the CNN model.

Augmentation:

Data augmentation is used to increase the size of a training dataset by altering the original images in some different ways. In this way, the model becomes powerful enough, and its performance increases when few examples are available in the dataset. Various classic transformations would involve random rotations, flipping, shifts, zooms, and cropping. Random horizontal flip is used on 3541 images because the dataset has few images. This technique is used to increase images in multiples of 2.

D. Proposed Workflow

The suggested model uses a Convolutional Neural Network (CNN) design to find problems in Phonocardiogram (PCG) signals by analyzing them with Gammatonegram. The process starts with the input PCG signals, which are changed using Gammatonegram computation to create time-frequency representations needed for capturing important features. Before feeding these representations into the CNN, several preprocessing techniques are applied. These include normalization to scale the data within a specific range, resizing the Gammatonegram images to a consistent dimension, and data augmentation to enhance the model's robustness by artificially expanding the training dataset through techniques like rotation, scaling, and flipping. Additionally, class weights are adjusted to address class imbalance, ensuring the model pays equal attention to underrepresented categories.

The CNN architecture is composed of three parallel processing streams. Each stream is initiated with a Convolutional layer that extracts out the spatial features. Then, there is Batch Normalization to make learning more stable, with subsequent use of the Rectified Linear Unit (RELU) as an activation function to introduce non-linearity. Such outputs from these branches are combined into a central block consisting of several layers

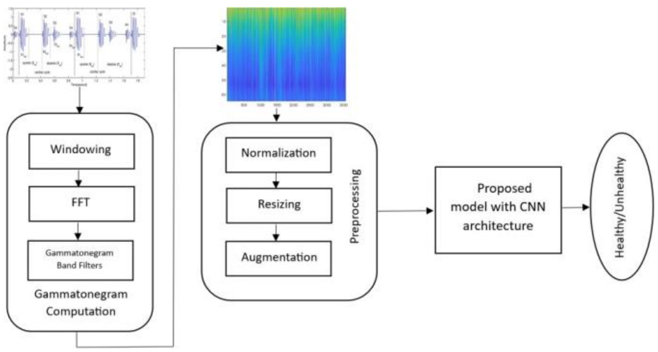


Fig. 3: Flowchart of CNN-Driven Detection of Abnormalities in PCG Signals using Gammatonegram Analysis.

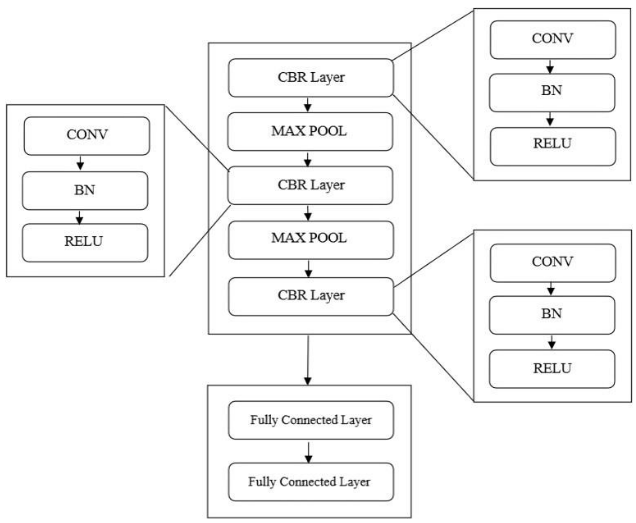


Fig. 4: Proposed model with CNN architecture.

of Convolution-BatchNorm-ReLU (CBR) combined with Max Pooling layers mixed in. Smaller feature maps are obtained by using the Max Pooling layers, which reduce the size of the feature maps without losing important patterns. The CBR layers are repeated in order to enhance feature extraction and allow the model to better capture small details in the signals. After passing through these convolutional layers, the features are flattened and passed as input to two fully connected layers. These layers serve as the model's decision-making components, transforming the extracted features into classification outputs. The final layer outputs a decision indicating whether the input PCG signal corresponds to a healthy or unhealthy heart condition. To optimize the performance of the model, several optimizers are used to adjust the model weights by minimizing the loss function during training.

E. Model

The convolutional neural network is one of the deep learning methods where the feature choice has changed with model learning. A neural network usually consists of three layers they are input layer, hidden layers, and output layer.

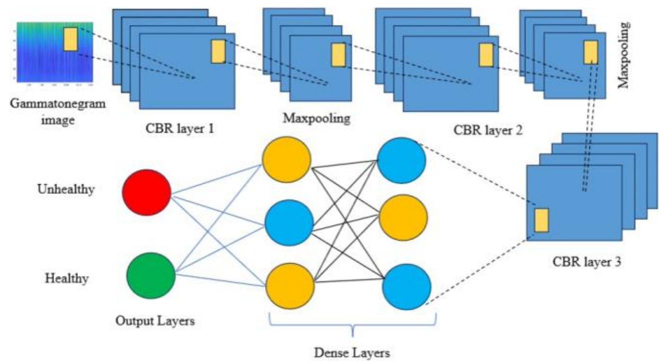


Fig. 5: Proposed CNN model.

As shown in Fig. 5. The proposal has been subjected to various layers of Conv2D, with the first layer having 12 filters, followed by 20 and 32 in the later stages. Each convolutional layer is followed by batch normalization and ReLU activation for normalization and introducing non-linearity for stabilized learning and modeling complex relations. Max-pooling with a filter size of 2x2 was used after the first two convolutional layers, reducing the feature maps from 150x150 to 75x75 and subsequently to 37x37. This retains only the most important elements of the feature maps. Deeper convolutional layers, represented by CBR layers 2 and 3, capture even more complex and high-level patterns. Additional filters were added to detect new patterns while reducing the image size in. It then flattens into a single line of 43,888 features and goes through fully connected layers (dense), which are 256, 128, and 2 in size. Finally, the sounds are classified as ”Healthy” or ”Unhealthy.” This gradually increased filter size and decreased detail help the CNN systematically gather important features from the input Gammatonegram images, thereby coming out with precise heart sound classification.

F. Optimizers

Optimizers, in general, play an important role in training neural networks, including CNNs. They adapt the model's weights in a manner that minimizes the loss function of the model, therefore making it perform better. Optimizers modify the model's parameters-weights-after the gradients are computed during backpropagation. The idea is to converge on an optimal set of parameters that minimize the loss function, hence yielding the optimal performance of the model. How this set of changes is accomplished may vary with the optimizer, which then reflects on the speed and manner in which the model will learn. The choice of optimizer can greatly influ- ence convergence speed, escaping local minimum, and overall performance of a model. The optimizers used in the proposed model are SGD, Adagrad, Adadelta, RMSProp, and Adam. More information about these optimizers is discussed below.

*1) **Stochastic Gradient Descent (SGD)**:* Stochastic Gradient Descent changes the model weights based on the gradient of the loss concerning each parameter. Contrary to regular Gradient Descent, which uses the whole dataset to calculate the gradient, in SGD, updates of weights are made for every training example or small groups of them. This makes it quicker and, moreover, far better when big datasets are taken into consideration.

*2) **Adagrad (Adaptive Gradient Algorithm)**:* Adagrad updates the learning rate independently for every parameter. It makes the learning rate smaller by the magnitude of past gradients. As a result, parameters with large gradients get smaller updates while parameters with small gradients get larger updates. This can be useful when dealing with sparse data.

*3) **Adadelta**:* Adadelta can be seen as a variant of Ada- grad that tries to solve the problem of decreasing learning rate. Rather than accumulating all formerly squared gradients, Adadelta restricts the accumulation of gradients to a fixed size. This allows the learning rate to remain adaptive over time.

*4) **RMSProp (Root Mean Squared Propagation)**:* RMSProp does the same thing as Adagrad but it uses an exponentially decaying average over the squared gradients, instead of accumulating all the squared gradients. This keeps the learning rate manageable and prevents it from shrinking too much.

*5) **Adam (Adaptive Moment Estimation)**:* Adam is one of the most popular optimizers, combining the advantages of both RMSProp and Momentum (another optimization technique that accelerates SGD). It computes adaptive learning rates for each parameter, and it includes momentum by keeping a running average of both the gradients and their second moments.

The optimizer parameters are

Number of Epochs: The algorithm is executed multiple times on the training dataset. In this case, we train the model for 25 epochs.

Batch Size: Batch size is among the most important hyperparameters in the training of neural networks. It defines the number of examples that are used in one model iteration or one forward and backward pass of the model. Batch Size gives the number of samples to update the model parameters. In this model, the batch size used is 32.

Learning rate: Learning rate is the scaling factor that is needed for updating the model weights. The proposed model uses different optimizers with different learning rates. It is a hyperparameter, which has a greater influence on both the improvement of training and activity of model which is based on the assessment metrics.

Loss Function: A loss function in general will measure how well your prediction model is doing with respect to predict the expected output. The loss used within this example model is cross entropy loss.

Class Weights: Class weights are used to handle imbalanced datasets where certain classes have significantly more samples than others. Applying class weights ensures that the model pays equal attention to all classes, preventing bias towards the majority class. The dataset used for this study consists of 3541 Gammatonegram images, separated into training and testing sets with a ratio of approximately 9:1.So class weights play an important role to help the model pay equal attention to all classes and preventing bias towards the majority class.

The below table shows the learning rates of optimizers.

TABLE I: Learning rates of optimizers

Optimizers	**Learning rates**
SGD	0.001
Adagrad	0.01
Adadelta	0.01
RMSProp	0.001
Adam	0.0001

IV. Result and Discussion

The Gammatonegram images derived from the raw PCG signals were processed through a CNN model, with the accuracy evaluated using different optimizers. The model was trained for 25 epochs, using a batch size of 32, and the training and testing performance was compared across various optimizers. The Adam optimizer showed exceptional results, achieving a test accuracy of 100% by the 25th epoch, with a learning rate of 0.0001.The below figures show the training and testing accuracies of each of the optimizers.

Fig 6 represents the graphs for the SGD optimizer, which reached the maximum training accuracy of 95.03% with a test accuracy of 89.70%. Fig. 7 represents the graph for the Adagrad optimizer, which reached the maximum training accuracy of 97.56% with a test accuracy of 96.01%. Fig. 8 Results with Adadelta optimizer shows that training accuracy is up to 96.76% and test accuracy reaches 92.69%. Fig. 9 Maximum of 91.98% on training accuracy and maximum of 82.39% on test accuracy achieved by RMSProp.Fig. 10 The Adam optimizer's performance is represented, which reached the highest training accuracy of 100.00% and best test accuracy of 100%.

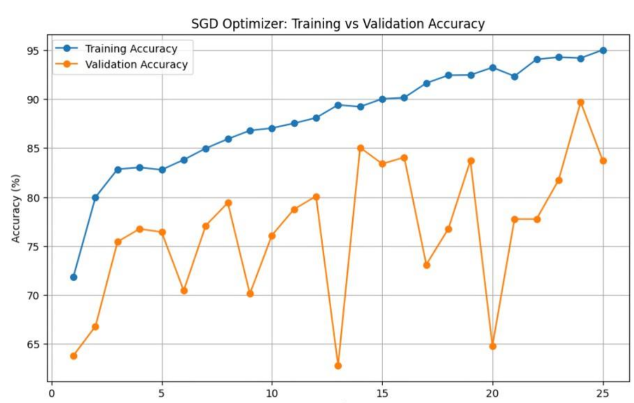


Fig. 6: Differentiation of training and testing accuracy with SGD optimizer

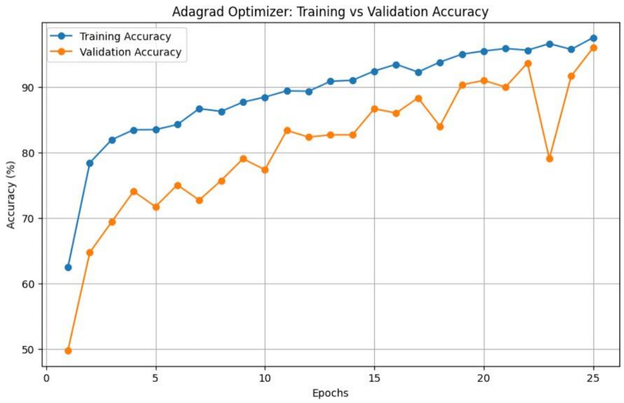


Fig. 7: Differentiation of training and testing accuracy with Adagrad optimizer

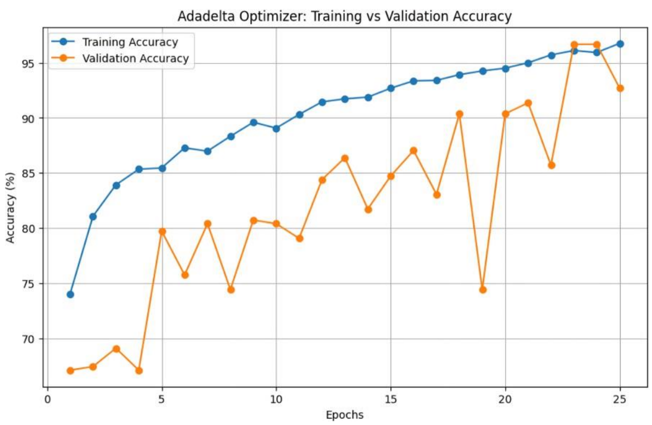


Fig. 8: Differentiation of training and testing accuracy with Adadelta optimizer

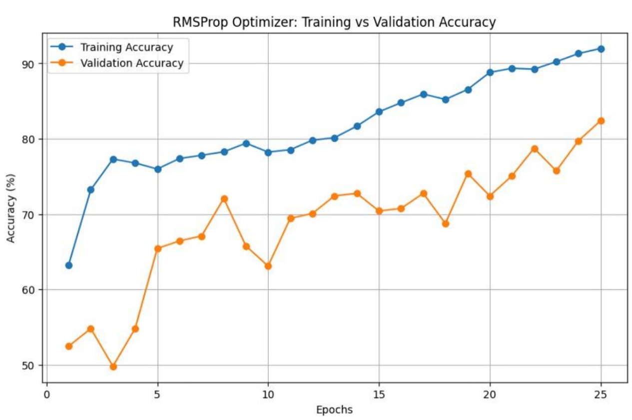


Fig. 9: Differentiation of training and testing accuracy with RMSProp optimizer

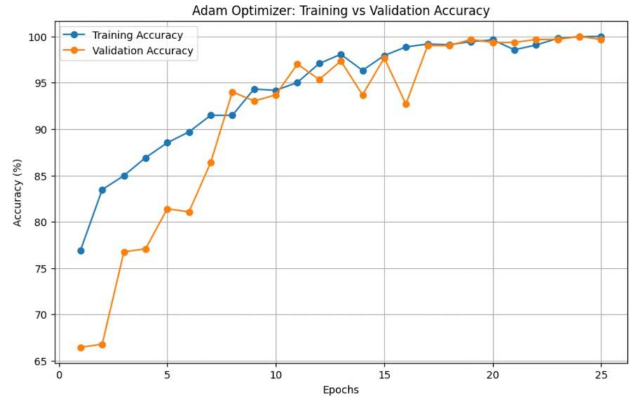


Fig. 10: Differentiation of training and testing accuracy with Adam optimizer

The comparison of training and testing accuracy for all optimizers is represented in below figure.

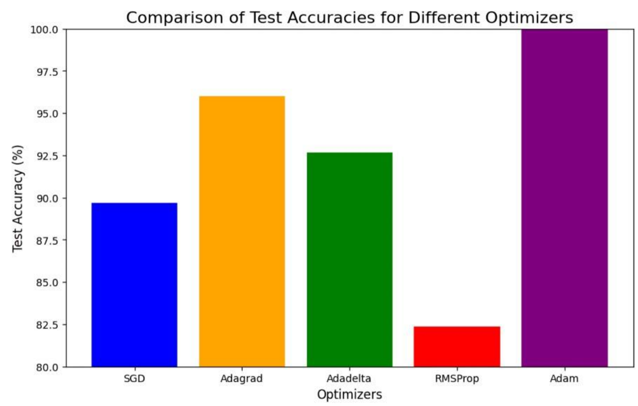


Fig. 11: Comparison of test accuracies across various optimizers

The above Fig. 11 compares the maximum test accuracies of all optimizers. The Adam optimizer achieved the highest training accuracy of 100%. The Adagrad and Adadelta optimizers also performed well, with test accuracies of 96.01% and 92.69%, respectively. In contrast, the SGD and RMSProp optimizers exhibited lower performance, with test accuracies of 89.70% and 82.39%. These results underscore the superior effectiveness of the Adam optimizer for the task of PCG signal classification using CNNs. Overall, the Adam optimizer proved to be the most efficient, yielding the highest accuracy and faster convergence, while other optimizers like Adagrad and Adadelta delivered moderate performance.

From the below table II , we can observe that various advanced techniques have been employed to significantly improve heart sound classification. The study by ASHINIKUMAR SINGH et al. applied ensemble learning with CNN models on the PhysioNet 2016 dataset, obtained accuracy of 99.51% [11]. Another study focused on heart sound classification using Gammatonegram images and texture-based

TABLE II: Accuracy comparsion for different models

S.No	Study & author	Models/Techniques	Accuracy
1	ASHINIKUMAR SINGH et al. Ensemble Learning for Accurate Prediction	CNN, Ensemble Learning	99.51%
2	Heart sound classification method using gammatonegram	CNN, Texture Feature Extraction	94%
3	Transfer Learning Research	YAMNet-based Transfer Learning (TL)	92.23% (PhysioNet), 99.83 (HVD dataset)
4	LVDD Detection using CatBoost	Pre-trained CNNs (VGG16, Xception, ResNet50), CatBoost with LDA	91.1%, 88.2%, 82.1%, 92.7%
5	Classification of Phonocardiogram Signals Using the Wavelet Scattering Transform and Equilibrium Optimization Approach.	K-nearest neighbour (KNN)	99.5%
6	Gammatonegram image classification	Proposed CNN model	100%

feature extraction methods like Linear Ternary Pattern (LTP) and Local Phase Quantization (LPQ), also on the PhysioNet CinC 2016 dataset. This approach reached an accuracy of 94% [12]. A study fine-tuned YAMNet-based pre-trained models using time-frequency representations, including spec- trograms and scalograms, on the PhysioNet/CinC Challenge 2016 dataset. This achieved accuracy of 92.23% on the Phy- sioNet dataset and improved accuracy 99.83% on the HVD dataset [13]. A study on detecting left ventricular diastolic dysfunction (LVDD) used pre-trained CNNs, such as VGG16, Xception, and ResNet50, combined with CatBoost classifiers. This approach resulted 88.2% accuracy [14]. Cross Wavelet Transform with the AlexNet model on the PhysioNet dataset resulted in an accuracy of 98% [15]. Finally, a CNN model using Gammatonegram images was applied to the PhysioNet Challenge 2016 dataset, where it achieved a perfect accuracy of 100%.

V. Conclusion

In conclusion, the proposed method using Gammatonegram-based analysis with CNN offers a simple yet effective approach for detecting abnormalities in PCG signals. By transforming PCG signals into Gammatonegram images, and applying preprocessing techniques like resizing and random horizontal flip augmentation, the model can accurately classify healthy and unhealthy heart sounds. Among the optimizers tested, the Adam optimizer achieved the highest accuracy of 100%, demonstrating its effectiveness. This study confirms that Gammatonegram representations are highly suitable for the classification of PCG signals, offering a reliable method for heart disease detection.

References

[1] Cheung, Ka Wang, M. Bonet, K. A. Frank, O. T. Oladapo, G. J. Hofmeyr, WHO Intrapartum Care Algorithms Working Group, Livia Ciabati et al. ”Clinical algorithms for management of fetal heart rate abnormalities during labour.” BJOG: An International Journal of Obstetrics & Gynaecology 131 (2024): 102-111.

[2] Tunnell, Natalie C., Sarah E. Corner, Andres D. Roque, Juliet L. Kroll, Thomas Ritz, and Alicia E. Meuret. ”Biobehavioral approach to distinguishing panic symptoms from medical illness.” Frontiers in Psychiatry 15 (2024): 1296569.

[3] Duca, S¸ tefania-Teodora, Ionu¸t Tudorancea, Radu-S¸ tefan Miftode, Adriana Chetran, Alexandru-Dan Costache, D. Robert Pava˘l, and Irina-Iuliana Costache-Enache. ”Heartbeat Chronicles: Decoding the Interplay of Echocardiography and Heart Rate Variability in Chronic Heart Failure Patients–Unraveling the Mysteries with Traditional and Advanced 24-Hour Holter ECG Parameters.” Romanian Journal of Cardiology.

[4] Lee, Joon-Hyop, EunKyung Ku, Yoo Seung Chung, Young Jae Kim, and Kwang Gi Kim. ”Intraoperative detection of parathyroid glands using artificial intelligence: optimizing medical image training with data augmentation methods.” Surgical Endoscopy 38, no. 10 (2024): 5732-5745.

[5] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease Prediction International Journal of Recent Technology and Engineering Vol - 7, No 6, Mar - 2019 ISSN - 2277-3878, Pages – 1754 – 1772.

[6] Lee, T. H., E. F. Shair, A. R. Abdullah, K. A. Rahman, and N. Nazmi. ”Comparison of Short Fast Fourier Transform and Continuous Wavelet Transform in Study of Stride Interval.”

[7] M.Sireesha, Srikanth Vemuru, S.N.Tirumala Rao “Classification Model for Prediction of Heart Disease Using Correlation Coefficient Technique” International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 2, March - April 2020, Pages- 2116 – 2123.

[8] Chambi, Edwin M., Jefry Cuela, Milagros Zegarra, Erasmo Sulla, and Jorge Rendulich. ”Benchmarking Time-Frequency Representations of Phonocardiogram Signals for Classification of Valvular Heart Diseases Using Deep Features and Machine Learning.” Electronics 13, no. 15 (2024): 2912.

[9] Arslan, Muhammad, Muhammad Mubeen, Muhammad Bilal, and Saadullah Farooq Abbasi. ”1D-CNN-IDS: 1D CNN-based Intrusion Detection System for IIoT.” arXiv preprint arXiv:2409.08529 (2024).

[10] Ahmad, Ahsan, Ali Husnain, Ashish Shiwlani, Ahad Hussain, Mahnoor N. Gondal, and Ayesha Saeed. ”Ethical and clinical implications of AI integration in cardiovascular healthcare.” World Journal of Advanced Research and Reviews 23, no. 3 (2024): 2479-2501.

[11] ASHINIKUMAR SINGH, S. I. N. A. M., S. I. N. A. M. Ajitkumar SINGH, and A. H. E. I. B. A. M. DINAMANI SINGH. ”Ensemble Learning for Accurate Prediction of Heart Sounds using Gammatonegram Images.” Turkish Journal of Electrical Engineering and Computer Sciences 32, no. 4 (2024): 555-573.

[12] Taneja, Kriti, Vinay Arora, and Karun Verma. ”Heart sound classification method using gammatonegram and SVM.” Multimedia Tools and Applications (2024): 1-37.

[13] Maity, Arnab, Akanksha Pathak, and Goutam Saha. ”Transfer learning-based heart valve disease classification from Phonocardiogram signal.” Biomedical Signal Processing and Control 85 (2023): 104805.

[14] Zheng, Yineng, Xingming Guo, Yang Yang, Hui Wang, Kangla Liao, and Jian Qin. ”Phonocardiogram transfer learning-based CatBoost model for diastolic dysfunction identification using multiple domain-specific deep feature fusion.” Computers in Biology and Medicine 156 (2023): 106707.

[15] Ismael, Mustafa R., Haider J. Abd, and Raad Z. Homod. ”Classification of Phonocardiogram Signals Using the Wavelet Scattering Transform and Equilibrium Optimization Approach.” (2024).

[16] For dataset refer this link : https://www.kaggle.com/datasets/brundavanamsatyasai/gammatone-images-of-heart-sounds

bb2

ORIGINALITY REPORT

12%	4%	10%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Brundavanam Satyasai, Rajeev Sharma, Mohan Bansal. "A Gammatonegram based Abnormality Detection in PCG Signals using CNN", 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP), 2023 Publication	1%
2	Arnab Maity, Akanksha Pathak, Goutam Saha. "Transfer learning based heart valve disease classification from Phonocardiogram signal", Biomedical Signal Processing and Control, 2023 Publication	1%
3	Singh, Shekhar. "Facial Expression Recognition Using Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) for Data Augmentation and Image Generation", University of Nevada, Las Vegas, 2024 Publication	1%

4 Yineng Zheng, Xingming Guo, Yang Yang, Hui Wang, Kangla Liao, Jian Qin. "Phonocardiogram transfer learning-based CatBoost model for diastolic dysfunction identification using multiple domain-specific deep feature fusion", Computers in Biology and Medicine, 2023 — 1%

Publication

5 "Electrocardiogram Classification for Arrhythmia using Convolutional Neural Network 2D and Adabound Optimizer", International Journal of Recent Technology and Engineering, 2020 — 1%

Publication

6 Wrigglesworth, Alan. "Credit Scoring Using Machine Learning: An Application of Deep Learning", University of Pretoria (South Africa), 2023 — 1%

Publication

7 Edvaldo Domingos, Blessing Ojeme, Olawande Daramola. "Experimental Analysis of Hyperparameters for Deep Learning-Based Churn Prediction in the Banking Sector", Computation, 2021 — <1%

Publication

8 docshare.tips — <1%

Internet Source

9	Murside Degirmenci, Mehmet Akif Ozdemir, Elif Izci, Aydin Akan. "Arrhythmic Heartbeat Classification Using 2D Convolutional Neural Networks", Research Square Platform LLC, 2020 Publication	<1%
10	Arnab Maity, Goutam Saha. "Time-Frequency Fragment Selection for Disease Detection from Imbalanced Phonocardiogram Data", 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2023 Publication	<1%
11	Submitted to University of Maryland, Global Campus Student Paper	<1%
12	Submitted to Altinbas University Student Paper	<1%
13	Submitted to Brunel University Student Paper	<1%
14	www.ncbi.nlm.nih.gov Internet Source	<1%
15	Kriti Taneja, Vinay Arora, Karun Verma. "Heart sound classification method using gammatonegram and SVM", Multimedia Tools and Applications, 2024 Publication	<1%

16 Sukanya P., Srinivasa B. Rao. "Stacked Denoising Autoencoder Based Parkinson’s Disease Classification using Improved Pigeon-Inspired Optimization Algorithm", Computer Science, 2024 <1%
Publication

---

17 link.springer.com <1%
Internet Source

---

18 smartech.gatech.edu <1%
Internet Source

---

19 Palani Thanaraj Krishnan, Parvathavarthini Balasubramanian, Snekhalatha Umapathy. "Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network", Physical and Engineering Sciences in Medicine, 2020 <1%
Publication

---

20 my.clevelandclinic.org <1%
Internet Source

---

21 usir.salford.ac.uk <1%
Internet Source

---

22 www.pubs2.ascee.org <1%
Internet Source

---

23 Shrunoti Deshmukh, Ajay Rathod, Harshal Sonawane, Roshani Raut, Anita Devkar. "Skin Cancer Classification Using CNN", 2023 <1%

International Conference on Applied Intelligence and Sustainable Computing (ICAISC), 2023
Publication

24	export.arxiv.org Internet Source	<1%
25	www.diva-portal.org Internet Source	<1%
26	www.mdpi.com Internet Source	<1%
27	Abhranta Panigrahi, Abdulhamit Subasi. "Magnetic resonance imagining-based automated brain tumor detection using deep learning techniques", Elsevier BV, 2023 Publication	<1%
28	Alaa M. Adel El-Shazli, Sherin M. Youssef, Abdel Hamid Soliman. "Intelligent Computer-Aided Model for Efficient Diagnosis of Digital Breast Tomosynthesis 3D Imaging Using Deep Learning", Applied Sciences, 2022 Publication	<1%
29	Gurjot Singh, Abhinav Verma, Lavanya Gupta, Anant Mehta, Vinay Arora. "An automated diagnosis model for classifying cardiac abnormality utilizing deep neural networks", Multimedia Tools and Applications, 2023 Publication	<1%

30	Melinda Melinda, Zharifah Muthiah, Fitri Arnia, Elizar Elizar, Muhammad Irhmasyah. "Image Data Acquisition and Classification of Vannamei Shrimp Cultivation Results Based on Deep Learning", MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, 2024 Publication	<1%
31	Oge Marques. "AI for Radiology", CRC Press, 2024 Publication	<1%
32	Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023 Publication	<1%
33	Vykintas Maknickas, Algirdas Maknickas. "Recognition of normal–abnormal phonocardiographic signals using deep convolutional neural networks and mel-frequency spectral coefficients", Physiological Measurement, 2017 Publication	<1%
34	Zishen Li, Yi Chang, Bjorn W. Schuller. "CNN-Based Heart Sound Classification with an Imbalance-Compensating Weighted Loss Function", 2022 44th Annual International	<1%