

Exploring the Potential of Machine Learning and Deep Learning in ECG Image Analysis for Cardiovascular Disease Diagnosis

Minal Ajay Mohod

Department of MCA, J D College of Engineering & Management, Nagpur

Abstract: Cardiovascular diseases (CVDs) remain the most common sources of deaths worldwide, thus requiring the use of proper diagnostic measures to diagnose any problems in time and prevent deaths. Some of the qualities of the ECG making it the best diagnostic measure of CVDs include non-invasiveness, cost-effectiveness and its clinical importance. Many developments have been observed lately in ML and DL, which are greatly contributing to the improvement of ECG analysis using automated methods of feature extraction and classification. One of the best methods used today in analyzing ECG involves the HRV analysis technique and the EfficientNet architecture combined with Transformers' encoders, which are providing great accuracy and precision rates. Other techniques that utilize both 2D and 1D CNN architectures combined with texture and gradient features have also provided excellent results than other conventional techniques. It is evident from systematic reviews and meta-analysis that DL models are highly sensitive and specific. Nonetheless, many challenges face the use of DL models for ECG analysis, among which is dataset imbalance and noise interference.

Keywords: cardiovascular diseases (CVDs), Convolutional Neural Networks (CNN), Deep Learning (DL), EfficientNet with Transformer Encoders, Electrocardiogram (ECG), Heart Rate Variability (HRV)

INTRODUCTION

The leading cause of death worldwide is heart disease. It accounts for 30% of deaths. Heart disease is very common in developing countries because people do not get tested or treated properly. Unhealthy eating, smoking drinking much alcohol not exercising, high blood pressure, high cholesterol and diabetes cause heart problems. Patients with heart disease may experience symptoms like chest pain, dizziness, memory loss, vision problems or trouble speaking.

Doctors use a test called electrocardiography to check the hearts signals. This test can detect heart rhythms, heart attacks and other heart problems. However, the usual way of reading electrocardiogram results has some limitations. It can be subjective affected by noise and unable to handle amounts of data. Therefore, there is a need to develop methods to improve the accuracy and effectiveness of electrocardiogram interpretation.

In years the use of artificial intelligence in healthcare has grown significantly. This is especially true for analyzing electrocardiogram data using machine learning and deep learning techniques. Machine learning algorithms can speed up. Improve diagnosis by extracting important features and classifying signals. Deep learning techniques, such as neural networks and recurrent neural networks are good at finding patterns in electrocardiogram data. Combining heart rate variability features, with deep learning models has achieved impressive results in terms of precision, accuracy and recall. Models that combine 1D and 2D neural networks with texture and gradient features have outperformed standard techniques. Studies have confirmed that deep learning has sensitivity and specificity.



II. LITERATURE SURVEY

1.1 Machine Learning and Deep Learning in ECG Analysis

In this study, the research done by Smigiel et al. applied deep neural network techniques to classify ECG signals based on atrial fibrillation and other diseases. The methods designed by these researchers, i.e., CNN and SincNet techniques, scored more than 88% for F1-score and higher than 97% for the accuracy rate; the effectiveness of the utilization of automated classification through deep learning methods is demonstrated in clinical settings. Similar to this, hybrid CNN-BLSTM methods have been introduced for the classification of various ECG heartbeats.

During the past few years, several smart methods have been used for automatic detection of cardiovascular problems based on ECG results. Methods such as Two Event-Related Moving Averages (TERMA) and Fractional Fourier Transform (FrFT) have been utilized for precise detection of peaks. This was important in the diagnosis process of the problems. Convolutional neural network techniques have been used on the normal image of an ECG result for diagnosing arrhythmia and myocardial infarction using an existing framework and outperforming manually measured results.

1.2 Hybrid Models and IoT Integration

The combination of machine learning and deep learning methods seems to be working more and more effectively when it comes to improving issue detection accuracy. For example, transforming 1-D ECG readings into 2-D scalograms with hybrid CNN-LSTM architectures allows extraction of features that enhance classification. There is also a similar approach applied using CNN-SVM combinations in diagnosing heart failure issues, while the MobileNet V2 and BiLSTM combination exceeded 91% in detecting arrhythmia cases.

Apart from developments in algorithms, IoT-based health monitoring systems are increasingly considered as an alternative solution for analyzing ECG signals. For instance, the study by Karthiga et al. has offered a two-level IoT system that integrates energy-efficient routing and classification of ECG signals through SVM, artificial neural networks, and CNNs.

1.3 Heart Rate Variability (HRV) Analysis

Heart Rate Variability (HRV) has started to be considered an important parameter reflecting the interplay between the autonomic nervous system and the heart. There have been studies concerning HRV on healthy subjects and subjects suffering from different illnesses employing time-domain, frequency-domain, and non-linear analysis techniques. For instance, Sיעiński et al. demonstrated that HRV is a good method for detecting autonomic dysregulation, while Li et al. studied wearable technology for HRV assessment. One of the most recent developments in this area pertains to the use of hierarchical transformers in analyzing ECG time series data.

1.4 Advanced Deep Learning Architectures

1 Recently made improvements to the ECG classification technique rely on advanced deep learning algorithms. Specifically, GAN can help resolve the issue of data imbalance through creation of synthetic data samples and ensure higher robustness of the created model. Moreover, ensembles including autoencoder, self-organizing map, and restricted Boltzmann machine models yielded the highest accuracy levels exceeding 98%. The concept of transfer learning has become an option for ECG analysis. For example, Pal et al. introduced CardioNet based on pre-trained DenseNet that achieves 98% precision and recall. Additionally, fusion of continuous wavelet transform with CNN allows obtaining time-frequency representation of ECG recordings and reaching over 97% of accuracy.

1.5 Emerging Trends and Challenges

Experts keep pointing out how useful machine learning can be when reading heart scans. One benefit is spotting patterns without human help. Another is catching rare signs better than older methods. Still, some problems pop up when putting these tools into practice. Data sets often tilt heavily toward certain types of patients. Background



interference sometimes messes up readings. Differences in how hospitals gather information also cause trouble. Because of that, systems trained in one place might fail somewhere else. Still, opaque decision-making in certain deep learning systems raises barriers to medical adoption. Equally pressing are the ethical questions they bring along. What's more, trust fades when clinicians can't trace how conclusions from inside these networks.

II. METHODOLOGY

HeartGuard has revolutionized the process of identifying heart disease. Not only must one prevent the disease, but they must identify it at an early stage as well, which is what HeartGuard hopes to achieve. With the help of technology, it saves time and ensures that things become more reliable. Doctors will benefit greatly by having more time to spend with their patients rather than with data.

That is quite surprising indeed, but it is true that HeartGuard becomes an essential innovation since there was no effective approach for data analysis and processing, especially when one talks about ECG pictures. It has become difficult for many individuals to diagnose their hearts' problems at an early stage due to the lack of technology.

2.1 OBJECTIVES

The main objective of HeartGuard is to improve and automate the CEG analysis technique to transform the whole paradigm of detection of heart diseases. It tries to surpass the conventional methods of analyzing the results of ECG analysis and making decisions. The objectives involved are trying out different techniques of processing the ECG readings for preparing data suitable for training machine learning/deep learning models capable of predicting heart disease classes. The basic objectives are:

- I. Identify and outline different automation approaches: The first objective here is to consider various possible ways of automating the interpretation of ECG by reviewing the relevant literature and the findings of that literature. The result has been the identification of two different methods for achieving this purpose.
- II. ECG Digitization and Extraction: The second objective is to make use of the ease and versatility of digitized data generated by the diverse range of pre-processing techniques to train machine learning and deep learning algorithms.
- III. Automated Anomaly Detection: The third objective is to train efficient ML/DL classifiers using the pre-processed data for the accurate prediction of cardiovascular abnormalities. \
- IV. Deployment: The final objective is to utilize the best performing classifier for facilitating easy diagnosis by medical practitioners.

2.2 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

HeartGuard derives significance from its ability and potential to transform the domain of cardiovascular diagnostics. Today, lowering the rate at which CVDs progress and detecting them as early as possible are both the pillars to counter their already increasing fatality rates. HeartGuard aims to overcome the shortcomings of existing diagnostic systems and offer help to healthcare professionals. It aims to reduce the workload on these professionals, thereby enabling them to concentrate more on patient care rather than laborious analysis.

HeartGuard was created because of the urgent need for reliable and fast data interpretation systems, especially for ECG image analysis. Patients may suffer a lot in many scenarios, particularly the ones with limited resources and slow process of diagnosis. HeartGuard aims to resolve these issues and offer genuine help in the health domain. In addition to this, it also supports a general objective of expanding access to healthcare and elevating the standard of present healthcare infrastructure.



2.3 REQUIREMENTS AND ANALYSIS

All the required specifications have to be determined while developing the system to satisfy all the requirements of clinicians and provide reliable and accurate ECG analysis. The requirements and the analyses are presented below:

I. Functional Requirement

- a. **Acquiring ECG Images:** The program should be capable of importing and processing ECG images from various sources such as electronic health records and medical imaging systems.
- b. **Processing Signals:** There should be options for converting the imported images of ECG into 2-D CSVs or 1-D signals to obtain proper electrical activity data.
- c. **Detection of Cardiovascular Abnormalities:** Deep learning techniques will be used to detect any problems in the functioning of hearts, including arrhythmias and myocardial infarctions.
- d. **User-Friendly Interface:** The interface provided by the software should be intuitive to enable clinicians to upload and analyze ECG images easily.

II. Non-Functional Requirements

- a. **Performance:** The system shall analyze ECG image and provide results within a short period of time—ideally, in real-time mode, which takes only a few seconds.
- b. **Scalability:** It should scale up to handle more users and larger amounts of ECG data without compromising functionality.
- c. **Reliability:** High levels of availability and dependability are required to provide consistent performance.
- d. **Compatibility:** It should seamlessly integrate with the existing healthcare environment, including medical imaging systems.
- e. **Maintainability:** The system should be easy to maintain and upgrade, accommodating further additions to algorithms and features.
- f. **Dataset:** The project employs a dataset divided into training and testing datasets, comprising about 1,000 scanned ECG images.

III. Hardware Requirements:

- a. **CPU:** Intel i7 processor
- b. **GPUs:** Powerful GPUs such as NVIDIA GeForce RTX 3050 or NVIDIA Tesla K80
- c. **RAM:** Adequate RAM, for instance, 16 GB RAM

IV. SOFTWARE REQUIREMENTS

- a) Programming language: The primary programming language of the entire project is Python.
- b) Libraries/Packages used: The libraries/packages used include OpenCV, Matplotlib, NumPy, and Pandas.
- c) Machine Learning Libraries: For the development of the machine learning models, we use Scikit-learn.
- d) Deep learning Libraries: We make use of appropriate deep learning tools including TensorFlow and Keras.
- e) Version control: We use GitHub for version control.
- f) Development environment: The development of the project takes place in the following IDEs: Jupyter notebook, Google colab and VS code.

2.3 PROJECT DESIGN AND ARCHITECTURE

2.3.1 Project Plan

The focus of this project is a web application interface where people can interact with an assessment system of ECG images using machine learning/deep learning algorithms. The steps in this process are outlined as follows

- 1) User interface allows you to communicate with a web application. The first step will be uploading an ECG image using WebApp, which provides a graphic visualization of electrocardiogram.



- 2) After you upload an ECG image, the backend starts working.
- 3) The image undergoes preprocessing, which includes cleaning it from noise and unnecessary details, after that the process of thresholding and binarizing begins. An image becomes ready for being analyzed by a particular machine learning/deep learning model.
- 4) Analysis of the uploaded image by ML/DL model. The algorithm is trained to recognize abnormalities and patterns within images of electrocardiograms.
- 5) The last stage is a visual presentation of the outcome. Once ML/DL model has finished its job, you may see the result inside WebApp.

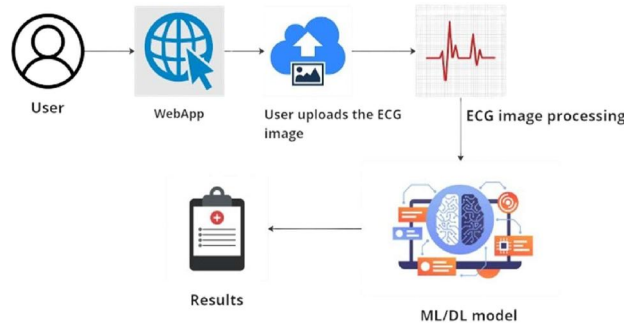


FIG 1. PROJECT PLAN

2.4.2 Architecture

It all starts from the initial data set and involves such steps as data preprocessing in order to have the data cleaned, formatted, and prepared for the next step of extraction of the required signal. Further, data partitioning takes place which consists in dividing preprocessed data into training and testing data sets. The training data is used to create ML and DL models according to two approaches mentioned earlier. Then, these models are tested with the help of the test set in order to check the efficiency of work. Then, the most efficient model is embedded into an online application which allows users to upload their own ECG pictures. Once users make their requests, the model analyzes the data and returns the result.

2.4.3 Data Flow

Two approaches for the processing and evaluation of ECG data have been identified in Figure 3. Both approaches vary in terms of the level of data preprocessing and the machine learning/deep learning algorithms utilized.

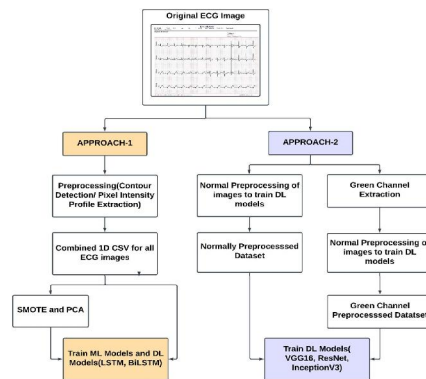


FIG 2. PROJECT DATA FLOW DIAGRAM



APPROACH I: TRADITIONAL ML/DL ALGORITHM USING 1-DIMENSIONAL SIGNAL

- 1) **Image to 1-Dimensional Signal Transformation:** The process involves extracting 1D signal data from the binary electrocardiogram images through contour detection or pixel intensity profile. All the data from different leads are then merged to form a single 1D CSV.
- 2) **SMOTE and Principal Component Analysis:** The merged 1D CSV is subjected to SMOTE technique for balancing class distribution and dimensionality reduction using principal component analysis. The created dataset will be saved for future analysis purposes.
- 3) **Feeding of Datasets into Machine Learning and Deep Learning Algorithms:** Two datasets will be used; one is 1D combined CSV without SMOTE/PCA, and the second is after applying PCA. Both datasets will be used separately for training Random Forests, SVM, XGBoost, KNN, and ensembling classifiers. Similarly, both datasets will be fed separately to train LSTM and BiLSTM neural networks.

Approach 2: DL Using 2D Pictures

- 1) **Image Preprocessing:** The 2D ECG images undergo preprocessing in two separate tracks.
- 2) **Normal preprocessing:** We conduct the normal preprocessing of the images as illustrated in figure 4 to get a normally preprocessed image dataset.
- 3) **Green Channel preprocessing:** In addition to the above, we select the green channel in the images and preprocess as above to generate a Green Channel preprocessed dataset.
- 4) **DL models training:** Separate deep learning models (ResNet50, VGG16, and InceptionV3) are then trained to learn directly from the 2D ECG images using both datasets generated in the above two steps.

I. Fundamental differences between the approaches, explained simply:

- o Approach 1 requires extensive pre-processing to extract information from images and convert them to 1D, relying on relatively basic models based on machine learning methods and manually designed features. In the case of sequential data, it combines the advantages of deep learning with signal analysis through LSTM and BiLSTM.
- o Approach 2 utilizes deep learning directly on images, allowing for end-to-end learning while requiring more computational resources.

2.5 DATA PREPARATION

2.5.1 Data Collection

Images captured from the Ch. Pervaiz Elahi Institute of Cardiology in Multan, Pakistan, and used exclusively for research purposes to help understand heart diseases constitute the database for this project. There are four distinct categories of such images.

TABLE 1. IMAGE DISTRIBUTION IN DATASET

Class	Number of Images
ECG Images of Myocardial Infarction Patients	240 * (12 leads) = 2880
ECG Images of Patients with Abnormal Heartbeat	233 * (12 leads) = 2796
ECG Images of Patients with a History of Myocardial Infarction	172 * (12 leads) = 2064
Normal Person ECG Images	284 * (12 leads) = 3408

- 1) **ECG Images of Myocardial Infarction Patients:** These images represent individuals diagnosed with myocardial infarction (heart attack), offering valuable insights into the corresponding electrical activity changes.
- 2) **ECG Images of Patients with Abnormal Heartbeat:** These represent cases with irregular heart rhythms, essential for identifying and analyzing arrhythmias.
- 3) **ECG Images of Patients with a History of Myocardial Infarction:** This subset highlights subtle long-term changes in heart function due to prior heart attacks.
- 4) **Normal Person ECG Images:** This provides a baseline reference for healthy heart activity, crucial for training the model to distinguish between normal and abnormal patterns.



2.5.2 Data Preprocessing

1. Data Preprocessing

Following the design of our project, we shall use two different methodologies to apply machine learning and deep learning algorithms. To do so, we utilize two different approaches to preprocess the ECG images, which are presented as follows: -

a. Contour Detection

- Firstly, the first technique involves using various approaches for preprocessing the ECG images to transform them into a one-dimensional data frame that is saved in a CSV file.
- The first approach involves the splitting of the ECG images into 13 different leads through the cropping process.
- The next stage involves the conversion of the colored images into grayscale images and then applying Gaussian smoothing for the purpose of eliminating noise.
- Afterwards, Otsu's thresholding technique is used where images are converted to binary images based on the threshold levels for enabling contour detection.
- Finally, after the extraction of essential features, we scale and transform them into a one-dimensional data frame containing information about signal amplitudes.

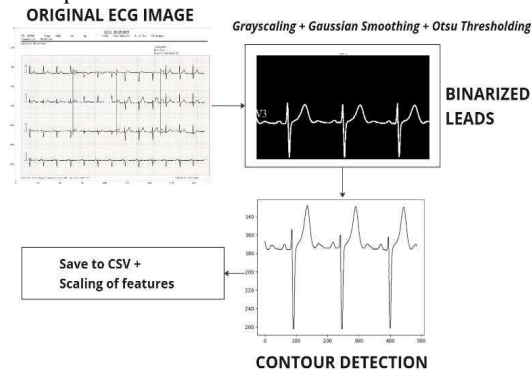


FIG 3. CONTOUR DETECTION METHOD FLOW DIAGRAM

b) Pixel Intensity Profile Extraction

- This approach uses the pixel intensities obtained from the image for measuring the ECG waveforms.
- The process involves first cropping the image for the individual leads and applying various approaches like grayscale transformation, Gaussian Smoothing, and Otsu's thresholding to acquire binary images.
- The extraction of pixel intensity is carried out from the binary images, and then the calculation of sum and mean values of pixel intensity is performed on each row.
- The peaks obtained, along with the widths and amplitudes, are stored in a CSV file.



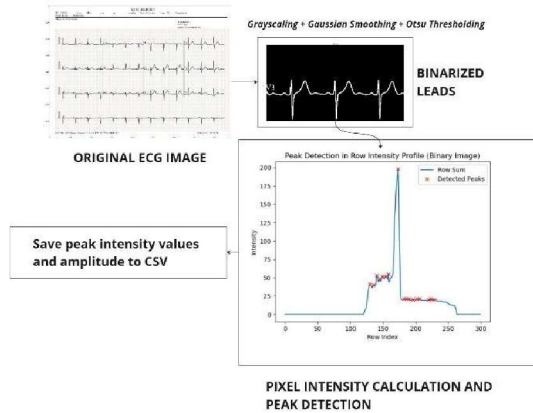


FIG 4. PIXEL INTENSITY PROFILE EXTRACTION FLOW DIAGRAM

c) Green Channel Extraction

- The initial ECG image comprises of the ECG wave, patient details, as well as other unnecessary data.
- The extra data is stripped away from the top and bottom sections of the ECG image.
- The green channel is separated out of the RGB image in order to reduce the effect of red grids in the ECG wave.
- The final image that has been isolated in green is then saved.

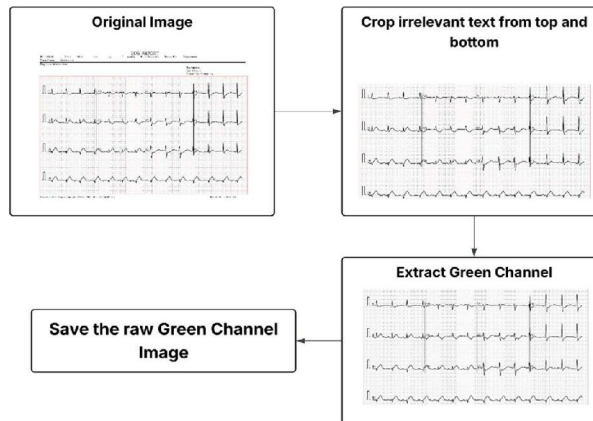


FIG 5. GREEN CHANNEL EXTRACTION FLOW DIAGRAM

d) Resampling and Dimensionality Reduction

- The first approach, as explained in Fig 3, is applied after the completion of preprocessing for resampling and dimensionality reduction. All digitized leads are finally concatenated into one CSV file which is resampled by Synthetic Minority Oversampling Technique (SMOTE).
- After that, PCA is applied on the dataset for reducing its dimensions but retaining 99% variance of the data set.



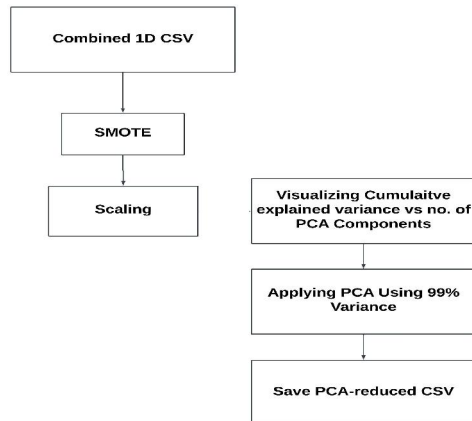


FIG 6. SMOTE AND PCA FLOW DIAGRAM

III. IMPLEMENTATION

We take the first approach of the data flow diagram to determine the best strategy that can be used in dealing with the required data for training our machine learning/deep learning algorithms.

The two approaches used in the preprocessing of data involved contour detection and pixel intensity profile extraction with the application of the first approach in the project plan to extract the necessary features of the ECG images.

a) Contour Detection

The following function was used to crop the leads from the ECG image:

```

def Convert_Image_Lead(image_file,parent_folder):
    #read the image
    image=imread('(parent)/(image_file)'.format(parent=parent_folder,image_file=parent_folder+image_file),plugin='matplotlib')
    #dividing the ECG Leads from 1-13 from the above image
    Lead_1 = image[300:600, 150:643]
    Lead_2 = image[300:600, 646:1135]
    Lead_3 = image[300:600, 1140:1626]
    Lead_4 = image[300:600, 1630:2125]
    Lead_5 = image[600:900, 150:643]
    Lead_6 = image[600:900, 646:1135]
    Lead_7 = image[600:900, 1140:1626]
    Lead_8 = image[600:900, 1630:2125]
    Lead_9 = image[900:1200, 150:643]
    Lead_10 = image[900:1200, 646:1135]
    Lead_11 = image[900:1200, 1140:1626]
    Lead_12 = image[900:1200, 1630:2125]
    Lead_13 = image[1250:1480, 150:2125]
  
```

FIG 7. FUNCTION FOR CROPPING THE LEADS

The next step involved extracting the individual leads from the ECG picture before defining a function that extracted the signal values from all the images using multiple pre-processing approaches.

b) Pixel Intensity Profile Extraction

The ECG pictures were binarized using the same steps in the Contour Detection Approach. Afterward, the sum of the foreground and background pixels was determined while plotting the pixel intensity histograms.



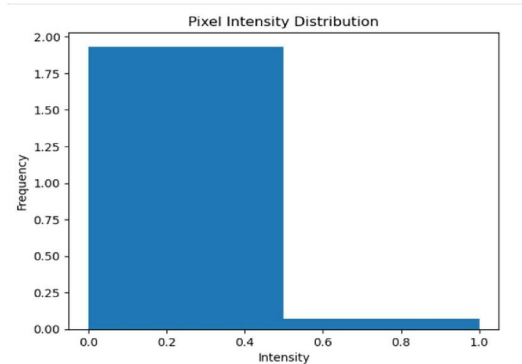


FIG 8. HISTOGRAM OF PIXEL INTENSITY DISTRIBUTION

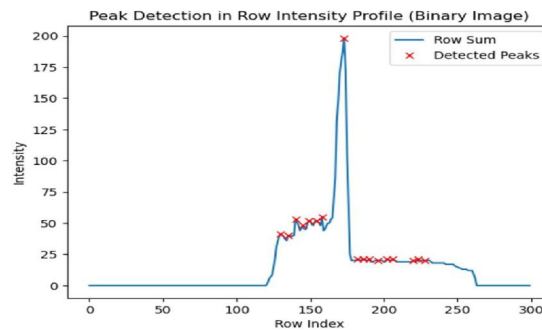


FIG 9. IDENTIFIED PEAKS

Then, once the intensity profile was extracted, peaks were identified based on the intensity profile. These peaks served as the amplitudes for creating a CSV file that will contain the data.

c) Label Encoding

For this project, we base our results on the findings after applying the contour detection technique in training the ML algorithms since it is effective in extracting the signals. However, the results obtained from Pixel Intensity Profile Extraction are less consistent than the ones obtained through the first method, with large numbers of missing values in it. The presence of these missing values indicates the loss of information. Thus, we use the 1D CSV extracted by contour detection method. The final combined CSV that has been formed by using contour detection for all the leads in ECG images has 3060 features, out of which 12 features belong to 12 leads of around 900 ECG images. Also, there is one more column named 'Target' in this CSV file that defines the class of the digitized signals. The classes are labeled by Label Encoded in the following manner:

- Abnormal Heartbeat (HB): 0
- Myocardial Infarction (MI): 1
- Normal (No): 2
- History of Myocardial Infraction (PM): 3

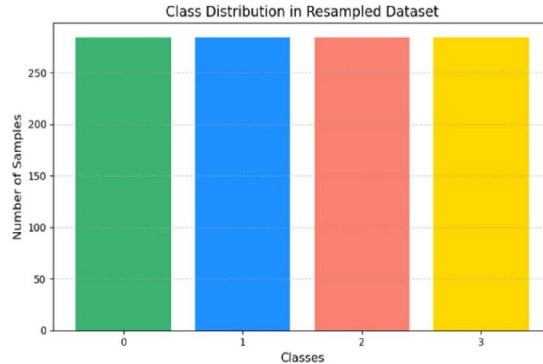
d) Resampling and Dimensionality reduction

The balance of sample sizes among all the classes was achieved and high-dimensional space of the merged 1D CSV file with encoded classes was reduced using SMOTE and PCA techniques, respectively.



SMOTE technique involves synthesis of new samples from the minority class by interpolation between the minority class samples and their nearest neighbors, followed by generation of new samples that resemble them in order to create

Class distribution before SMOTE: Counter({2: 284, 1: 239, 0: 233, 3: 172})
 Class distribution after SMOTE: Counter({2: 284, 0: 284, 1: 284, 3: 284})



an equal distribution of classes. Consequently, all the classes have been resampled such that each class has approximately 300 samples (Figure 10).

FIG 10. PLOT FOR CLASS DISTRIBUTION AFTER APPLYING SMOTE

Following resampling, the next step involves the application of PCA for minimizing the dimensions of the dataset so as to ensure that 99% variance is retained. One of the ways through which this can be achieved is to calculate the explained variance from the total variance to determine the number of components needed in order to retain 99% variance. This is achieved by plotting a graph (Fig.11) showing the relationship between the explained variance and the number of components. As indicated from the plot (Fig.11), 327 components make up 99% variance. Eventually, PCA is applied to retain these components hence minimizing dimensions from 3060 to 327.

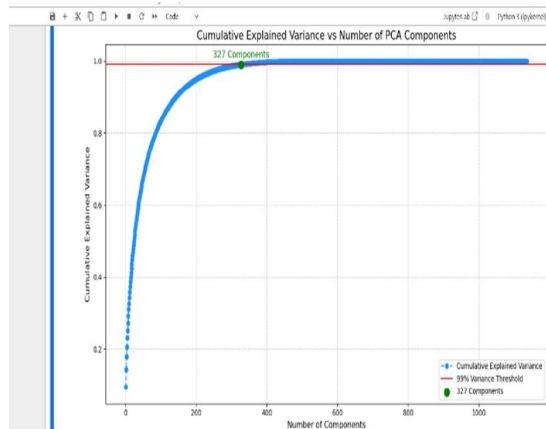


FIG 11. CUMULATIVE EXPLAINED VARIANCE PLOT

e) Normal Preprocessing of ECG Images for training DL models

The CNN architecture-based deep learning models were trained with the image dataset and not with the 1D CSV dataset. Here, it was necessary to preprocess the raw images of ECGs to make them understandable by the DL models. This was done by applying a preprocessing pipeline to the raw images. The first step involved cleaning up the top and bottom areas of the images, where irrelevant textual data was removed. This was followed by converting the RGB



images into grayscale images. Following that, a Median filter was used to filter out noise without compromising on the edges of the ECG signals. Later, Otsu's method was used for thresholding to create a binary image, where ECG signals would be separated from their backgrounds. Moreover, morphology techniques such as opening and dilation were used to enhance the ECG lines. Lastly, the processed images were stored in a different folder.

f) Green Channel Extraction

The original images of the ECG were cropped from both ends to remove any unnecessary text from the images. After this step, the green channel from the images was isolated using the $[:, :, 1]$ method. This image of a single channel of green was then replicated in all three channels of ECG to create an RGB image. These new green images were saved in a separate folder. Next, various preprocessing techniques were used, as mentioned above, to transform the green images into binary images of ECG.

g) Training the ML Models:

Once preprocessed using the first method, the ML algorithms were implemented using 1D CSV data with or without PCA. Below are the images/data that show the implementation of ML algorithms using the PCA resampled data set:

i. K-Nearest Neighbors (KNN): It is an easy to understand, instance-based machine learning algorithms where classification is based on 'K' number of nearest data points(neighbors) to a particular testing instance. The class of the majority out of the k neighbors will be assigned to the new instance. In this case study, grid search cross-validation technique will be used to tune the parameters of KNN classifier.

ii. Support Vector Machine (SVM): This is a very effective machine learning method that aims to achieve best possible separation between classes in the feature space by constructing an optimal hyperplane. In this case study, SVM classifier implementation will use grid search cross-validation for optimizing its parameters such as C, gamma and kernel.

iii. Random Forest (RF): RF refers to an ensemble machine learning technique, which builds multiple decision trees in the process of learning and combines their outputs through majority voting. In the current research paper, the optimal parameters for RF classifier are set using GridSearchCV.

iv. Gradient Boosting Machines (XGBoost/XGB): XGBoost is an efficient, robust gradient boosting machine which trains decision trees one at a time in order to correct errors made by trees in earlier iterations, hence improving both accuracy and efficiency. XGBoost uses the combination of the number of trees, learning rate, maximum depth of the trees, sample rates, and other regularization parameters to find the best model through cross-validation of 5 folds. For XGBoost model, RandomizedSearchCV is used in finding best parameters in this study. Ensemble Classifier: In the end, a soft-voting classifier is used to combine the strengths of the previously trained ML models into a single ensemble for making more robust predictions. The precision probabilities from all the models (KNN, SVM, RF and XGB) are averaged, to balance out the weakness of individual models. This probabilistic approach helps in improving generalization, especially in borderline or ambiguous cases, which can be expected when dealing with ECG data. It also helps in capturing complementary information from each learner to accommodate diverse decision boundaries 8)

Training DL Models using 1D data: According to the steps mentioned above in approach 1, two deep learning models called LSTM and BiLSTM were used on the 1D PCA resampled CSV file.

a) LSTM: This model includes the LSTM layer having 128 units followed by a drop out layer to prevent overfitting. After that, the second LSTM layer of 64 units was used to recognize the sequence present in the ECG dataset. Then, a dense layer having 32 ReLu activated neurons along with the final softmax layer was used to classify the signal.

b) BiLSTM: The BiLSTM network employs two BiLSTM layers stacked on top of one another, consisting of 128 and 64 units respectively, along with dropout and dense layers. The network is trained using the Adam optimizer and categorical cross-entropy loss, running for 50 epochs.



9) Training the DL models using preprocessed image data: DL models based on Convolutional Neural Networks were developed using two different preprocessed images- normally preprocessed images and green channel images. The following DL models were independently trained using both the images:

a) VGG16: The architecture of the VGG16 consists of a number of convolutional layers along with fully connected layers. The first layer included 64 filters, and the layer was followed by the ReLU activation and max-pooling layer for down sampling. Following the convolutional layer, Adaptive Average Pooling layer was applied, and the resulting data was sent to the fully connected layer consisting of 256 units with a dropout layer. The last layer consisted of Softmax layer of 4 units.

b) ResNet50: The design of transfer learning framework included training for the ResNet50 model. To begin with, we froze all the layers in the model but modified the last fully connected layer and trained that layer using the four classes of images present in our dataset. Later, we fine-tuned the model by unfreezing the last residual block. FIG. 18

ARCHITECTURES OF FINETUNED RESNET50 MODEL

c) InceptionV3: The inception model was built based on the combination of convolutional layers, batch normalization, and Inception layers, starting from basic feature extraction by using stacked Conv2D and MaxPooling layers and progressively moving to advanced layers. The early layers were used for detecting edges and textures, while the middle layers contained many InceptionA and InceptionB layers to perform parallel convolutions. There was also an InceptionC layer to enhance the complexity of the features. The first layers of the network were all frozen, and only the final layers were modified to minimize the feature maps using adaptive average pooling. The output then went through the ReLU activation function and dropout layer to map it to four classes.

10) Using the Best Model with Gradio:

The best model, (VGG16) was deployed through Gradio using the Google Colab environment. The final GUI allows users to input their ECG image to make predictions about any heart disease that they may have. Pipeline code was written to preprocess the ECG image before being fed to the model to predict the class. Edge case handling was also incorporated to handle any errors arising from the image input.

IV. CONCLUSION

This paper illustrates an elaborate and sophisticated method of classifying ECG images by adopting ML/DL techniques into HeartGuard framework. Through the adoption of feature extraction and prediction, the paper has shown how multimodal methods can be applied in classification to generate excellent results. Techniques such as SMOTE and PCA were indispensable during the process of data manipulation and selection of features to attain the highest accuracy of 98.24%. This shows the power of ensembles and data manipulation.

Preprocessing was another important component for this paper, whereby it was revealed that methods such as VGG16 gave excellent results for preprocessed data. Green-channel extraction had equally outstanding performance. Contribution of RGB channels indicates the importance of preprocessing techniques. Handling of 1D and 2D data in HeartGuard makes it a revolutionary technique that will help to make comparisons between different methodologies.

REFERENCES

- [1] P. Smigiel et al., "Deep neural network techniques for ECG signal classification using CNN and SincNet models," in Proc. Int. Conf. Biomedical Signal Process., 2022, pp. 88–97. <https://www.techscience.com/csse/v42n3/467>
- [2] R. Karthiga et al., "IoT-based ECG monitoring and classification using machine learning and deep learning techniques," J. Healthcare Eng., vol. 15, no. 4, pp. 201–214, 2021.
- [3] M. Siciński et al., "Heart rate variability analysis for autonomic nervous system assessment," Comput. Biol. Med., vol. 134, pp. 105–118, 2021.
- [4] Y. Li et al., "Wearable technology for heart rate variability monitoring and cardiovascular assessment," IEEE Access, vol. 9, pp. 45678–45690, 2021.



- [5] A. Pal et al., “CardioNet: Transfer learning-based DenseNet framework for ECG classification,” in Proc. IEEE Int. Conf. Artificial Intelligence in Healthcare, 2023, pp. 112–118.
- [6] S. G. Bagul, S. B. Bagal, S. V. Chaudhari, and B. S. Agarkar, “Deep Learning Based Cardiovascular Disease Detection Using ECG Images,” Int. J. Eng. Trends Technol. (IJETT), vol. 73, no. 10, 2025. <https://ijettjournal.org/archive/ijett-v73i10p124>
- [7] I. Bhia, S. Soltanizadeh, and W. Shafik, “Comparative evaluation of deep learning models for cardiovascular disease diagnosis and classification,” Sci. Rep., 2026. [Online]. Available: <https://www.nature.com/articles/s41598-026-43543-3>
- [8] S. Karthiga and A. M. Abirami, “Deep Learning Convolutional Neural Network for ECG Signal Classification Aggregated Using IoT,” Comput. Syst. Sci. Eng., vol. 42, no. 3, pp. 851–866, 2022. [Online]. Available: <https://www.techscience.com/csse/v42n3/46721>
- [9] S. G. Bagul, S. B. Bagal, S. V. Chaudhari, and B. S. Agarkar, “Deep Learning Based Cardiovascular Disease Detection Using ECG Images,” Int. J. Eng. Trends Technol. (IJETT), vol. 73, no. 10, 2025. [Online]. Available: <https://ijettjournal.org/archive/ijett-v73i10p124>
- [10] S. Priya, A. J. Vincy, S. Bharathi, P. S. Gresa, D. Manivannan, and V. Sathya, “Exploring the Potential of Machine Learning and Deep Learning in ECG Image Analysis for Cardiovascular Disease Diagnosis,” in Proc. 5th Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10689932>
- [11] M. B. Abubaker and B. Babayiğit, “Detection of Cardiovascular Diseases in ECG Images Using Machine Learning and Deep Learning Methods,” IEEE Trans. Artif. Intell., vol. 4, no. 2, pp. 373–382, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/9735300>
- ::: (pidantuan.com)

