Arrhythmia Detection Using Deep Learning

Prof. Rudresh H M[1], Pavithra M U[2], Nisha S[3], Meghana H G[4], Preethi M L[5], Dr. Rajashekar K J[6]

[1,2,3,4,5,6] *Department of Information Science and Engineering, Kalpataru Institute of Technology, Tiptur*

doi.org/10.64643/IJIRTV12I7-188820-459

Abstract—**Arrhythmias, or irregular heart rhythms, pose serious health risks and require timely detection to prevent severe complications. Traditional diagnostic methods such as manual electrocardiogram (ECG) interpretation can be slow and susceptible to human error. This study proposes a deep learning–based arrhythmia detection system designed to improve diagnostic accuracy and efficiency. A Kaggle dataset is preprocessed and augmented to enhance quality and address class imbalance, followed by an 80:10:10 split for training, validation, and testing. Multiple convolutional neural network (CNN) architectures—including ResNet50, Efficient Net, VGG16, VGG19, and a hybrid ResNet50-EfficientNet model—are trained and optimized through hyperparameter tuning and augmentation strategies. The best-performing models are integrated into a user-friendly web application built with HTML/CSS for the frontend and Flask for the backend. The system enables users to upload medical images, receive automated arrhythmia predictions, and access information on possible treatments and precautions. Overall, the project highlights the capability of deep learning to automate and enhance arrhythmia diagnosis, offering a scalable and accessible tool that supports clinical decision-making and improves healthcare outcomes.**

Index Terms—**Arrhythmia Detection, ECG Signals, Deep Neural Networks, Transfer Learning, Hybrid Feature Extraction, Medical Image Processing.**

I. INTRODUCTION

Cardiovascular diseases (CVDs) continue to be the leading cause of morbidity and mortality worldwide, accounting for millions of deaths each year. Among the wide range of CVDs, arrhythmias—disorders characterized by irregular heart rhythms—are particularly prevalent. These conditions can vary from harmless irregularities to severe, life-threatening abnormalities. If left undetected or untreated, arrhythmias may lead to serious complications such as stroke, heart failure, cardiac arrest, or even death. Therefore, early and accurate detection of arrhythmias is essential to reduce associated risks and improve patient outcomes.

Electrocardiograms (ECGs) are the gold standard for monitoring and diagnosing cardiac rhythm disorders. However, manual interpretation of ECG data requires expert cardiologists and can be both time-consuming and prone to human error. Additionally, the availability of skilled professionals is often limited, especially in resource-constrained healthcare settings. These challenges highlight the urgent need for automated diagnostic tools that can aid clinicians by providing fast, accurate, and consistent analysis of ECG signals.

project proposes a robust and scalable system for classifying cardiovascular diseases using ECG images.

The primary objective of this work is to perform a comparative analysis of two popular deep learning architectures: Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNNs are highly effective in extracting spatial features from images, while RNNs are designed to handle sequential data. This study evaluates their performance to determine which model is better suited for automated ECG-based cardiovascular disease classification.

In summary, this project addresses the critical global health challenge of cardiovascular disease diagnosis by integrating cutting-edge AI with medical imaging, offering an efficient and accessible solution to improve patient care worldwide.

II. LITERATURE SURVEY

A. Early methods used manual ECG feature extraction for arrhythmia detection. Wilson et al. (2010), in “A Comparative Study of Arrhythmia Detection Using Rule-Based and Statistical Techniques”, relied on predefined thresholds for heart rate and waveform characteristics. These techniques often struggled with noisy or complex ECG patterns.

B. Early methods used manual ECG feature extraction for arrhythmia detection. Wilson et al. (2010), in “A Comparative Study of Arrhythmia Detection Using Rule-Based and Statistical Techniques”, relied on predefined thresholds for heart rate and waveform characteristics. These techniques often struggled with noisy or complex ECG patterns.

C. Machine-learning models (SVM, KNN) improved accuracy but required heavy preprocessing. Zhang et al. (2015), in “Classification of Arrhythmias Using Machine Learning Techniques”, showed that feature engineering was critical, and these models had limited scalability for large datasets.

D. Deep learning enabled automatic feature learning from raw ECG signals. Hannun et al. (2018), in “Convolutional Neural Networks for Automated Arrhythmia Detection”, demonstrated that hierarchical features could be learned automatically, improving detection consistency.

E. CNN-based models showed strong performance in identifying abnormal heartbeats. Rajpurkar et al. (2017), in “Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks”, showed CNNs effectively captured spatial patterns in ECG images and spectrograms.

F. LSTM networks improved detection by capturing temporal ECG patterns. Faust et al. (2018), in “ECG-Based Arrhythmia Classification Using Long Short-Term Memory Networks”, highlighted their strength in modeling sequential dependencies in long ECG time-series data.

G. Hybrid CNN–LSTM models further enhanced classification accuracy. Yildirim et al. (2020), in “Hybrid Deep Learning Framework for Arrhythmia Detection Using ECG Data”, combined CNNs for spatial features and LSTMs for temporal dynamics, achieving higher performance.

H. Recent works use large ECG datasets to build more robust arrhythmia detection systems. Chen et al. (2021), in “Transfer Learning for ECG-Based Arrhythmia Detection Using Pretrained Models”, showed that transfer learning and data augmentation improved generalization.

I. Attention-based deep learning models have been applied to highlight important ECG segments during classification. Kumar et al. (2022), in “Real-Time Arrhythmia Detection Using Deep Learning and Web-Based Interfaces”, explored web-based deployment to enhance accessibility and reliability.

III. SYSTEM DESIGN

A Data Flow Diagram (DFD) is a graphical tool used to represent how data moves through a system. It shows processes**,** data stores**,** data flows**,** and external entities**.** DFDs are commonly used in software engineering, system analysis, and database design.

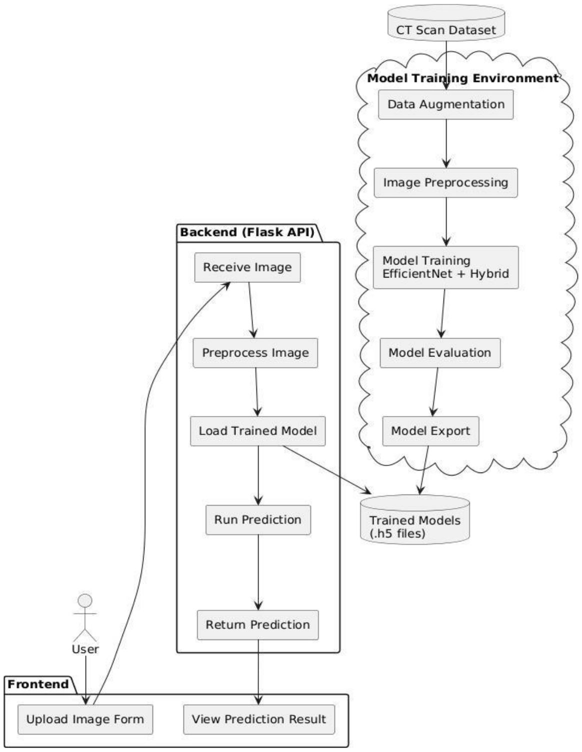


FIG 3.1: Data Flow

CT scan images are processed and used to train a prediction model. The trained model is loaded in the Flask backend, which handles image input from the user. It generates a prediction and sends the result to the frontend, where the user can view it.

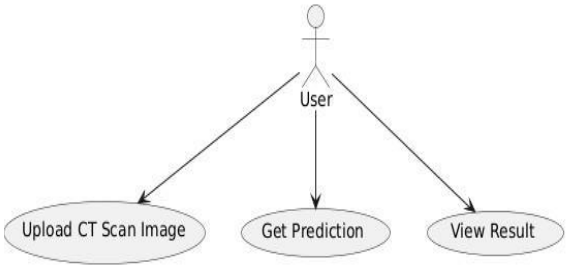


FIG 3.2: User Interface

The user interface allows the user to upload a CT scan image, request a prediction, and then view the result. Through these simple steps, the system provides an easy and interactive way for users to access the model's output.

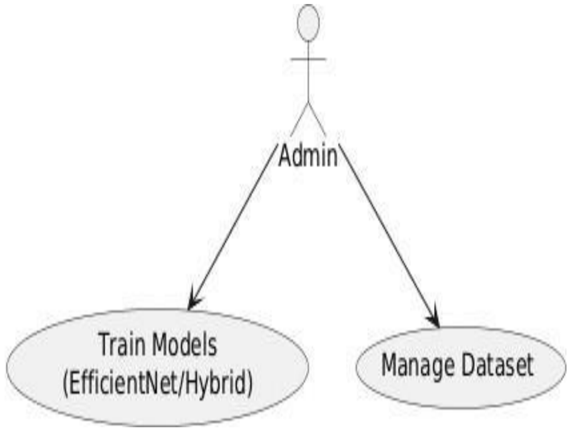


FIG 3.3: Use Case Diagram

The use case diagram shows the admin's main responsibilities in the system. The admin manages the dataset and trains the prediction models, including EfficientNet and the hybrid model. These actions ensure that the system stays updated, accurate, and ready for user predictions.

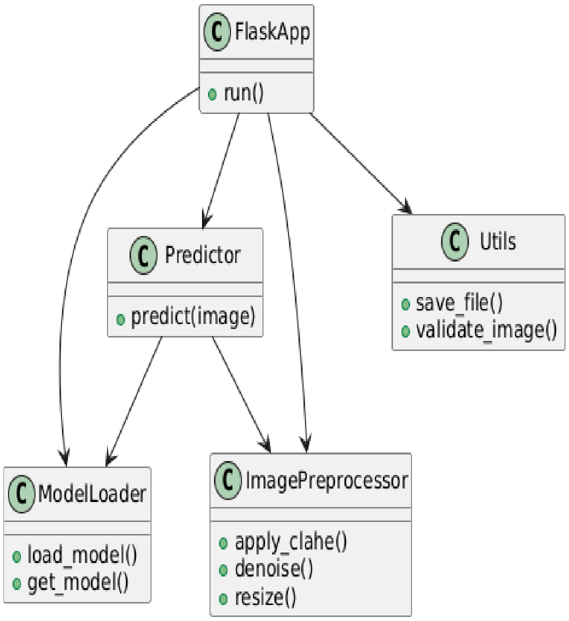


FIG 3.4: Class Diagram

The class diagram illustrates the structure of the system, where the Flask App class manages the application flow. The Predictor class handles prediction tasks using support from Model Loader for loading models and Image Preprocessor for preprocessing images. The Utils class provides essential functions such as file saving and image validation, making the system modular and easy to maintain.

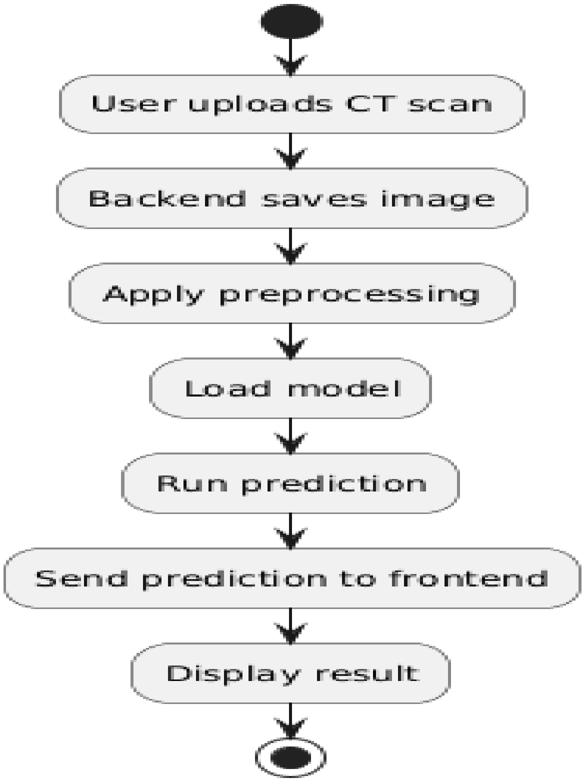


FIG 3.5: Activity Diagram

The activity diagram shows the step-by-step process of generating a prediction. The user first uploads a CT scan, which the backend saves and preprocesses. The system then loads the trained model, runs the prediction, and sends the result to the frontend. Finally, the user sees the displayed output.

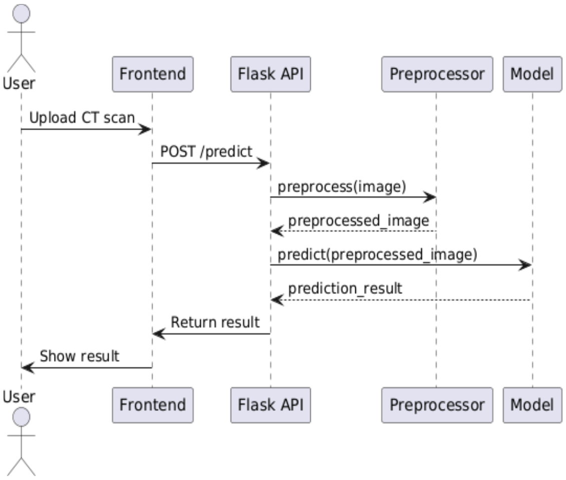


FIG 3.6: Sequence Diagram

The sequence diagram shows how the system processes a CT scan image step-by-step. The user uploads the image through the frontend, which sends it to the Flask API. The backend preprocesses the image and passes it to the model for prediction. The prediction result is then returned to the frontend, where it is displayed to the user.

V. METHODOLOGY

ECG data were collected from publicly available databases, such as MIT-BIH Arrhythmia, including anonymized records of atrial fibrillation, ventricular tachycardia, and bradycardia, with raw signals and cardiologist-labeled annotations. Preprocessing removed noise and artifacts using band-pass filtering and wavelet transforms, followed by segmentation into individual heartbeats or fixed-length windows. Features included time-domain (heart rate, QRS duration), frequency-domain (FFT coefficients), and morphological characteristics.

For modeling, traditional machine learning algorithms (SVM, Random Forest, k-NN) were trained on extracted features, while deep learning models (CNN, LSTM, and hybrid CNN–LSTM) were trained end-to-end on raw or minimally preprocessed signals. Cross-validation was used to improve robustness and reduce overfitting.

The system ran on Intel Core i5/i7 or AMD Ryzen processors, with optional NVIDIA GPUs for deep learning. Software included Python 3.8+, NumPy, Pandas, Scikit-learn, TensorFlow/Keras, PyTorch, and ECG tools like BioSPPy and WFDB. Deployment utilized Flask or FastAPI, with optional dashboards via Streamlit/Dash and mobile apps via Android Studio or Flutter. Git/GitHub ensured version control.

A. CNN Algorithm Code

```
import os
import numpy as np import shutil
from tqdm import tqdm
from tensorflow.keras.preprocessing.image
import ImageDataGenerator, load_img, img_to_array, array_to_img
# Corrected dataset paths input_dataset_paths = {
#F:Fusion beat
"F": "ecg_img/train/F/F", # N:Normal beat
"N": "ecg_img/train/N/N", # Q:Unknown
"Q": "ecg_img/train/Q/Q", # S:Superventricular
"S": "ecg_img/train/S/S", # V:Ventricular
202 2025-26
Dept. of ISE, KIT, Tiptur 19
"V": "ecg_img/train/V/V"
}
output_dataset = "Dataset2"
# Ensure the output dataset directory exists if os.path.exists(output_dataset):
shutil.rmtree(output_dataset) # Remove old dataset if it exists
os.makedirs(output_dataset)
# Define augmentation parameters datagen = ImageDataGenerator( rotation_range=15, zoom_range=0.01, horizontal_flip=True
)
# Process each class folder
for class_name, class path in tqdm(input_dataset_paths.items(), desc="Processing Classes"): if not os.path.exists(class_path):
print(f" Skipping {class_name}, folder not found.") continue
# Create output directory for this class
output_class_path = os.path.join(output_dataset, class_name) os.makedirs(output_class_path, exist_ok=True
# Get all image file paths
images = [os. path.join(class_path, img) for img in os. listdir(class_path) if img. endswith(('png', 'jpg', 'jpeg'))]
```

VI. RESULTS AND DISCUSSION

This study evaluates the performance of CNN and RNN models for classifying cardiovascular diseases from ECG images, considering accuracy, precision,
recall, F1-score, and computational efficiency. ECG images were collected, resized to 100×100 pixels, normalized, and augmented through rotations, zooms, and horizontal flips to balance classes and improve model generalization. The CNN model consisted of convolutional, pooling, and dense layers with ReLU activations, while the RNN model employed LSTM layers to capture temporal dependencies. Both models were trained and tested on the same dataset, and a user interface was developed to allow live ECG image predictions using the trained CNN model. figures.

A. Results

ARYTHMIA DETECTION USING DEEPLEARNING

Upload Clear Images

FIG 4.1: Frontend Page

It shows the frontend page of the Arrhythmia Detection Using Deep Learning system. This interface serves as the entry point for users, providing an intuitive and visually informative layout. It displays essential visual elements related to cardiac monitoring and sets the foundation for interacting with the system's arrhythmia detection features.

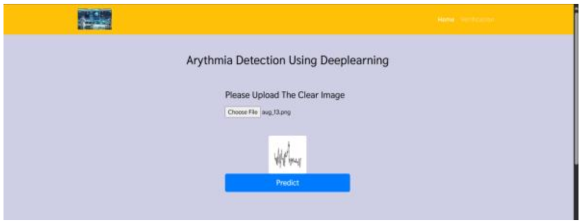
Our Services

ARYTHMIA DETECTION USING DEEPLEARNING

We Can Able to Classify The Arythmia.

FIG 4.2: Frontend Page

Illustrates the services section of the Arrhythmia Detection Using Deep Learning system. This page highlights the core functionality of the application, showcasing how the system supports arrhythmia classification through advanced deep learning models. The layout provides users with an overview of the available service and guides them toward further interaction with the system features.



Arythmia Detection Using Deeplearning

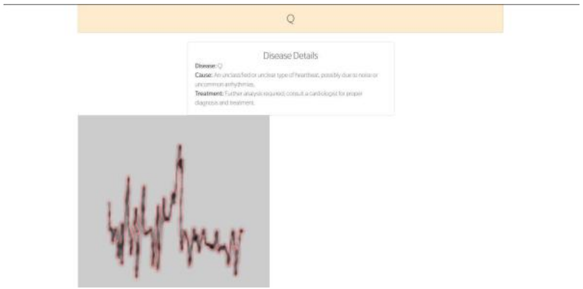
Please Upload The Clear Image

Choose File aug_13.png

Predict

FIG 4.3: Image upload Page

shows the image upload page of the Arrhythmia Detection Using Deep Learning system. This interface allows users to upload ECG images, which are then processed by the model to identify potential arrhythmias. The page is designed to be simple and user-friendly, ensuring smooth interaction and efficient submission of input data for analysis.



Q

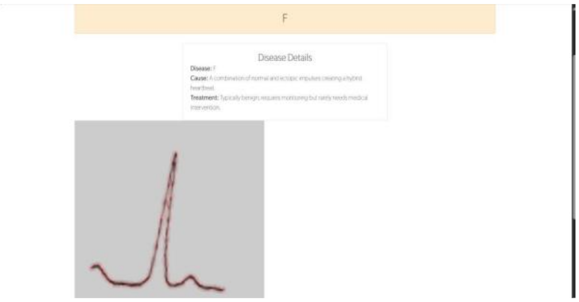
Disease Details

Disease: Q

Cause: An unclassified or unclear type of heartbeat, possibly due to noise or uncommon arrhythmias.

Treatment: Further analysis required; consult a cardiologist for proper diagnosis and treatment.

FIG 4.4: Result



F

Disease Details

Disease: F

Cause: A combination of normal and ectopic impulses creating a hybrid heartbeat.

Treatment: Typically benign, requires monitoring but rarely needs medical intervention.

FIG 4.5: Result

This output helps users understand the detected abnormality and supports further clinical interpretation.

VII. CONCLUSION

This study developed and evaluated deep learning models for ECG image classification, focusing on CNN (EfficientNetB0) and RNN (LSTM) architectures. ECG images were preprocessed—resized to 224×224 pixels, normalized, and augmented through rotations, flips, and zooms—to balance classes and improve generalization. The EfficientNetB0 CNN, extended with pooling and dense layers, demonstrated high effectiveness in extracting spatial features such as P-waves, QRS complexes, and T-waves, achieving balanced precision, recall, and F1-scores across all classes. A hybrid CNN-RNN model, created by reshaping CNN outputs for LSTM processing, did not improve performance and added computational complexity, indicating that RNNs are less suited for static image-based ECG classification. Grad-CAM was applied to enhance interpretability.

Future work includes transitioning to raw ECG signal analysis using sequential models, leveraging larger real-world datasets, automated hyperparameter optimization, ensemble learning, mobile/edge deployment, and clinical validation to enhance practical applicability.

REFERENCES

[1] A. K. Hannun et al., “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” Nature Medicine, vol. 25, pp. 65–69, 2019.

[2] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in Proc. Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[4] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[5] R. R. Selvaraju et al., “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, pp. 436–444, 2015.

[7] P. Rajpurkar et al., “Deep Learning for ECG Classification: A Review,” Computers in Biology and Medicine, vol. 136, 104775, 2021.

[8] M. Zihlmann, D. Perekrestenko, and M. Tschannen, “Convolutional Recurrent Neural Networks for Electrocardiogram Classification,” Computing in Cardiology, vol. 44, 2017.

[9] MIT-BIH Arrhythmia Database, PhysioNet, Available: https://physionet.org/content/mitdb/1.0.0/. Accessed May 202

[10] N. Xiao et al., “Deep Learning-Based ECG Arrhythmia Classification: A Systematic Review,” Applied Sciences, vol. 13, no. 8, 4964, 2023.

[11] K. Malleswari et al., “Deep learning-assisted arrhythmia classification using 2-D ECG spectrograms,” EURASIP Journal on Advances in Signal Processing, 2024.

[12] M. Ullah, A. M. Yafi, and F. Elbattah, “Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation,” arXiv preprint, arXiv:2005.06902, 2020.

[13] F. N. Salamon and M. Santos, “A Hybrid Deep Learning Approach for ECG-Based Arrhythmia Classification,” Bioengineering, vol. 9, no. 4, 152, 2022.

[14] T. Warrick and N. Homsi, “Cardiac Arrhythmia Detection from ECG Combining Convolutional and Long Short-Term Memory Networks,” arXiv preprint, arXiv:1801.10033, 2018.

[15] S. Li et al., “Automatic cardiac arrhythmias classification using CNN and attention-based RNN network,” Biomedical Signal Processing and Control, vol. 86, 105125, 2023.

[16] H. Gupta and S. Pal, “Deep convolutional neural networks-based ECG beats classification to diagnose cardiovascular conditions,” Biomedical Engineering Letters, vol. 11, pp. 89–98, 2021.